LCARS: A Spatial Item Recommender System

HONGZHI YIN and BIN CUI, Peking University
YIZHOU SUN, Northeastern University
ZHITING HU, Peking University
LING CHEN, University of Technology, Sydney

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 to the traditional collaborative filtering-based recommender systems. The problem becomes even more challenging when people travel to a new city where they have no activity information.

In this article, we propose LCARS, a location-content-aware recommender system that offers a particular user a set of venues (e.g., restaurants and shopping malls) 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 cooccurrence patterns and exploiting item contents. The online recommendation part takes a querying user along with a querying city as input, and automatically combines the learned interest of the querying user and the local preference of the querying city to produce the top-k recommendations. To speed up the online process, a scalable query processing technique is developed by extending both the Threshold Algorithm (TA) and TA-approximation algorithm. We evaluate the performance of our recommender system on two real datasets, that is, DoubanEvent and Foursquare, and one large-scale synthetic dataset. 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. Besides, the experimental analysis results also demonstrate the excellent interpretability of LCARS.

Categories and Subject Descriptors: H.3.3 [**Information Search and Retrieval**]: Information Search and Retrieval—*Information filtering*; I.2.6 [**Artificial Intelligence**]: Learning; J.4 [**Computer Applications**]: Social and Behavior Sciences

General Terms: Algorithms, Design, Experimentation, Performance

Additional Key Words and Phrases: Recommender system; location-based service; probabilistic generative model; TA algorithm; cold start

ACM Reference Format:
Hongzhi Yin, Bin Cui, Yizhou Sun, Zhiting Hu, and Ling Chen. 2014. LCARS: A spatial item recommender system. ACM Trans. Inf. Syst. 32, 3, Article 11 (June 2014), 37 pages.
DOI: http://dx.doi.org/10.1145/2629461

This research is supported by the National Natural Science Foundation of China under Grant No. 61272155.
Authors' addresses: H. Yin, B. Cui, and Z. Hu, Key Lab of High Confidence Software Technologies (MOE), School of EECS, Peking University, Science Building 1, Beijing 100871, China; email: bin.cui@pku.edu.cn; Y. Sun, College of Computer and Information Science Northeastern University, 360 Huntington Avenue, Boston, MA 02115; L. Chen, QCIS, University of Technology, Sydney, Australia.

DOI: http://dx.doi.org/10.1145/2629461

1. INTRODUCTION

Newly emerging event-based social network services (EBSNs), such as Meetup (www.meetup.com), Plancast (www.plancast.com) and DoubanEvent (www.douban.com/events/) have provided convenient online platforms for users to create, spread, track and attend social *events* which are going to be held in some physical locations [Liu et al. 2012]. On these web services, users may propose social events, ranging from informal get-togethers (e.g., movie night and dining out) to formal activities (e.g., culture salons and business meetings) by specifying when, where and what the event is. After the created event is available to the public, other users may express their intent to join event by replying "yes," "no" or "maybe" online. Meanwhile, the advances in location-acquisition and wireless communication technologies enable users to add a location dimension to traditional networks, fostering a 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* (e.g., restaurants in New York) via mobile devices.

In this article, 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 recommendation, and the second question corresponds to event recommendation. By answering these two questions, we can satisfy the personalized information needs for people in their daily routines and trip planning. For simplicity, we use 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 location l_u such as a city, 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 exploring 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 [Levandoski et al. 2012; Horozov et al. 2006], which directly use collaborative filtering-based methods [Ricci and Shapira 2011] over spatial items. Second, the observation of travel locality [Levandoski et al. 2012] makes the task more challenging if a user travels to a new place where he/she has no activity history. The observation of travel locality on EBSNs and LBSNs shows that users tend to travel a limited distance when visiting venues and attending events. In the analysis of Foursquare data, we observe that 45% of users travel 10 miles or less and 75% of users travel 50 miles or less. Another 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 [Scellato et al. 2011b], 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, because 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 in l_v. However, the pure CF-based methods fail to do so. For the item-based CF [Linden et al. 2003; Sarwar et al. 2001; Deshpande and Karypis 2004], there

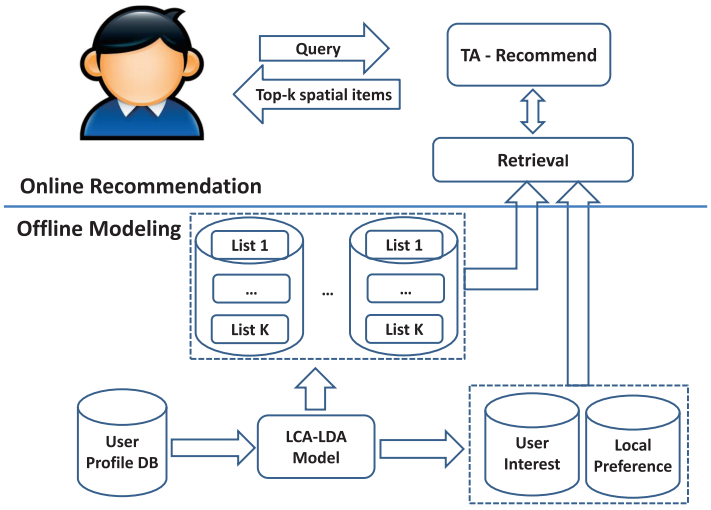


Fig. 1. The Architecture Framework of LCARS.

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 [Adomavicius and Tuzhilin 2005], 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 activity histories left by other people when they traveled in the querying city are valuable resources 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, especially the *new city* problem. 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. It is worth mentioning that LCA-LDA can also capture the item cooccurrence patterns to link relevant items together, just like item-based collaborative filtering methods. Thus, LCA-LDA not only has the ability to link spatial items together by their common users, just like the collaborative filtering-based methods, but also possesses the capability of linking items together which do not have any overlap of users by their contents. So LCA-LDA is able to facilitate people's travel not only near their home regions but also to cities that are new to them. 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 certain granularity such as cities. For each location, as is shown in Figure 1, we pre-compute K lists of items according to the K latent topics learned by offline model LCA-LDA. In each list L_z, items are sorted based on their generative probabilities with respect to the corresponding topic z. At query time, we access items from the K sorted lists within the querying city l_u and compute top-k items by running the TA algorithm. The algorithm has the nice property of terminating early without scanning all items. Specifically, it terminates when the ranking score of the k-th item in the result list is higher than the threshold score. This TA-based scheme allows us to efficiently return the top-k recommendations by examining the minimum number of items.

In practice, the querying user may be satisfied with an *approximation* top-k list to get a faster query response. Given $\rho > 1$, we define a ρ-approximation to the top-k recommendations for a ranking function $S(u, l_u, v)$ to be a ranked list of k items (each along with its ranking score) such that for each v among these k items and each v' not among these k items, $\rho S(u, l_u, v) \geq S(u, l_u, v')$, while $S(u, l_u, v) \geq S(u, l_u, v')$ needs to hold in the exact top-k recommendation. Thus, to provide an approximated top-k recommendation with a faster online speed, we design a TA-approximation algorithm based on the TA scheme. Furthermore, the TA-approximation algorithm enables an interactive process where at all times LCARS can show the querying user its current view of the top-k recommendations along with a guarantee of ρ-approximation to the exact answer. At any time, the querying user can decide whether he/she is satisfied with the recommendation results and stop the algorithm.

To sum up, we focus on the problem of spatial item recommendation in this article, especially the spatial item recommendation to users who travel out of town (e.g., in new cities). Note that we presented our preliminary study of spatial item recommendation in [Yin et al. 2013b]. In this article, we extended [Yin et al. 2013b] with an in-depth investigation and performance analysis. Specifically, this article makes the following additional contributions: First, we provide a more comprehensive analysis and review of related work. Second, we present the detailed inference procedure of our LCA-LDA model, and provide extensive analysis and discussion about our designed TA-based online algorithm. Third, based on the TA scheme, we design a new TA-approximation algorithm to provide approximated top-k answers online, which enriches our spatial item recommender system LCARS by enabling an interactive process. Fourth, we conduct a more extensive performance analysis (e.g., recommendation effectiveness, recommendation efficiency and the tradeoff between them) and case study (e.g., profile study, local preference influence study and analysis of latent topics), using two publicly available real-life datasets and one large-scale synthetic dataset.

The primary contributions of our research are summarized as follows.

—We argue that both 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.

Table I. Notations Used in the Article

SYMBOL	DESCRIPTION
U, V, L, C	the set of users, spatial items, locations, content words
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}
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

—We design a TA-approximation algorithm to provide an approximated top-k recommendation, enabling an interactive process where LCARS can show the querying user its progressive view of the top-k recommendations along with a guarantee of ρ-approximation to the exact answer.
—We conduct extensive experiments to evaluate the performance of our recommender system on two large-scale real datasets. 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. Besides, the empirical analysis results also show the excellent interpretability of our LCARS.

The remainder of the article is organized as follows. Section 2 details the offline modeling part of our recommender system LCARS. Section 3 presents the online recommendation part of LCARS. We report the experimental results in Section 4. Section 5 reviews the related work and we conclude the article in Section 6.

2. OFFLINE MODELING

In this section, we first introduce the key data structures and notations used in this article, and then present the offline modeling 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 article. Table I lists the relevant notations used in this article.

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 $S \subseteq U \times V \times L$, 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, that is, $(u,\ v,\ l_v,\ c_v) \in D$ where $u \in U$, $v \in V$, $l_v \in L$, and $c_v \in C_v$ (i.e., 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, that is, $D_{uv} = \{(u,\ v,\ l_v,\ c_v) : c_v \in C_v\}$.

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, $D_{uv} \subseteq D_u$.

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

It is worth mentioning that each topic z corresponds to two topic models in our work, that is, 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 and local attractions.

2.2. Location-Content-Aware LDA Model

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.

2.2.1. 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. As shown in Figure 2, LCA-LDA considers both 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 computed according to the following LCA-LDA model.

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

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

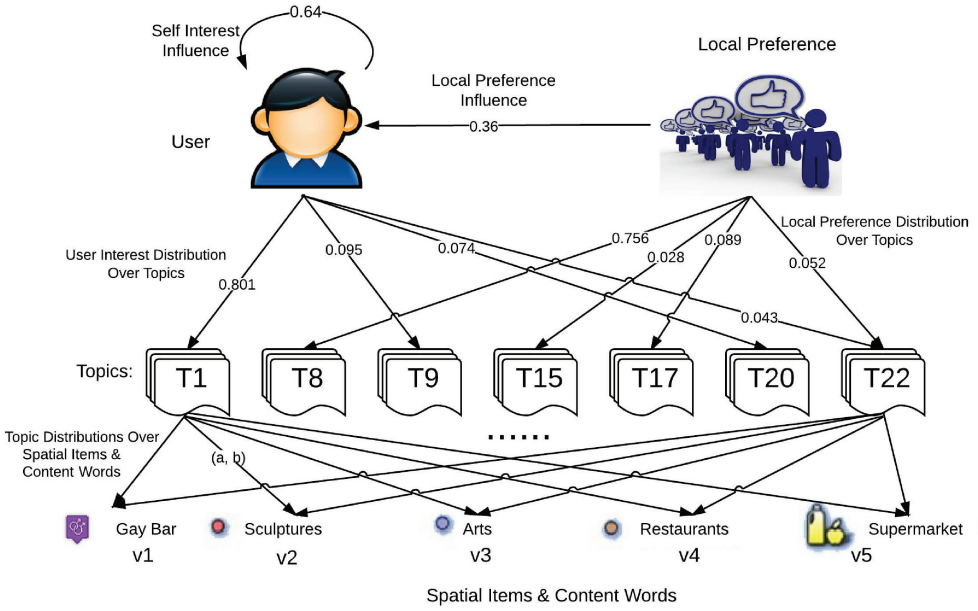


Fig. 2. Example of LCA-LDA model.

$1-\lambda_u$. 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).

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

$$P(v|\theta_u, \theta'_{l_u}, \phi, \phi') = \sum_{c \in C_v} P(v, c|\theta_u, \theta'_{l_u}, \phi, \phi') \tag{2}$$

$$P(v|\theta_u, \phi, \phi') = \sum_{c \in C_v} P(v, c|\theta_u, \phi, \phi') \tag{3}$$

$$P(v|\theta'_{l_u}, \phi, \phi') = \sum_{c \in C_v} P(v, c|\theta'_{l_u}, \phi, \phi'), \tag{4}$$

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. As shown in Figure 2, the weight (a, b) on the edge linking topic $T1$ and item $v2$ represents the probabilities of $T1$ generating item $v2$ and its associated content word "Sculptures", respectively. It should be noted 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 cooccurrence and content word cooccurrence patterns. This encodes our prior knowledge that spatial items having many common users or similar content should be clustered into the same

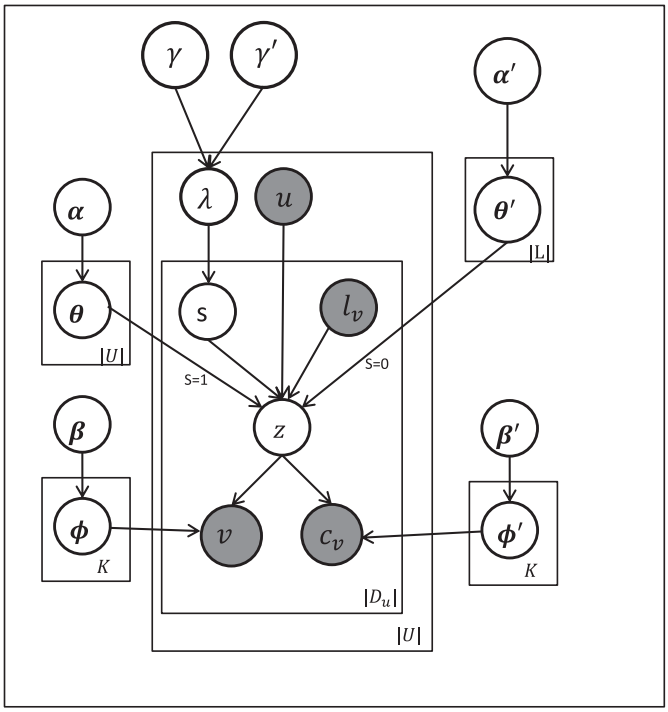


Fig. 3. The graphical representation of LCA-LDA.

topic with high probability.

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

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

Summary of LCA-LDA. The proposed LCA-LDA is a latent class statistical mixture model. It can be represented by a graphical model in Figure 3 and a generative process in Algorithm 1. The model discovers (1) user's personal interest distribution over latent topics, θ_u; (2) local preference distribution over latent topics, θ'_{l_u}; (3) topic distribution over items, ϕ_z; (4) topic distribution over content words, ϕ'_z; (5) the mixing weight λ_u. The generative model aims to capture the process of human behaviors and/or reasoning for decision making. For example, a querying user u wants to choose a venue v in city l_u to visit. The person may choose one based on personal interest or choose the one that is most popular in l_u. In the case that u wants to choose the venue based on personal interest (with a certain probability λ_u), a topic z is first chosen according to the personal interest distribution θ_u, and then the selected topic z in turn generates a venue v and relevant content words C_v following on the topic's item and content word

ALGORITHM 1: Probabilistic generative process in LCA-LDA

Input: a user profile dataset D;
Output: estimated parameters θ, θ', ϕ, ϕ' and λ;
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

generative distributions (i.e., ϕ_z and ϕ'_z), respectively. In the case that user u follows the local preference of l_u, l_u would generate an item and its content words following on l_u's preference distribution θ'_{l_u} similarly. Thus, this model simulates the process that how u picks the spatial item v, including how the local preference of l_u influences u's decision. As a running example in Figure 2, the user is influenced by personal interest and the local preference with probabilities 0.64 and 0.36, respectively. The top-4 topics of the user's interest and the local preference are also shown respectively, where the weights representing user's personal interest and the local preference in the topics are labeled in the corresponding edges. We can see that there is only one overlapped topic for the user and the local preference, and their dominated topics are different (i.e., T1:0.801 vs. T8:0.756). The probabilities of topic generating items and their associated content words are also labeled in the corresponding edges. For example, the weights (a, b) on the edge linking topic T1 and item v2 represent the probabilities of T1 generating item v2 and its associated content word "Sculptures", respectively.

With the model hyperparameters α, α', β, β', γ and γ', the joint distribution of the observed and hidden variables $\boldsymbol{v}$, $\boldsymbol{c_v}$, $\boldsymbol{z}$ and $\boldsymbol{s}$ can be written as follows.

$$\begin{aligned} &P(\boldsymbol{v}, \boldsymbol{c_v}, \boldsymbol{z}, \boldsymbol{s}|\alpha, \alpha', \beta, \beta', \gamma, \gamma') \\ &= \int \cdots \int P(\boldsymbol{v}|\phi, \boldsymbol{z}) P(\phi|\beta) P(\boldsymbol{c_v}|\phi', \boldsymbol{z}) P(\phi'|\beta') \\ &\quad P(\boldsymbol{z}|\theta, \theta', \boldsymbol{s}) P(\theta|\alpha) P(\theta'|\alpha') P(\boldsymbol{s}|\lambda) P(\lambda|\gamma, \gamma') \\ &\quad d\phi\ d\phi'\ d\theta\ d\theta'\ d\lambda \end{aligned} \tag{7}$$

2.2.2. Model Inference. The computation of the posterior distribution of the hidden variables is intractable for the LCA-LDA model. Therefore, we follow the studies [Tang et al. 2008, 2012] and use approximate method collapsed Gibbs sampling to obtain samples of the hidden variable assignment and to estimate unknown parameters $\{\theta, \theta', \phi, \phi', \lambda\}$ in the LCA-LDA. As for the hyperparameters α, α', β, β', γ and γ', for simplicity, we

take a fixed value (i.e., $\alpha = \alpha' = 50/K$, $\beta = \beta' = 0.01$, $\gamma = \gamma' = 0.5$). Note that Gibbs sampling allows the learning of a model by iteratively updating each latent variable given the remaining variables. In the sampling procedure, we begin with the joint probability of all user profiles in the dataset. 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.

To begin with, we need to compute the conditional probabilities $P(s_{ui}|\boldsymbol{s}_{\neg ui}, \boldsymbol{z}, \boldsymbol{v}, \boldsymbol{c})$ and $P(z_{ui}|\boldsymbol{s}, \boldsymbol{z}_{\neg ui}, \boldsymbol{v}, \boldsymbol{c})$, where $\boldsymbol{s}_{\neg ui}$ and $\boldsymbol{z}_{\neg ui}$ represent the s and z assignments for all the spatial items except v_{ui} respectively. According to the Bayes rule, we can compute these conditional probabilities in terms of the joint probability distribution of the latent and observed variables shown in Equation (7). Next, to make the sampling procedure clearer, we factorize this joint probability as:

$$P(\boldsymbol{v}, \boldsymbol{c_v}, \boldsymbol{z}, \boldsymbol{s}) = P(\boldsymbol{v}|\boldsymbol{z})P(\boldsymbol{c_v}|\boldsymbol{z})P(\boldsymbol{z}|\boldsymbol{s})P(\boldsymbol{s}). \tag{8}$$

By integrating out the parameter ϕ in Equation (7) we can obtain the first term in Equation (8):

$$P(\boldsymbol{v}|\boldsymbol{z}) = \left(\frac{\Gamma(\sum_v \beta_v)}{\prod_v \Gamma(\beta_v)}\right)^K \prod_z \frac{\prod_v \Gamma(n_{zv} + \beta_v)}{\Gamma(\sum_v (n_{zv} + \beta_v))}, \tag{9}$$

where n_{zv} is the number of times that spatial item v has been generated by topic z. $\Gamma(\cdot)$ is the gamma function. Similarly, for the second term $P(\boldsymbol{c_v}|\boldsymbol{z})$ in Equation (8), we integrate out the parameter ϕ' and get:

$$P(\boldsymbol{c_v}|\boldsymbol{z}) = \left(\frac{\Gamma(\sum_c \beta'_c)}{\prod_c \Gamma(\beta'_c)}\right)^K \prod_z \frac{\prod_c \Gamma(n_{zc} + \beta'_c)}{\Gamma(\sum_c (n_{zc} + \beta'_c))}, \tag{10}$$

where n_{zc} is the number of times that content word c has been generated by topic z.

Next, we evaluate the third term $P(\boldsymbol{z}|\boldsymbol{s})$ in Equation (8). By integrating out the parameters θ_u and θ'_l, we compute:

$$\begin{aligned} P(\boldsymbol{z}|\boldsymbol{s}) = &\left(\frac{\Gamma(\sum_z \alpha_z)}{\prod_z \Gamma(\alpha_z)}\right)^{|U|} \prod_u \frac{\prod_z \Gamma(n_{uz} + \alpha_z)}{\Gamma(\sum_z (n_{uz} + \alpha_z))} \\ &\cdot \left(\frac{\Gamma(\sum_z \alpha'_z)}{\prod_z \Gamma(\alpha'_z)}\right)^{|L|} \prod_l \frac{\prod_z \Gamma(n_{lz} + \alpha'_z)}{\Gamma(\sum_z (n_{lz} + \alpha'_z))}, \end{aligned} \tag{11}$$

where $|U|$ is the number of users, and $|L|$ is the number of locations (e.g., cities); n_{uz} is the number of times that topic z has been sampled from the multinomial distribution specific to user u; n_{lz} is the number of times that topic z has been sampled from the multinomial distribution specific to location l.

Last, we need to derive the fourth term $P(\boldsymbol{s})$. By integrating out λ_u we have:

$$P(\boldsymbol{s}) = \left(\frac{\Gamma(\gamma + \gamma')}{\Gamma(\gamma)\Gamma(\gamma')}\right)^{|U|} \prod_u \frac{\Gamma(n_{us_1} + \gamma)\Gamma(n_{us_0} + \gamma')}{\Gamma(n_{us_1} + n_{us_0} + \gamma + \gamma')}, \tag{12}$$

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.

Now, the conditional probability can be obtained by multiplying and canceling of terms in Equations (9–12). Thus, we first sample the coin s according to the posterior

probability:

$$P(s_{ui} = 1|\boldsymbol{s}_{\neg ui}, \boldsymbol{z}, .) \propto \frac{n_{uz_{ui}}^{\neg ui} + \alpha_{z_{ui}}}{\sum_z \left(n_{uz}^{\neg ui} + \alpha_z\right)} \times \frac{n_{us_1}^{\neg ui} + \gamma}{n_{us_0}^{\neg ui} + n_{us_1}^{\neg ui} + \gamma + \gamma'} \tag{13}$$

$$P(s_{ui} = 0|\boldsymbol{s}_{\neg ui}, \boldsymbol{z}, .) \propto \frac{n_{l_{ui}z_{ui}}^{\neg ui} + \alpha'_{z_{ui}}}{\sum_z \left(n_{l_{ui}z}^{\neg ui} + \alpha'_z\right)} \times \frac{n_{us_0}^{\neg ui} + \gamma'}{n_{us_0}^{\neg ui} + n_{us_1}^{\neg ui} + \gamma + \gamma'}, \tag{14}$$

where 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 $s_{ui} = 1$:

$$P(z_{ui}|s_{ui} = 1, \boldsymbol{z}_{\neg ui}, \boldsymbol{v}, \boldsymbol{c}, .) \propto \frac{n_{uz_{ui}}^{\neg ui} + \alpha_{z_{ui}}}{\sum_z \left(n_{uz}^{\neg ui} + \alpha_z\right)} \frac{n_{z_{ui}v_{ui}}^{\neg ui} + \beta_{v_{ui}}}{\sum_v \left(n_{z_{ui}v}^{\neg ui} + \beta_v\right)} \frac{n_{z_{ui}c_{ui}}^{\neg ui} + \beta'_{c_{ui}}}{\sum_c \left(n_{z_{ui}c}^{\neg ui} + \beta'_c\right)}, \tag{15}$$

when $s_{ui} = 0$:

$$P(z_{ui}|s_{ui} = 0, \boldsymbol{z}_{\neg ui}, \boldsymbol{v}, \boldsymbol{c}, .) \propto \frac{n_{l_{ui}z_{ui}}^{\neg ui} + \alpha'_{z_{ui}}}{\sum_z \left(n_{l_{ui}z}^{\neg ui} + \alpha'_z\right)} \frac{n_{z_{ui}v_{ui}}^{\neg ui} + \beta_{v_{ui}}}{\sum_v \left(n_{z_{ui}v}^{\neg ui} + \beta_v\right)} \frac{n_{z_{ui}c_{ui}}^{\neg ui} + \beta'_{c_{ui}}}{\sum_c \left(n_{z_{ui}c}^{\neg ui} + \beta'_c\right)}. \tag{16}$$

After a sufficient number of sampling iterations, the approximated posterior can be used to get estimates of parameters by examining the counts of (s, z) assignments to four-tuple $(u,\ v,\ l,\ c)$. Specifically, during the parameter estimation, the algorithm keeps track of a $K \times |V|$ (topic by spatial item) count matrix, a $K \times |C|$ (topic by content word) count matrix, an $|U| \times 2$ (user by coin) count matrix, an $|U| \times K$ (user by topic) count matrix and an $|L| \times K$ (location by topic) count matrix. Given these matrices, we can estimate the parameters θ, θ', ϕ, ϕ' and λ as follows:

$$\hat{\theta}_{uz} = \frac{n_{uz} + \alpha_z}{\sum_{z'}(n_{uz'} + \alpha_{z'})} \tag{17}$$

$$\hat{\theta}'_{lz} = \frac{n_{lz} + \alpha'_z}{\sum_{z'}(n_{lz'} + \alpha'_{z'})} \tag{18}$$

$$\hat{\phi}_{zv} = \frac{n_{zv} + \beta_v}{\sum_{v'}(n_{zv'} + \beta_{v'})} \tag{19}$$

$$\hat{\phi}'_{zc} = \frac{n_{zc} + \beta'_c}{\sum_{c'}(n_{zc'} + \beta'_{c'})} \tag{20}$$

$$\hat{\lambda}_u = \frac{n_{us_1} + \gamma}{n_{us_1} + n_{us_0} + \gamma + \gamma'}. \tag{21}$$

3. ONLINE RECOMMENDATION

In this section, we present the online recommendation part of our recommender system LCARS. Given a querying user u with a querying location l_u to which u is going to travel,

the online part of LCARS will efficiently compute ranking scores for all spatial items within l_u and then return the top-k items as the recommendation results.

3.1. Ranking Score Computation for Spatial Items

The ranking scores of spatial items are computed using the knowledge, such as user interest θ, local preference θ', mixing weight λ, topics ϕ and ϕ', learned by the offline model LCA-LDA. To improve the online query performance, we propose a ranking framework in Equation (22) 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 topic z which is learned 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 preference weight of query $(u,\ l_u)$ on topic z. It is worth mentioning that the main time-consuming components of $W(u, l_u, z)$ are also computed offline (e.g., $\hat{\theta}_{uz}$, λ_u and $\hat{\theta}'_{l_u z}$), and the online computation is just a simple linear fusion process, as is shown in Equation (23). This design enables maximum precomputation for the problem considered, and in turn minimizes the query time. At query time, the ranking score $S(u, l_u, v)$ in Equation (22) only needs to aggregate $F(l_u, v, z)$ over K topics by a simple weighted sum function, in which the weight is $W(u, l_u, z)$. From Equations (23) and (24), 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 cooccurrence information and the similarity of item contents to produce recommendations.

$$S(u, l_u, v) = \sum_z F(l_u, v, z) W(u, l_u, z) \tag{22}$$

$$W(u, l_u, z) = \hat{\lambda}_u \hat{\theta}_{uz} + (1 - \hat{\lambda}_u) \hat{\theta}'_{l_u z} \tag{23}$$

$$F(l_u, v, z) = \begin{cases} \hat{\phi}_{zv} \sum_{c_v \in C_v} \hat{\phi}'_{z c_v} & v \in V_{l_u} \\ 0 & v \notin V_{l_u}. \end{cases} \tag{24}$$

3.2. TA Algorithm for Top-k Recommendation

The straightforward method of generating the top-k items needs to compute the ranking scores for all items according to Equation (22), which is computationally inefficient, especially when the number of items becomes large (e.g., millions of items). To speed up the online process of producing recommendations, we extend the Threshold-based Algorithm (TA) [Fagin et al. 2001], which is capable of finding the top-k results by examining the minimum number of items.

We first partition all spatial items into locations at a predefined granularity 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 (24). Given K topics, we carry out this procedure for each topic z (i.e., having spatial items along with their scores on the same topic z in each sorted list L_z). When receiving a query $q = (u, l_u)$, we first obtain K ranked lists L_z, $z \in \{1, 2, \ldots, K\}$, of spatial items in location l_u, and compute the query preference weights $W(u, l_u, z)$ on each topic z, $z \in \{1, 2, \ldots, K\}$. We then run the Algorithm 2 to compute the top-k spatial items from the K lists and return them (along with their ranking scores) in the priority list L. As is shown in Algorithm 2, we first maintain a priority list PL for the K lists where the priority of a list L_z is the

ALGORITHM 2: Threshold-based algorithm

Input: A query $(u,\ l_u)$; the query preference weights $W(u, l_u, z), z \in \{1, 2, \ldots, K\}$; priority lists $(L_1, \ldots, L_K)$;
Output: List L with all the k highest ranked spatial items;

```
/* PL, L and L_z are priority lists in which elements are automatically sorted
   according to their priorities. They have five operations:
   insert(element,priority) inserts an element into the list with a specific
   priority; get() returns the head element; remove() removes the head element;
   get(k) returns the k-th element; remove(k) removes the k-th element;
   hasMore() returns true if the list is non-empty.                          */
PL, L = ∅;
/* initialize the threshold score S_Ta                                        */
S_Ta = max;
for z = 1 to K do
    v = L_z.get();
    Compute S(u, l_u, v) according to Equation (22);
    PL.insert(z, S(u, l_u, v));
end
S_Ta = Compute_Threshold();
while true do
    nextListToCheck = PL.get();
    PL.remove();
    v = L_nextListToCheck.get();
    L_nextListToCheck.remove();
    if v ∉ 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();
        Compute S(u, l_u, v) according to Equation (22);
        PL.insert(nextListToCheck, S(u, l_u, v));
        S_Ta = Compute_Threshold();
    end
    else
        break;
    end
end
return L;
```

ALGORITHM 3: Function Compute_Threshold()

Input: A priority list PL; priority lists $(L_1, \ldots, L_K)$; the query preference weights $W(u, l_u, z)$, $z \in \{1, 2, \ldots, K\}$;
Output: The threshold score S_{Ta};

```
1 S_Ta = 0;
2 for i = 1 to K do
3 |   z = PL.get(i);
4 |   v = L_z.get();
5 |   S_Ta = S_Ta + W(u, l_u, z)F(l_z, v, z);
6 end
7 return S_Ta;
```

ranking score (i.e., $S(u, l_u, v)$) of the first item v in L_z (Lines 3–7). In each iteration, we select the most promising item (i.e., the first item) from the list that has the highest priority in PL and add it to the result list L (Lines 10–17). When the size of L is no less than k, we will examine the k-th item in the result list L. If the ranking score of the k-th item is no less than the *threshold score* (i.e., S_{Ta}), which is computed in Algorithm 3, the Threshold-based Algorithm terminates early without checking any subsequent items (Lines 19–22). Otherwise, the k-th item is either replaced by the current item if its ranking score is lower than that of the current one, or reserved if otherwise (Lines 23–26). In the end of each iteration, we update the priority of current list as well as the threshold score (Lines 29–34).

Algorithm 3 illustrates the computation of the threshold score, which is obtained by aggregating the maximum $F(l_u, v, z)$ represented by the first item in each list L_z. Consequently, it is the maximum possible ranking score that can be achieved by remaining unexamined items. Hence, if the ranking score of the k-th item in the result list L is no less than the threshold score, L can be returned immediately because no remaining item will have a higher ranking score than the k-th item.

3.2.1. Discussion. It is easy to understand that Algorithm 2 is able to correctly find the top-k items with the monotone aggregation function $S(u, l_u, v)$ defined in Equation (22). We will now prove it formally.

THEOREM 1. *Algorithm 2 is able to correctly find the top-k items with the monotone aggregation function $S(u, l_u, v)$ defined in Equation (22).*

PROOF. Let L be a ranked list returned by Algorithm 2 which contains the k spatial items that have been seen with the highest ranking scores. We only need to show that every item of L has a ranking score at least as high as any other item v not in L. By definition of L, this is the case for each item v that has been seen in running Algorithm 2. So assume that v was not seen, and the score of v in each topic z is $F(l_u, v, z)$. For each ranked list L_z, let $\widetilde{v}_z$ be the last item seen in the list. Therefore, $F(l_u, v, z) \leq F(l_u, \widetilde{v}_z, z)$, for every z. Hence, $S(u, l_u, v) \leq S_{Ta}$ where S_{Ta} is the threshold score computed in Algorithm 3. The inequality $S(u, l_u, v) \leq S_{Ta}$ holds because of the monotonicity of the aggregation function $S(u, l_u, v)$ defined in Equation (22). But by definition of L, for every v' in L we have $S(u, l_u, v') \geq S_{Ta}$. Therefore, for every v' in L we have $S(u, l_u, v') \geq S_{Ta} \geq S(u, l_u, v)$, as desired. □

Besides, Algorithm 2 has another nice property that it is instance optimal with accessing the minimum number of items, and no deterministic algorithm has a lower optimality ratio [Fagin et al. 2001]. We use the word "optimal" to reflect the fact that Algorithm 2 is best deterministic algorithm. Intuitively, instance optimality

corresponds to optimality in every instance, as opposed to just worst case or the average case. There are many algorithms that are optimal in a worst-case sense, but are not instance optimal.

Next, we will investigate the instance optimality of Algorithm 2 by an intuitive argument. If A is an algorithm that stops earlier than Algorithm 2 in a certain case, before A finds k items whose ranking score is at least equal to the threshold score S_{Ta}, then A must make a mistake, since the next unseen item v might have a ranking score equal to $F(l_u, \widetilde{v}_z, z)$ in each topic z, and hence have ranking score $S(u, l_u, v) = S_{Ta}$. This new item, which A has not even seen, has a higher ranking score than some item in the top-k list that was output by A, and so A erred by stopping too soon.

3.3. TA-Approximation Algorithm for Top-k Recommendation

In practice, the querying user u may be satisfied with an *approximation* top-k list. Assume $\rho > 1$, a *ρ-approximation* to the top-k answers for the aggregation function $S(u, l_u, v)$ is defined to be a list of k items (each along with its ranking score) such that for each v' among these k items and each v not among these k items, $\rho S(u, l_u, v') \geq S(u, l_u, v)$. Note that the same definition with $\rho = 1$ gives the exact top-k answers. We can modify Algorithm 2 to find a *ρ-approximation* to the top-k answers by modifying the rule in Line 20 to "if $S(u, l_u, v') \geq S_{Ta}/\rho$", namely as soon as at least k items have been seen whose ranking score is at least equal to S_{Ta}/ρ, then halt. Let us call this approximation algorithm TA-ρ.

THEOREM 2. *The TA-ρ algorithm is able to correctly finds a* ρ-approximation *to the top-k answers for the monotone aggregation function $S(u, l_u, v)$ defined in Equation* (22).

PROOF. This follows from a straightforward modification of the proof of Theorem 1. □

ALGORITHM 4: Interactive Process

```
if S(u, l_u, v') ≥ S_Ta/ρ then
    Show the current view of the top-k list L to the querying user;
    Get feedback from the querying user;
    if the querying user is satisfied with L or ρ is equal to 1 then
        break;
    end
    else
        ρ = ηρ;
        /* η is a predefined decay rate with the range (0 ··· 1)        */
    end
end
```

Furthermore, we can easily modify Algorithm 2 into an interactive process where at all times LCARS can show the querying user its current view of the top-k recommendations along with a guarantee of ρ-approximation to the correct answer. Specifically, we can modify the rule in Lines 20–21 "if $S(u, l_u, v') \geq S_{Ta}$ then break;" to an interactive process illustrated in Algorithm 4. Thus, the querying user can decide whether he/she is satisfied with the current recommendation results and can stop the algorithm early at any time when the condition $S(u, l_u, v') \geq S_{Ta}/\rho$ is met. An example is shown in Figure 4. The first results shown in the left column are unsatisfied, and the user clicks the "dislike". The system next reduces the ρ and recomputes the results shown in the right column. The querying user is satisfied with the current recommendations, and

Fig. 4. An example of interactive process.

the algorithm can terminate. Otherwise, the algorithm will continue to improve the approximation degree by decaying ρ value, and produce more accurate recommendations until the querying user's expectations are met, or the exact top-k recommendations are produced (i.e., the ρ is decayed to one).

4. EXPERIMENTS

In this section, we first describe the settings of experiments including the datasets, 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 their tradeoff. We also study the interpretability of our LCARS by analyzing the learned user profiles, the effect of users' personal interests and the local preferences in users' decision making for travelling, and the latent topics learned by LCARS.

4.1. Experimental Settings

4.1.1. Datasets. In this article, we utilize one synthetic dataset with 10 million spatial items and two real-life datasets for the performance evaluation. The detailed description for two real datasets are listed as follows.

—*DoubanEvent*. 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 checking-in online. This dataset consists of 100,000 users, 300,000 events and 3,500,000 check-ins. Most of check-in records are located in China's four largest cities: Beijing, Shanghai, Guangzhou and Shenzhen. To guarantee the validity of the experimental results, each user in our dataset has provided at least 10 check-ins. Figure 5 describes the distribution information of both users and events over cities. For instance, 22% of users live in the city of Beijing and 24% of events are held in Beijing. 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,

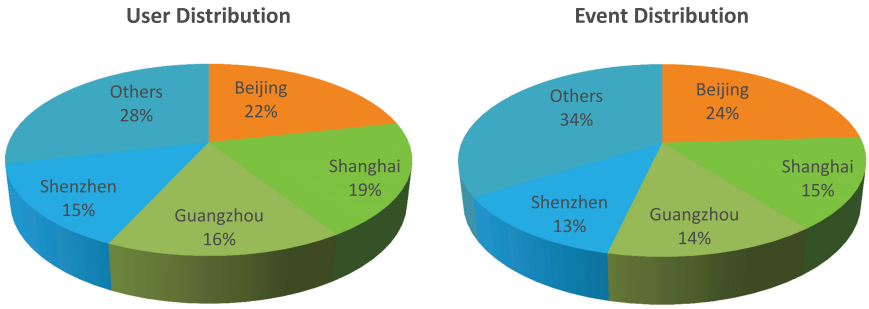


Fig. 5. User and event distributions over cities.

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].

—*Foursquare*. Another publicly available LBSNs dataset, Foursquare [Gao et al. 2012], is also used in our experiment. Foursquare is one of the most popular online LBSNs. It has more than 30 million users and 3 billion check-ins as of January, 2013[2]. The web site itself does not provide a public API to access users' check-in data; however, it provides an alternative way for users to link their twitter accounts with Foursquare, and then share the check-in message as tweets to Twitter. Previous works [Gao et al. 2012; Scellato et al. 2011b; Bauer et al. 2012; Cheng et al. 2011] used this way to collect the data from Twitter for studying check-in behaviors. Similarly, the dataset used in our experiment was collected by getting access to the check-in tweets through the Twitter REST API from January 2011 to July 2011. This dataset contains 11, 326 users, 182, 968 venues and 1, 385, 223 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[3] to partition all the spatial items into cities according to their latitudes and longitudes. 2) For the DoubanEvent dataset, we then use NLP toolkits[4] to extract a set of content words for each event from its summary and category description. To guarantee the quality of content words, we use tf-idf techniques to rank all content words associated with each event and finally keep top five ranked ones.

Note that, although we only utilize the city granularity to generate recommendation to end-users for evaluation in this article, our approach can be easily extended to facilitate the recommendation task at various granularities, by dividing the space into multi-scale regions, inferring their local preferences offline, and automatically selecting proper region when making recommendations. Specifically, we can use a spatial tree structure, for instance, pyramid structure [Sarwat et al. 2013; Aref and Samet 1990], to partition and index the space. Thus, the space can be divided recursively into numerous cells at different levels with different granularities. After the space partition, we use LCA-LDA model to learn the local preference θ'_l for each cell l. When receiving a query, LCARS searches the index to find a suitable granularity, and produces top-k recommendations. If the querying user changes the granularity of the querying region, LCARS can simply traverse the index and re-compute top-k recommendations. It is

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

worth mentioning that the local preferences can be constructed offline, which will not affect the efficiency of online recommendation. Thus, LCARS can also support another recommendation scenario where end-users do not need to input querying locations explicitly, and LCARS can automatically locate them by the interfaces of location-based services such as Google Maps and Yelp, and then find the regions with proper granularity.

4.1.2. 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 [Ye et al. 2011], 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 user-based 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.

—*Social Trust Ensemble (STE)*. Social Trust Ensemble, proposed in [Ma et al. 2009], is a probabilistic matrix factorization framework which linearly fuses the users' tastes and their friends' favors together to produce recommendations. It should be noted that the mixing weights in STE are manually set rather than learned automatically from the data. Besides, the mixing weights in BTE are not personalized (i.e., all users in a dataset share the same mixing weights), ignoring the differences between users.

—*Category-based k-Nearest Neighbors Algorithm (CKNN)*. Following recently proposed location-based recommendation technique [Bao et al. 2012] for dealing with the problem of data sparsity, a category-based KNN algorithm is implemented as our competitor. CKNN first projects a user's activity history into a well-designed category space and models each user's preference with a weighted category hierarchy. Meanwhile, it infers the authority of each user in a city with respect to different category of spatial items according to their activity histories using HITS model [Kleinberg 1999]. When receiving a query $q = (u, l)$, CKNN first selects a set of high-quality users N_u in the querying city who have the same or similar preferences with the querying user u. Then,CKNN constructs a user-item matrix using the selected users N_u and their visited spatial items. Finally, CKNN employs a traditional user-based CF model over the user-item matrix to infer the querying user's rating of a candidate item. The general intuition behind a user-based CF model is that similar users rate the same items similarly. Formally, the rating that the querying user u would give to spatial item v is calculated as follows.

$$S(u, v) = \sum_{u' \in N_u} Sim(u, u') \times r(u', v) \tag{25}$$

where $Sim(u, u')$ denotes the similarity between u and u' which is computed according to their weights in the category hierarchy rather than the traditional Cosine value between two users' item vectors; $r(u', v)$ represents the rating that u' gave to item v.

—*Item-based k-Nearest Neighbors Algorithm (IKNN)*. IKNN is the most common way that people come up with, which applies the collaborative filtering method directly over the spatial items. 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 in the querying city by computing the Cosine similarity between the

querying user's and other users' item vectors. Finally, the spatial items in the user-specific querying city that have a relatively high ranking score will be recommended. It should be noted that when IKNN cannot help the querying user find k nearest neighbors in the querying city, we recommend the most popular local ones.

—*LDA*. Following previous works [Jin et al. 2005; Chen et al. 2009], a standard LDA-based method is implemented as one of our baselines. In this model, each user is viewed as a document, and spatial items visited by the user are viewed as the words appeared in the document. 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 (22) 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 (22) where $F(l_u, v, z) = \hat{\phi}_{zv}$ and $W(u, l_u, z) = \hat{\lambda}_u \hat{\theta}_{uz} + (1 - \hat{\lambda}_u)\hat{\theta}'_{l_u z}$.

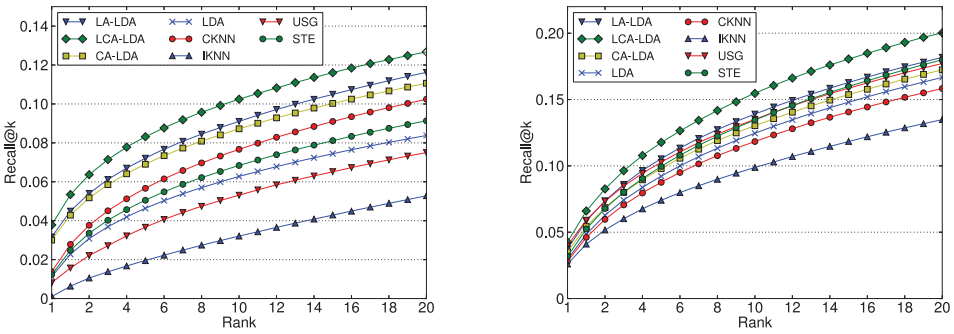
—*Content-Aware LDA (CA-LDA)*. As another component of the LCA-LDA model, CA-LDA means our method without exploiting the location information of spatial items, that is, local preference. It can capture the prior knowledge that spatial items with the same or similar contents are more likely to belong to the same topic. This model is similar to the ACT model [Tang et al. 2008] in the methodology. For online recommendation, the ranking score is computed using our ranking framework in Equation (22) where $F(l_u, v, z) = \hat{\phi}_{zv} \sum_{c_v \in C_v} \hat{\phi}'_{zc_v}$ and $W(u, l_u, z) = \hat{\theta}_{uz}$.

4.1.3. Evaluation Methods. We evaluate both the effectiveness of the suggested recommendations and the efficiency for generating online recommendations as well as their tradeoff in our LCARS.

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 designed dividing strategies, we split the user activity history S into the training dataset $S_{training}$ and the test set S_{test}. To evaluate the recommender models, we adopt the similar testing methodology and the measurement Recall@k applied in Cremonesi et al. [2010], Chen et al. [2009], Koren [2008], and Yin et al. [2012]. Specifically, for each test case $(u,\ v,\ l_v)$ in S_{test}, we follow this methodology.

(1) We compute the ranking score for item v as well as all other spatial items located in city l_v and unvisited by u before.
(2) We form a ranked list by ordering all these 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 the test item v precedes all the unvisited items (i.e., $p = 0$).
(3) 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



(a) Users Traveling in New Cities (b) Users Traveling in Home Cities

Fig. 6. Top-k performance on DoubanEvent.

we have a miss. The probability of a hit increases with the increasing value of k. When k is equal to the number of spatial items located in l_v, 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:

$$Recall@k = \frac{\#hit@k}{|S_{test}|}, \tag{26}$$

where $\#hit@k$ denotes the number of hits in the test set, and $|S_{test}|$ is the number of all test cases. It should be noted that both DoubanEvent and Foursquare datasets have a low density, which usually results in relatively low recall values [Yuan et al. 2013; Konstas et al. 2009]. In addition, the spatial items visited by user u may represent only a small portion of spatial items that u is interested in, so the hypothesis that all the unvisited spatial items are non-relevant to user u tends to underestimate the computed recall with respect to the true recall. However, this experimental setting and evaluation are fair to all comparison approaches. So, we focus on the relative improvements we achieve, instead of the absolute values in this article.

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. At the same time, we explore the benefit that our designed TA-based query processing technique brings to the online recommendation part.

Trade-off between Recommendation Effectiveness and Efficiency. We also study the tradeoff between recommendation effectiveness and recommendation efficiency in our LCARS using TA-ρ algorithm by varying ρ values.

4.2. Recommendation Performance of LCARS

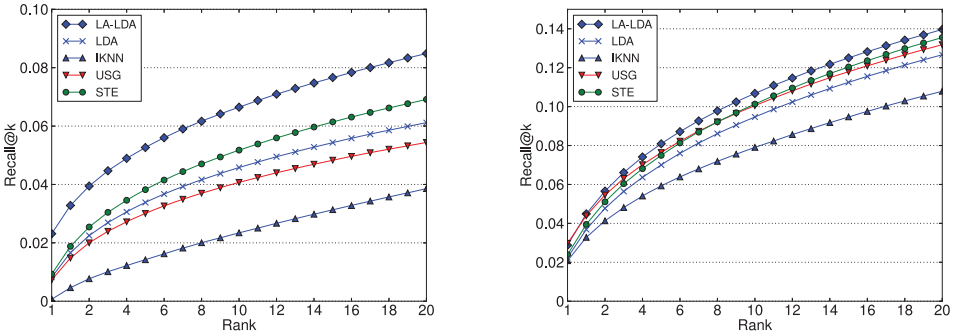
In this subsection, we first report the performance of our LCARS on the recommendation effectiveness and then compare the time costs of different recommendation algorithms.

4.2.1. 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 6 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 6(a) where querying cities are new cities, the recall of LCA-LDA is about 0.1 when $k = 10$, and 0.126 when $k = 20$ (i.e., the model has a probability of 10% of placing an appealing event within the querying city in the top-10 and 12.6% of placing it in the top-20). Clearly, our proposed LCA-LDA model outperforms other competitor recommendation algorithms significantly. First, IKNN and USG perform the worst in the new city setting. Both of them are traditional location-based recommendation algorithms and cannot alleviate the data sparsity problem in new cities. Specifically, without exploiting the content/category information of spatial items to build a bridge, they cannot transfer the users' preferences learned in home cities to new cities, and hence fail to find k preference-similar users for the querying user in the new city setting. Second, STE and USG perform better than LDA and IKNN, respectively, due to the benefits brought by considering social influence from friends. Besides, STE and LDA outperform both USG and IKNN consistently, showing the advantages of latent factor models which overcome the data sparsity problem, to some extent, by dimension reduction. But STE performs worse than CKNN and our LCA-LDA, which shows that exploiting social influences and dimension reduction are not enough to alleviate the *new city* problem although they can alleviate the data sparsity problem to some extent. As is analyzed in Cho et al. [2011], most of a querying user's friends live in the same city with the querying user, and they also have few footprints in the querying city that is new to the querying user due to the property of travel locality [Sarwat et al. 2013; Levandoski et al. 2012]. That is why exploiting social and geographical influence cannot help much when alleviating the *new city* problem. Third, CKNN, which was proposed for solving the *new city* problem [Bao et al. 2012], performs better than STE, IKNN, USG and LDA, as is expected. CKNN depends on a well-designed category hierarchy to facilitate users' preferences across cities. So, it can find k high-quality users who have similar/same preferences with the querying user. But, this method ignores the observation of the shift of users' preferences, that is, people's preferences or behavioral patterns may change when they travel in different cities, especially in cities that are new to them. So, CKNN would fail to make accurate recommendations in the case where users' preferences shift, while our proposed LCA-LDA and LA-LDA models can still work well in this case because they exploit the local preferences/attractions of the querying city to produce recommendations, that is, what most of people have done when they travel in the querying city. That is why our proposed LCA-LDA model performs much better than CKNN. Fourth, LA-LDA outperforms LDA, justifying the benefit brought by considering local preferences, and 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 6(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 6(a). The main differences are that 1) all recommendation algorithms perform better in the home city setting than in the new city setting and 2) the performance gaps between different recommendation methods narrow, because the data sparsity problem is not so severe in the home city setting. Another observation is that USG and STE 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 social and geographical influences. However, the performances of USG and STE are not so well in the new city setting, as shown in Figure 6(a), which shows that exploiting social and geographical influences is not enough to alleviate the *new city* problem although it can



(a) Users Traveling in New Cities (b) Users Traveling in Home Cities

Fig. 7. Top-k performance on Foursquare.

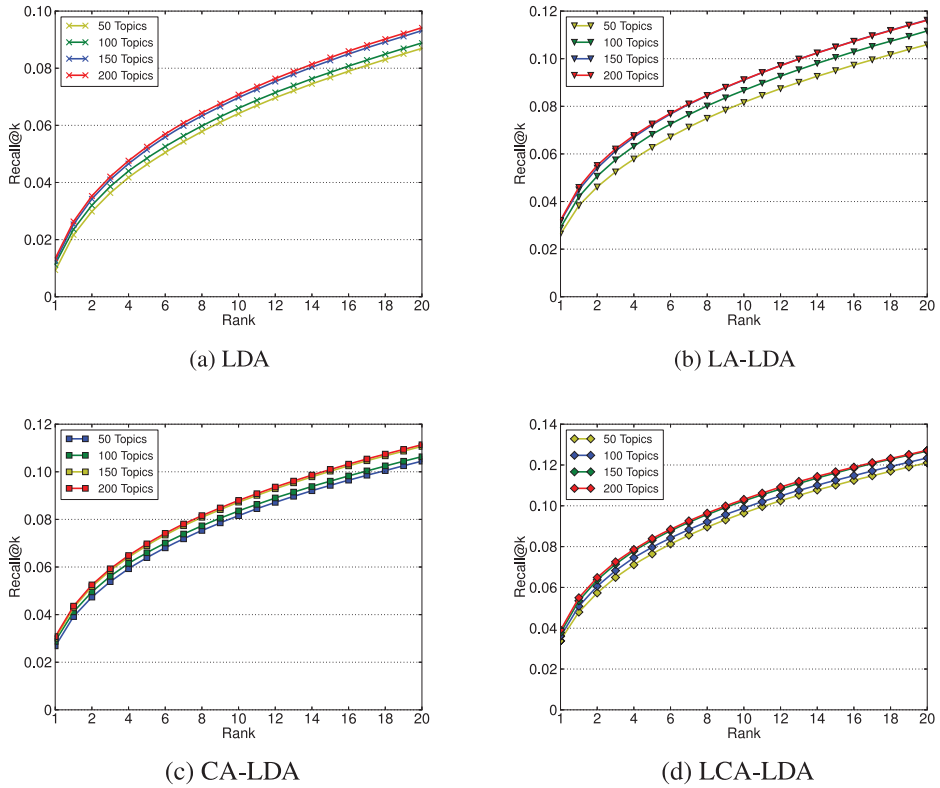
alleviate the data sparsity problem to some extent. The third observation is that CKNN outperforms LDA and STE in Figure 6(a) while LDA and STE slightly exceeds CKNN in Figure 6(b), showing that existing latent factor models (e.g., LDA and STE) better suit the home city setting which is almost the same as the traditional recommendation setting (e.g., movie recommendation), and the hybrid methods (e.g., CKNN) are more capable of overcoming the difficulty of data sparsity, that is, the *new city* problem.

Figure 7 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, STE, 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 6, 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. We only show the experimental results for the new city setting since the experimental results for the home city setting is similar.

As for the hyperparameters α, α', β, β', γ and γ', following existing works [Tang et al. 2008, 2012], 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 (e.g., $K = 50$, 100, 150, 200.) and present the results in Figure 8(a) to 8(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. It should be noted that the performance reported in Figure 6 is achieved with 150 latent topics.

4.2.2. Efficiency and Scalability 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 are different numbers



(a) LDA (b) LA-LDA

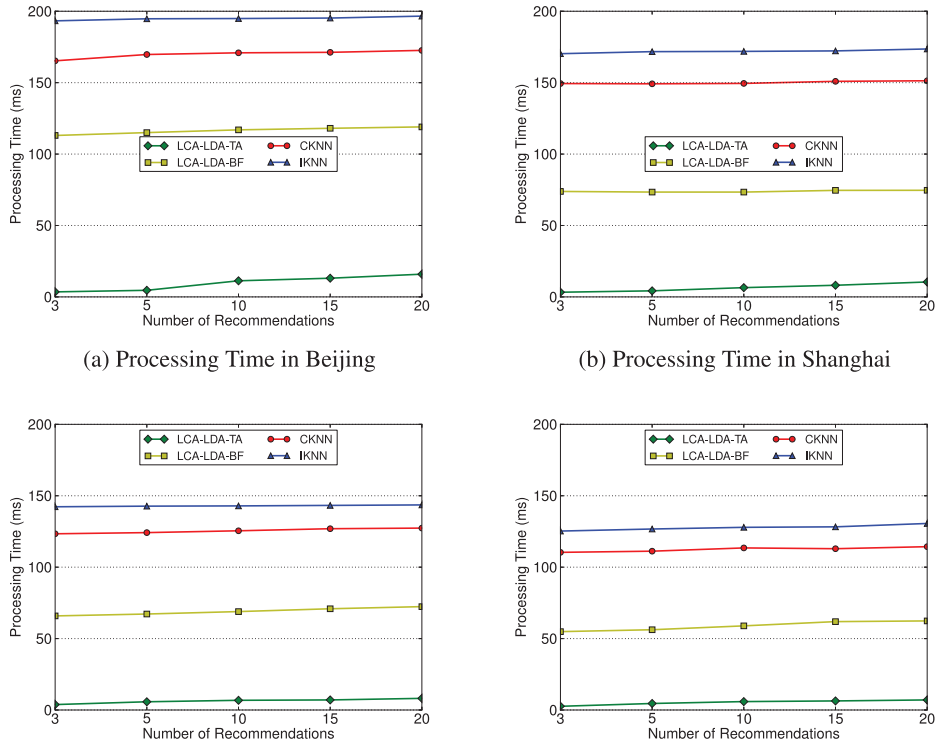
(c) CA-LDA (d) LCA-LDA

Fig. 8. Impact of the number of latent topics.

of events and users in these four cities (i.e., $|V_{Beijing}| > |V_{Shanghai}| > |V_{Guangzhou}| > |V_{Shenzhen}|$, $|U_{Beijing}| > |U_{Shanghai}| > |U_{Guangzhou}| > |U_{Shenzhen}|$). All the recommendation algorithms were implemented in Java (JDK 1.6) and run on a Linux Server with 64G RAM.

For the online recommendation of LCARS, we adopt two methods to utilize the knowledge learned offline by LCA-LDA to produce recommendations. The first is called LCA-LDA-TA proposed in Section 3.2, 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 among them to recommend to the test user.

Figure 9(a)–9(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 produce top-10 recommendations in 11.4 ms, 6.7 ms, 6.1 ms and 5.4 ms for the querying cities Beijing, Shanghai, Guangzhou and Shenzhen, respectively. 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-based query processing technique; 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 learned 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:



(a) Processing Time in Beijing

(b) Processing Time in Shanghai

(c) Processing Time in Guangzhou

(d) Processing Time in Shenzhen

Fig. 9. Efficiency w.r.t recommendations.

$TS_{Beijing} > TS_{Shanghai} > TS_{Guangzhou} > TS_{Shenzhen}$, which is, however, due to different causes for different algorithms. The time costs of LCA-LDA-TA and LCA-LDA-BF mainly depend on the number of items while that of CKNN and IKNN mainly depend on the number of users.

To evaluate the scalability of LCARS, we create a large-scale synthetic dataset containing 1 million users, 10 million spatial items and 100 million check-ins in total. Each spatial item is associated with a city and 5 content words. We control the number of available spatial items to vary from 1 million to 10 millions to test the scalability of LCARS, since the online time cost of LCARS mainly depends on the number of available spatial items, given a query and the number of recommendations. Note that we do not compare LCARS with CKNN and IKNN in this experiment since the runtime costs of CKNN and IKNN mainly depend on the number of available users rather than items. Given a query, CKNN and IKNN need to scan all available users and compute their similarity with the querying user to find k nearest neighbors in the querying city. As shown in Figure 9, LCA-LDA-TA performs better than them significantly.

Figure 10 presents the time cost of producing top-10 recommendations with varying number of available items from 1 million to 10 millions. From the figure, we can see that both algorithms exhibit highly desirable scaling characteristics—linear in the amount of available items to recommend. This is confirmed by a linear regression applied to the running time data, which yields an R^2 value close to one. Results also demonstrate that LCA-LDA-TA is much faster than LCA-LDA-BF: 170 ms compared to 1660 ms when the number of available items is 10 million (improvement by a factor of 9.76).

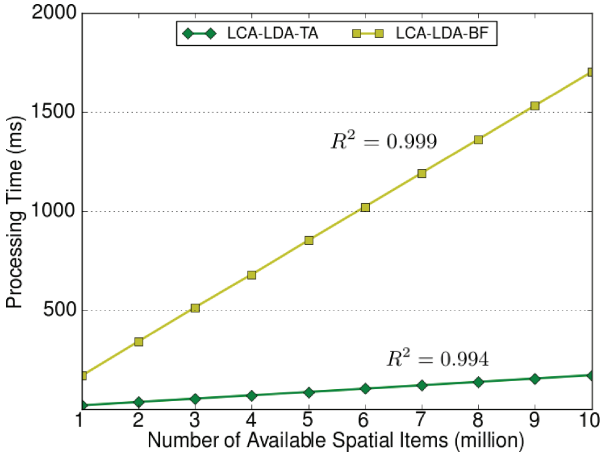
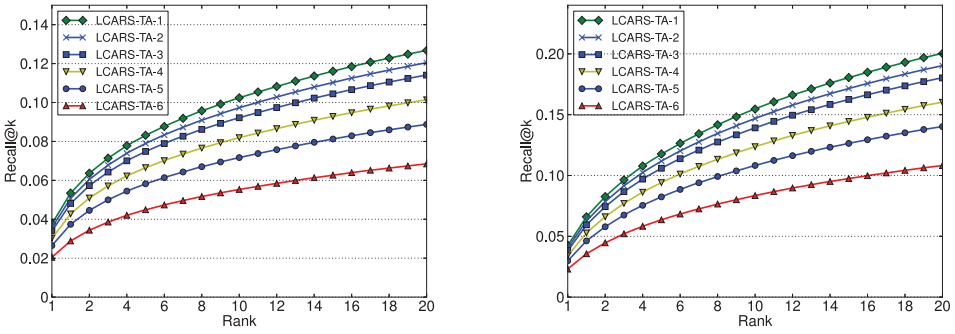


Fig. 10. Scalability of LCARS on the synthetic dataset.

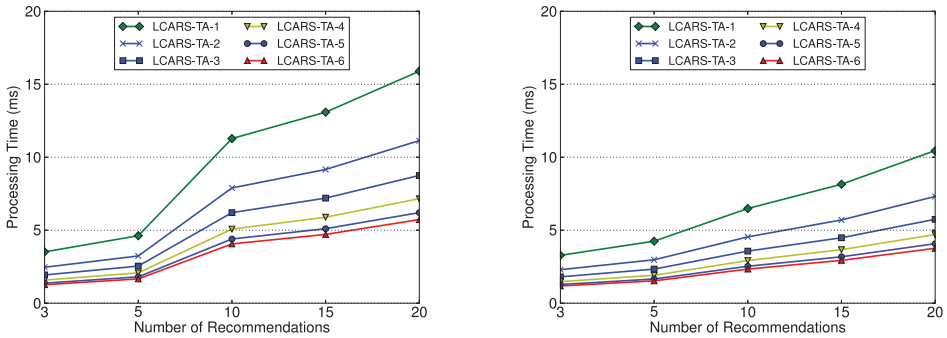


(a) Users Traveling in New Cities (b) Users Traveling in Home Cities

Fig. 11. Recommendation effectiveness of LCARS on DoubanEvent dataset, using TA-ρ algorithm.

4.3. Performance of LCARS with TA-Approximation Algorithm

In this subsection, we study the performance of LCARS with TA-approximation algorithm, varying the ρ values from 1 to 6. Figure 11 and 12 show the recommendation effectiveness and efficiency of LCARS with TA-ρ algorithm by varying the ρ values from 1 to 6, respectively. From the figures, we observe that the recommendation effectiveness decreases with the increasing ρ value while the recommendation efficiency increases with the increasing ρ value, that is, the time cost of online recommendation decreases with the increasing ρ value. Note that LCARS-TA-1 in Figures 11 and 12 is the LCARS with exact TA algorithm. By comparison between Figures 11 and 12, we can see that there is a tradeoff between recommendation effectiveness and efficiency. Users can specify the value of ρ according to their needs, higher recommendation accuracy or faster query response. Fortunately, the TA-approximation algorithm can reduce the time cost of online recommendation drastically (i.e., about 50%) at a little cost of recommendation accuracy when $\rho \leq 3$. Although the benefit of 50% time cost reduction is small in our case study (e.g., several ms), it would become greatly significant when the number of spatial items available is huge (e.g., millions of items).



(a) Processing Time in Beijing (b) Processing Time in Shanghai

Fig. 12. Recommendation efficiency of LCARS on DoubanEvent, using TA-ρ algorithm.

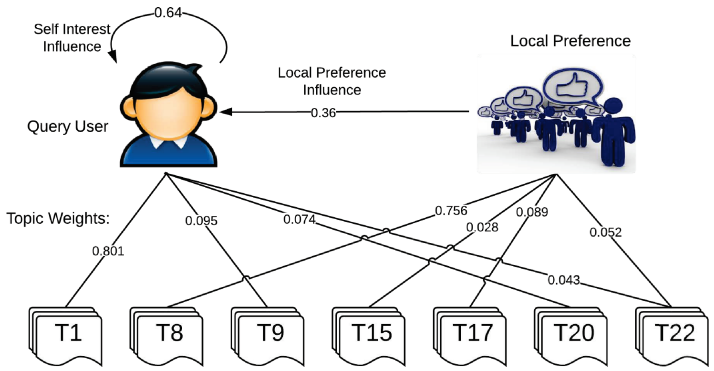


Fig. 13. Example user profile and local preference influence learned from DoubanEvent, by using LCA-LDA.

4.4. Profile Study

Both the personal interest of querying user u, the local preference of the querying city l_u and their influences to u's decision-making can be learned automatically through our LCA-LDA model to build user profiles. In this section, we first analyze a user profile to facilitate a better understanding of the user's visiting behaviors. In Figure 13, we show the profiles of user 102 who comes from Shanghai and the querying city Beijing. As shown, user 102 is influenced by the local preference with influence probability value 0.36 . Also, the top-4 topics of the user's interest and the local preference of Beijing are also shown respectively, where the weights representing user's personal interest and the local preference in the topics are labeled in the corresponding edges. Notice that there is only one overlapped topic for user 102 and the local preference of Beijing, and their dominated topics are different (i.e., T1 vs. T8).

We also show two location profiles for two largest cities in China, Beijing and Shanghai, in Figure 14 where top-4 topics for each city are presented. By comparing the two location profiles, we can observe that there is only one overlapped topic between Beijing's local preference and Shanghai's local preference. This indicates the phenomenon of preference locality which suggests that users from a spatial region prefer spatial items that are manifestly different from spatial items preferred by users from other regions. This observation is consistent with the principle of homophily [Wasserman and Faust 1994] in social network studies – birds of a feather flock together. Note that we show the contents of topics T8, T9 and T22 in Table II.

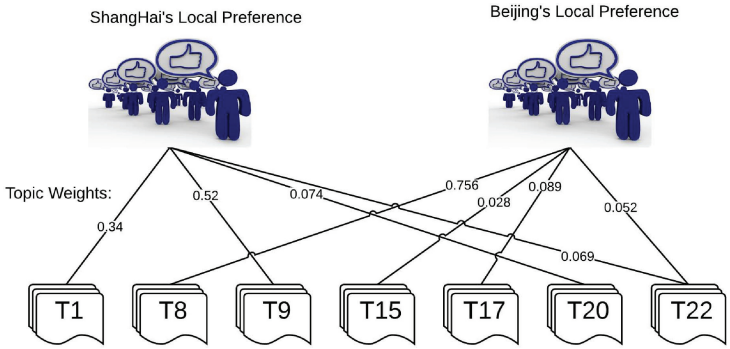
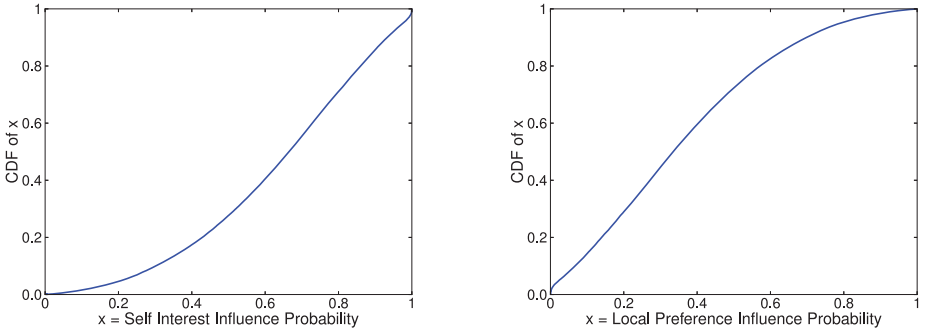


Fig. 14. Example location profiles learned from DoubanEvent, by using LCA-LDA.

Table II. Latent Topics Learned by LCA-LDA

T8			T22			T9		
Event ID	Category	Location	Event ID	Category	Location	Event ID	Category	Location
18852778	culture salon	Beijing	14232509	exhibition	Beijing	18567203	concert	Shanghai
11177738	culture salon	Shanghai	18833193	exhibition	Shanghai	18131435	concert	Beijing
18845712	culture salon	Nanjing	18761132	exhibition	Beijing	18898584	concert	Shanghai
18833831	culture salon	Beijing	18619185	exhibition	Xi'an	18825734	concert	Shanghai
18129058	culture salon	Wuhan	18818656	exhibition	Shanghai	18710070	concert	Guangzhou
18840452	culture salon	Beijing	18696716	exhibition	Beijing	18465268	concert	Chengdu
18867591	culture salon	Guangzhou	18800412	exhibition	Beijing	18631346	concert	Beijing
18953054	culture salon	Beijing	12104434	exhibition	Shanghai	18394935	concert	Shanghai

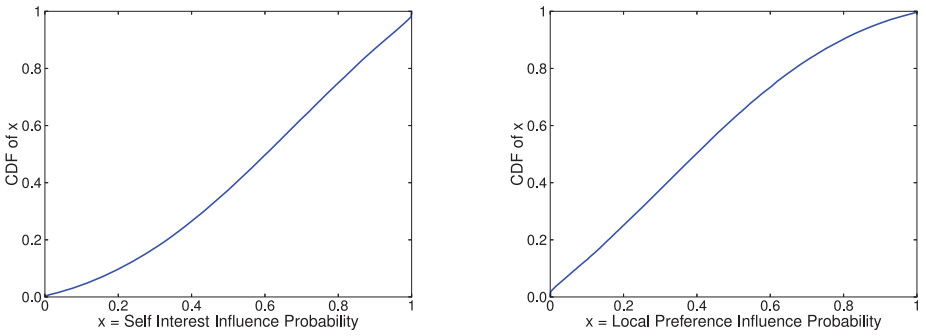


(a) Self Interest Influence (b) Local Preference Influence

Fig. 15. Local preference influence result (DoubanEvent).

4.5. 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 learned 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 dataset are shown in Figure 15, where Figure 15(a) plots the cumulative distribution of self interest influence probabilities, and Figure 15(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,



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

Fig. 16. Local preference influence result (Foursquare).

Table III. Latent Topics Learned by LDA

T9			T8			T16		
Event ID	Category	Location	Event ID	Category	Location	Event ID	Category	Location
18825734	concert	Shanghai	18852778	culture salon	Beijing	18020482	exhibition	Guangzhou
18830050	film	Shanghai	18840452	culture salon	Beijing	18425473	concert	Guangzhou
18818656	exhibition	Shanghai	18629384	film	Beijing	18847061	culture salon	Guangzhou
16578267	film	Shanghai	18432390	drama	Beijing	18937837	party	Guangzhou
18567203	concert	Shanghai	18668341	concert	Beijing	18847604	film	Guangzhou
17364244	drama	Shanghai	18041992	exhibition	Beijing	18829026	concert	Guangzhou
18053337	concert	Shanghai	18953054	culture salon	Beijing	18412853	drama	Guangzhou
13892914	culture salon	Shanghai	18478314	drama	Beijing	17364134	concert	Guangzhou

Figure 15(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, but 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 4.2.1).

Figure 16(a) and 16(b) show, respectively, the self interest influence probabilities and local preference influence probabilities learned from the Foursquare data by LA-LDA model. We observe that the trend of the CDF curve in Figure 16 is similar to that in Figure 15. As shown in Figure 16, although the self-interest influence probability is generally higher than that of the local preference, the local preference still plays an important role in the user decision-making for visiting. For example, the local preference influence probability of about 40% of users is higher than 0.5. This finding also shows the necessity of exploiting the local preference in spatial item recommendation.

4.6. Analysis of Latent Topic

To analyze why our proposed location content-aware probabilistic generative model LCA-LDA performs better than LDA in the task of top-k recommendation, especially spatial item recommendation in new cities, we investigate the latent information learned from LCA-LDA and LDA.

Table II and Table III respectively show three latent topics (e.g., T8, T22, and T9) learned by LCA-LDA and LDA on the DoubanEvent dataset. For each topic, we present the top eight events with the highest probabilities, including their IDs,

categories and locations. For locations, we present only the cities rather than detailed addresses because of space constraints. Each event can be browsed online by combining event-ID and the prefix URL[5]. For example, the event 18852778 can be accessed at www.douban.com/event/11177738/. By comparing the topics in Tables II and III, we observe that the events in each latent topic learned by LCA-LDA not only share the same category, but are also located in different cities. In contrast, the topics learned by LDA are not category-consistent. For example, concerts, culture salons, films and exhibitions are mixed up in the topics T9, T8 and T16 learned by LDA. Besides, the events in each topic learned by LDA take place in the same city. For example, the events in topic T9 are located in Shanghai and the events in topic T8 are held in Beijing.

The comparative results reveal that when we use existing topic model LDA to analyze the user activity history, we are unable to discover the users' interests in the features (latent topics) of spatial items such as "exhibition" and "concert", and most of the estimated topics describe the user's spatial area of activity instead of user interests. That is because 1) the user's choice of spatial items is largely influenced by her/his geographical coordinates, and spatial items in the user's immediate neighborhood are likely to be chosen; 2) traditional latent factor models (e.g., topic models and matrix factorization methods) aim to capture item cooccurrence patterns. Another finding is that exploiting the content information of spatial items facilitates the clustering of items which are not only category-alike but also geo-diversity, alleviating the *new city* problem. The experimental results also explains the superiority of LCA-LDA and CA-LDA in the recommendation performance (Section 4.2.1).

5. RELATED WORK

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

5.1. Topic Model

Research in statistical models for cooccurrence data has given rise to a variety of useful topic models in the domain of text mining. The most representative models include PLSA [Hofmann 1999a, 1999b] and LDA [Blei et al. 2003; Griffiths and Steyvers 2004]. Yin et al. [2013a] proposed a probabilistic mixture model to detect both temporal topics and stable topics in a unified way. In these studies, they do not consider the location information of documents, so they do not focus on geographical topics. Recently, a brunch of work aims to study the geographic distributions of some topics in social media. For example, Yin et al. [2011] studied the distributions of some geographical topics (e.g., beach, hiking and sunset) in USA using geo-tagged photos acquired from Flickr. [Mei et al. 2006] proposed a probabilistic approach to model the subtopic themes and spatiotemporal theme patterns simultaneously in weblogs. Pozdnoukhov and Kaiser [2011] explored the space-time structure of topical content from a large number of geo-tweets. The social media data generated in a geo-region is still used as static features to depict a region. On the other hand, a few literatures have reported that human mobility can describe the functions of regions. For example, Qi et al. [2011] observed that the getting on/off amount of taxi passengers in a region can describe the social activity dynamics in the region. Yuan et al. [2012] proposed a topic model which considers both static features (POIs) of a region and human mobility between regions to infer the functions (e.g., residential, commercial and entertainment) of an area (a set of nearby locations).

In the recommendation domain, topic models have been applied to collaborative filtering. Jin et al. [2005] proposed an approach based on LDA to discover the hidden semantic relationships among items for recommendations. In Chen et al. [2009], a

[5]http://www.douban.com/event/.

hard-constraint-based LDA method was used to deal with user-community data, where each user is viewed as a document, and the communities that this user joins are viewed as words in the document. In contrast, Kang and Yu [2010] proposed a soft-constraint-based LDA method for community recommendations. We refer to these topic model-based CF methods as "traditional" recommendation techniques which produce recommendations using non-spatial user activity history, failing to exploit the location information of user activities. Kurashima et al. adopted the topic model for estimating user's interests based on geo-tag based histories on Flickr [2010]. However, the topic model proposed in Kurashima et al. [2010] fails to alleviate the *new city* problem because most of the discovered latent topics describes user's immediate area of activity instead of user interests (i.e., the estimated topics groups nearby locations). To facilitate the discovered topics to capture user interests, a Geo Topic Model was proposed in Kurashima et al. [2013] which jointly estimates both the user's interests and activity area hosting the user's home and office. Compared with our proposed LCA-LDA model, Geo Topic Model not only ignores the influence from the local preference, but also fails to exploit the contents of spatial items and analyze geographic distributions of the discovered topics.

5.2. General Recommender System

The Recommendation has been one of the most important services for many e-commerce and social networking services (e.g., netfix.com, amazon.com and facebook.com). The goal is to recommend an accurate list of items that match the target users' preferences. Collaborative filtering, Social filtering and Content-based filtering techniques are three widely adopted approaches for recommender systems [Adomavicius and Tuzhilin 2005]. All of them discover users' personal interests and utilize these discovered interests to find relevant items. Collaborative filtering techniques [Herlocker et al. 1999; Chen et al. 2009; Sarwar et al. 2001; Deshpande and Karypis 2004] automatically suggest relevant items for a given user by making use of the activity/rating history of a group of similar users or friends (i.e., user-based and social collaborative filtering) or a set of similar items (i.e., item-based collaborative filtering). The content-based recommendation [Ricci and Shapira 2011] is based on the assumption that descriptive features of an item tell much about a user's preference to the item. Thus a recommender system makes a decision for a user by matching a user's personal interest to the descriptive features of items. Recommender systems using pure collaborative filtering approaches tend to fail when little knowledge about the user is known or when no one has similar interest with the user. For example, if a user has little rating history or there is rare overlap between his/her and others' rating history, the item rating information of others cannot help. To overcome the data sparsity problem, a probabilistic matrix factorization framework, Social Trust Ensemble (STE), was proposed in Ma et al. [2009]. STE linearly fuses the users' tastes and their friends'favors together to produce recommendations. This method can alleviate the data sparsity, to some extent, by exploiting friends' preferences. It should be noted that the mixing weights in STE are manually set rather than learned automatically from the data. Although content-based method is capable of coping with the issue of lacking knowledge, it fails to account for community endorsement. For example, even though we know a user is interested in Italian restaurants, content-based methods may possibly recommend a bad Italian restaurant to him/her due to the lack of consideration of other users' opinions. As a result, there has been amount of research focusing on combining the advantages of both collaborative filtering/social filtering and content-based methods [Popescul et al. 2001; Basilico and Hofmann 2004; Kim et al. 2006; Ye et al. 2012]. Our proposal in this work is not only able to integrate the ideas behind collaborative filtering and content-based methods

but also incorporate the influence from the local preference into the recommendation process.

5.3. Nonpersonalized Location-Based Recommendations

Early location-based services employ KNN technique [Hjaltason and Samet 1999] and its variants [Papadias et al. 2005] to simply retrieve k objects nearest to a querying user and completely ignore the notion of user personalization. Recent generic spatial item recommendation systems encapsulate public opinions on spatial items and provide users with the most popular spatial items. For example, the spatial activity recommendation system [Zheng et al. 2010] mines GPS trajectory data embedded with user-provided tags in order to detect interesting activities located in a city. Besides considering only the number of user visits in GPS trajectory data, Zheng et al. [2009] also considered the travel experience of users. Cao et al. [2010] further extended this work to extract semantically meaningful and significant locations by considering the relations between locations and between locations and users. Instead of learning from the GPS data, recently, Venetis et al. [2011] study the problem of ranking nearby places by analyzing direction queries derived from large user populations. Therefore, generic location-based services exploit the location information of ratings in a fundamentally different manner. These methods cannot provide personalized recommendations because they do not consider individual preferences.

5.4. Personalized Location-Based Recommendations

Traditional location-based services require users to express explicit preference constraints, and then use skyline-based techniques [Sharifzadeh and Shahabi 2006] or sequential top-k query processing strategy [Marian et al. 2004] to retrieve interesting locations for users. There are several studies [Monreale et al. 2009; Alvarez-Lozano et al. 2012; Giannotti et al. 2007] that address the problem of predicting 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 which are concise representations of behaviors of moving objects as sequences of regions frequently visited with a typical travel time.

Recently, a branch of recent research focuses on automatically learning user preferences from the user's location history to make recommendations by using collaborative filtering models. Specifically, several researchers [Sarwat et al. 2013; Levandoski et al. 2012; Ye et al. 2010, 2011; Takeuchi and Sugimoto 2006] deposited user's activity history into user-venue matrix where a row corresponds to a user's venue history and each column denotes a venue like a restaurant. Each entry in the matrix represents the number of visits of a particular user paying to a physical venue. Then, a user-based CF model is used to infer users' preference to an unvisited venue. Geo-measured and friend-based collaborative filtering [Ye et al. 2010] produces recommendations by using only ratings that are from a querying user's social-network friends who live in the same city. Similar to the geo-measured and friend-based method, LARS [Sarwat et al. 2013; Levandoski et al. 2012] is a location-aware recommender system that uses location-based ratings to produce recommendations. LARS exploits user location information through user partitioning techniques which makes recommendations by considering only the ratings generated by users who are spatially close to the querying user. This method makes the sparse user rating matrix more sparse, degenerating the issue of data sparsity. LARS utilizes item location information through the travel penalty technique that penalizes the recommendation rank of items further in travel distance to querying users. It should be pointed out that all aforementioned methods which solely use a CF-based method, either the user-based or the item-based, cannot handle the data sparsity problem [Desrosiers and Karypis 2011] well, let alone the *new city* problem, as analyzed in Section 1.

There is a line of research focusing on reducing the data sparsity to some extent. Zheng et al. [2010] applied latent factor models such as matrix factorization to a user-venue matrix. Recently, factorization machine(FM) [Rendle 2012] was proposed to exploit the context information of user activities (e.g., location and time) to produce recommendations, but FM is designed for the task of rating prediction, not for the top-k recommendation. Gao et al. [2012] 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. Ye et al. [2011] exploited the *geographical clustering phenomenon* to improve the recommendation performance, with a unified framework to linearly fuse both user interest, social influence and geographical influence. Woerndl et al. [2011] 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. Noulas et al. [2012] proposed a random walk-based model to recommend new venues over a user-venue graph by combining social network and venue visit history data. Although these methods can alleviate the data sparsity problem, to some extent, by reducing dimensions and exploiting social /geographical influences, these methods do not work well in the *new city* setting because there are few, even no overlapped users between spatial items which are located at home cities and new cities respectively, that is, there is a gap between spatial items located in different cities, especially disjoint cities. Specifically, when we use existing latent factor models to analyze user activity history data, the learned latent factors cannot capture the semantic information of spatial items such as categories and genres, and most of them describe the location information of spatial items. The top spatial items (with high weights) in each latent factor appear to be within short distance, but share few semantics (e.g., categories and genres). Thus, the estimated latent factors represent geographical clusters, rather than semantic clusters due to the fact that most of people's activities are spatially clustered rather than clustered by semantic [Cheng et al. 2012; Kurashima et al. 2013; Cho et al. 2011]. Thus, it is most difficult for an estimated latent factor to cluster two spatial items located in different cities. Besides, as analyzed in multiple location-based social network datasets [Cho et al. 2011], more than 90% of users' friends live in the same cities with themselves, and most of their activity histories are in their living cities [Scellato et al. 2011b]. So, exploiting social influence cannot solve the *new city* problem well. Compared with our proposed LCARS, these methods neither exploit the content information of spatial items to build a bridge to link content-similar items together, especially spatial items located in different cities, nor take into account local preference and attractions.

Instead of using pure CF-based methods, Bao et al. [2012] proposed a hybrid method to alleviate *new city* problem and here we call it CKNN. Specifically, CKNN first projects a user's activity history into a well-designed category space and models each user's preferences with a weighted category hierarchy. Meanwhile, it infers the authority of each user in a city with respect to different category of spatial items according to their activity histories using HITS model [Kleinberg 1999]. When receiving a query $q = (u,\ l)$, CKNN first selects a set of high-quality users N_u in querying city l who have the same or similar preferences with the querying user u. Then, CKNN constructs a user-item matrix using the selected users N_u and their visited spatial items. Finally, CKNN employs a traditional user-based CF model over the user-item matrix to infer the querying user's rating of a candidate item. The advantage of CKNN over pure CF-based models is that it can find a set of local users who have similar preferences with the querying user by projecting users' activity histories into a well-designed category hierarchy. Compared with our proposed LCA-LDA model, CKNN ignores the observation of the shift of users' preferences, that is, people's preferences or behavioral patterns may change when they travel in different cities, especially in cities that are new to them. For example, a user

u living in a small city likes food very much, but does not like shopping. When she/he travels in HongKong, especially for the first time, the user is very likely to visit local shopping centers. CKNN would fail to recommend shopping centers in top positions in this case because most users in N_u prefer food rather than shopping centers, while our proposed LCA-LDA model can still work well in this case because it exploits the local preferences/attractions of the querying city to produce recommendations besides the querying user's interest, that is, what most of people have done when they travel in the querying city. That is why our proposed LCA-LDA model performs much better than CKNN.

The interest in location-based data spans beyond the domain of spatial item recommendation (e.g., points of interest and events). Many recent literatures [Scellato et al. 2011a; Wang et al. 2011; Cho et al. 2011; Quercia and Capra 2009] 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.

5.5. Advantages of Our Proposed LCARS

Our proposed location-content-aware recommender system distinguishes from the aforementioned works in the following four aspects: 1) We project a user's activity history into a latent space which integrates content knowledge of spatial items. This method handles the data sparsity problem and enables clustering of spatial items which do not share any user. So 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 (e.g., home city). 2) We take into account both user interest and local preference to produce recommendations. This mixture modeling method mimics the process of the user's decision making on spatial items. The local preference, which has been neglected before, is a valuable resource for making recommendations since people generally want to see local attractions and attend local popular events, especially when they travel to an unfamiliar city. 3) The ideas of integrating local preference's influence, collaborative filtering and content-based methods into a probabilistic generative model is unexplored. 4) Our proposed LCA-LDA model can generate an interpretable representation of each user profile which can be presented alongside item recommendation. Providing an interpretable user profile is very helpful for users to trust the recommender system and accept the recommendations because users can understand why and how the recommendations were made by viewing their estimated user profiles. 5) An efficient and scalable query processing technique enables LCARS to produce fast online top-k recommendations. Besides, LCARS can be deployed to support the scenarios of interactive recommendations and real-time mobile recommendations.

6. CONCLUSIONS

This article proposed a location-content-aware recommender system, LCARS, which provides a user with spatial item recommendations within the querying city based on the individual interest and the local preference mined from the user's activity history. LCARS can facilitate people's travel not only near their living areas but also to a city that is new to them even if they do not have any activity history there. 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 publicly available real datasets, DoubanEvent and Foursquare. According to the experimental results, our approach significantly outperforms existing recommendation methods in effectiveness. The results also justify each component proposed in our system, for instance, taking into account of local preference and item content information. Meanwhile, the proposed scalable

query processing technique based on TA and TA-ρ approximation algorithms, improves the efficiency of our approach significantly, enabling an online recommendation scenario and an interactive process. Besides, the experimental results also show that LCARS can generate an interpretable representation of each user profile which can be presented alongside spatial item recommendation.

REFERENCES

Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* 17, 6, 734–749.

Jorge Alvarez-Lozano, J. Antonio García-Macías, and Edgar Chávez. 2012. User location forecasting at points of interest. In *Proceedings of the RecSys Workshop on Personalizing the Local Mobile Experience (LocalPeMA'12)*. ACM, New York, 7–12.

Walid G. Aref and Hanan Samet. 1990. Efficient processing of window queries in the pyramid data structure. In *Proceedings of the 9th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. 265–272.

Jie Bao, Yu Zheng, and Mohamed F. Mokbel. 2012. 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 (SIGSPATIAL'12)*. ACM, New York, 199–208.

Justin Basilico and Thomas Hofmann. 2004. Unifying collaborative and content-based filtering. In *Proceedings of the 21st International Conference on Machine Learning (ICML'04)*. ACM, New York.

Sandro Bauer, Anastasios Noulas, Diarmuid O' . Seaghdha, Stephen Clark, and Cecilia Mascolo. 2012. Talking places: Modelling and analysing linguistic content in Foursquare. In *Proceedings of the ASE/IEEE International Conference on Social Computing and 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust (SOCIALCOM-PASSAT'12)*. IEEE, 348–357.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *J. Mach. Learn. Res.* 3, 993–1022.

Xin Cao, Gao Cong, and Christian S. Jensen. 2010. Mining significant semantic locations from GPS data. *Proc. VLDB Endow.* 3, 1–2, 1009–1020.

Wen-Yen Chen, Jon-Chyuan Chu, Junyi Luan, Hongjie Bai, Yi Wang, and Edward Y. Chang. 2009. Collaborative filtering for orkut communities: discovery of user latent behavior. In *Proceedings of the 18th International Conference on World Wide Web (WWW'09)*. ACM, New York, 681–690.

Chen Cheng, Haiqin Yang, Irwin King, and Michael R Lyu. 2012. Fused matrix factorization with geographical and social influence in location-based social networks. In *Proceedings of the Conference on Artificial Intelligence (AAAI'12)*.

Zhiyuan Cheng, James Caverlee, Kyumin Lee, and Daniel Z. Sui. 2011. Exploring millions of footprints in location sharing services. In *Proceedings of the International Conference on Weblogs and Social Media*. Lada A. Adamic, Ricardo A. Baeza-Yates, and Scott Counts, Eds., The AAAI Press.

Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and mobility: User movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11)*. ACM, New York, 1082–1090.

Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys'10)*. ACM, New York, 39–46.

Mukund Deshpande and George Karypis. 2004. Item-based top-N recommendation algorithms. *ACM Trans. Inf. Syst.* 22, 1, 143–177.

Christian Desrosiers and George Karypis. 2011. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender Systems Handbook*, 107–144.

Ronald Fagin, Amnon Lotem, and Moni Naor. 2001. Optimal aggregation algorithms for middleware. In *Proceedings of the 20th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS'01)*. ACM, New York, 102–113.

Huiji Gao, Jiliang Tang, and Huan Liu. 2012. gSCorr: Modeling geo-social correlations for new check-ins on location-based social networks. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM'12)*. ACM, New York, 1582–1586.

Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. 2007. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07)*. ACM, New York, 330–339.

Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proc. Nat. Acad. Sci. U.S.A.* 101, Suppl. 1, 5228–5235.

Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. 1999. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*. ACM, New York, 230–237.

Gísli R. Hjaltason and Hanan Samet. 1999. Distance browsing in spatial databases. *ACM Trans. Database Syst.* 24, 2, 265–318.

Thomas Hofmann. 1999a. Probabilistic latent semantic analysis. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence, UAI'99*. 289–296.

Thomas Hofmann. 1999b. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*. ACM, New York, 50–57.

Tzvetan Horozov, Nitya Narasimhan, and Venu Vasudevan. 2006. Using location for personalized POI recommendations in mobile environments. In *Proceedings of the International Symposium on Applications on Internet (SAINT'06)*. IEEE, 124–129.

Xin Jin, Yanzan Zhou, and Bamshad Mobasher. 2005. A maximum entropy web recommendation system: Combining collaborative and content features. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'05)*. ACM, New York, 612–617.

Yujie Kang and Nenghai Yu. 2010. Soft-constraint based online LDA for community recommendation. In *Proceedings of the Advances in Multimedia Information Processing, and 11th Pacific Rim Conference on Multimedia: Part II (PCM'10)*. Springer, 494–505.

Byeong Man Kim, Qing Li, Chang Seok Park, Si Gwan Kim, and Ju Yeon Kim. 2006. A new approach for combining content-based and collaborative filters. *J. Intell. Inf. Syst.* 27, 1, 79–91.

Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *J. ACM* 46, 5, 604–632.

Ioannis Konstas, Vassilios Stathopoulos, and Joemon M. Jose. 2009. On social networks and collaborative recommendation. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'09)*. ACM, 195–202.

Yehuda Koren. 2008. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)*. ACM, New York, 426–434.

Takeshi Kurashima, Tomoharu Iwata, Takahide Hoshide, Noriko Takaya, and Ko Fujimura. 2013. Geo topic model: Joint modeling of user's activity area and interests for location recommendation. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining (WSDM'13)*. ACM, New York, 375–384.

Takeshi Kurashima, Tomoharu Iwata, Go Irie, and Ko Fujimura. 2010. Travel route recommendation using geotags in photo sharing sites. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM'10)*. ACM, New York, 579–588.

Justin J. Levandoski, Mohamed Sarwat, Ahmed Eldawy, and Mohamed F. Mokbel. 2012. LARS: A location-aware recommender system. In *Proceedings of the IEEE 28th International Conference on Data Engineering (ICDE'12)*. IEEE, 450–461.

Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Comput.* 7, 1, 76–80.

Xingjie Liu, Qi He, Yuanyuan Tian,Wang-Chien Lee, John McPherson, and Jiawei Han. 2012. Event-based social networks: linking the online and offline social worlds. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12)*. ACM, New York, 1032–1040.

Hao Ma, Irwin King, and Michael R. Lyu. 2009. Learning to Recommend with social trust ensemble. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'09)*. ACM, 203–210.

Amélie Marian, Nicolas Bruno, and Luis Gravano. 2004. Evaluating top-k queries over web-accessible databases. *ACM Trans. Database Syst.* 29, 2, 319–362.

Qiaozhu Mei, Chao Liu, Hang Su, and ChengXiang Zhai. 2006. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proceedings of the 15th international conference on World Wide Web (WWW'06)*. ACM, New York, 533–542.

Anna Monreale, Fabio Pinelli, Roberto Trasarti, and Fosca Giannotti. 2009. WhereNext: A location predictor on trajectory pattern mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*. ACM, New York, 637–646.

Anastasios Noulas, Salvatore Scellato, Neal Lathia, and Cecilia Mascolo. 2012. A randomwalk around the city: New venue recommendation in location-based social networks. In *Proceedings of the ASE/IEEE International Conference on Social Computing and 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust (SOCIALCOM-PASSAT'12)*. IEEE, 144–153.

Dimitris Papadias, Yufei Tao, Kyriakos Mouratidis, and Chun Kit Hui. 2005. Aggregate nearest neighbor queries in spatial databases. *ACM Trans. Database Syst.* 30, 2, 529–576.

Alexandrin Popescul, Lyle H. Ungar, David M. Pennock, and Steve Lawrence. 2001. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI'01)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 437–444.

Alexei Pozdnoukhov and Christian Kaiser. 2011. Space-time dynamics of topics in streaming text. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks (LBSN'11)*. ACM, New York, 1–8.

Guande Qi, Xiaolong Li, Shijian Li, Gang Pan, Zonghui Wang, and Daqing Zhang. 2011. Measuring social functions of city regions from large-scale taxi behaviors. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops*. 384–388.

Daniele Quercia and Licia Capra. 2009. FriendSensing: recommending friends using mobile phones. In *Proceedings of the 3rd ACM Conference on Recommender Systems (RecSys'09)*. ACM, New York, NY, USA, 273–276.

Steffen Rendle. 2012. Factorization Machines with libFM. *ACM Trans. Intell. Syst. Technol.* 3, 3, Article 57.

Francesco Ricci and Bracha Shapira. 2011. *Recommender Systems Handbook*. Springer.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW'01)*. ACM, New York, 285–295.

Mohamed Sarwat, Justin J. Levandoski, Ahmed Eldawy, and Mohamed F. Mokbel. 2013. LARS*: An efficient and scalable location-aware recommender system. *IEEE Trans. Knowl. Data Eng.* 99, 1.

Salvatore Scellato, Anastasios Noulas, Renaud Lambiotte, and Cecilia Mascolo. 2011b. Socio-spatial properties of online location-based social networks. In *Proceedings of the International Conference on Weblogs and Social Media*. Lada A. Adamic, Ricardo A. Baeza-Yates, and Scott Counts, Eds., AAAI Press.

Salvatore Scellato, Anastasios Noulas, and Cecilia Mascolo. 2011a. Exploiting place features in link prediction on location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11)*. ACM, New York, 1046–1054.

Mehdi Sharifzadeh and Cyrus Shahabi. 2006. The spatial skyline queries. In *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB'06)*. VLDB Endowment, 751–762.

Yuichiro Takeuchi and Masanori Sugimoto. 2006. CityVoyager: An outdoor recommendation system based on user location history. In *Proceedings of the 3rd International Conference on Ubiquitous Intelligence and Computing (UIC'06)*. Springer, 625–636.

Jie Tang, SenWu, Jimeng Sun, and Hang Su. 2012. Cross-domain collaboration recommendation. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12)*. ACM, New York, 1285–1293.

Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. ArnetMiner: Extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)*. ACM, New York, 990–998.

Petros Venetis, Hector Gonzalez, Christian S. Jensen, and Alon Halevy. 2011. Hyper-local, directions-based ranking of places. *Proc. VLDB Endow.* 4, 5, 290–301.

DashunWang, Dino Pedreschi, Chaoming Song, Fosca Giannotti, and Albert-Laszlo Barabasi. 2011. Human mobility, social ties, and link prediction. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11)*. ACM, New York, 1100–1108.

Stanley Wasserman and Katherine Faust. 1994. *Social Network Analysis: Methods and Applications*. Vol. 8, Cambridge University Press.

Wolfgang Woerndl, Johannes Huebner, Roland Bader, and Daniel Gallego-Vico. 2011. A model for proactivity in mobile, context-aware recommender systems. In *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys'11)*. ACM, New York, 273–276.

Mao Ye, Xingjie Liu, and Wang-Chien Lee. 2012. Exploring social influence for recommendation: a generative model approach. In *Proceedings of the 35th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'12)*. ACM, New York, 671–680.

Mao Ye, Peifeng Yin, and Wang-Chien Lee. 2010. Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS'10)*. ACM, New York, NY, USA, 458–461.

Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'11)*. ACM, New York, 325–334.

Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen. 2012. Challenging the long tail recommendation. *Proc. VLDB Endow.* 5, 9, 896–907.

Hongzhi Yin, Bin Cui, Hua Lu, Yuxin Huang, and Junjie Yao. 2013a. A unified model for stable and temporal topic detection from social media data. In *Proceedings of the 29th IEEE International Conference on Data Engineering (ICDE'13)*. IEEE.

Hongzhi Yin, Yizhou Sun, Bin Cui, Zhiting Hu, and Ling Chen. 2013b. LCARS: A location-content-aware recommender system. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'13)*. ACM, New York.

Zhijun Yin, Liangliang Cao, Jiawei Han, Chengxiang Zhai, and Thomas Huang. 2011. Geographical topic discovery and comparison. In *Proceedings of the 20th International Conference on World Wide Web (WWW'11)*. ACM, New York, 247–256.

Jing Yuan, Yu Zheng, and Xing Xie. 2012. Discovering regions of different functions in a city using human mobility and POIs. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12)*. ACM, New York, NY, USA, 186–194.

Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. 2013. Time-aware point-of-interest recommendation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'13)*. ACM, 363–372.

Vincent W. Zheng, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative location and activity recommendations with GPS history data. In *Proceedings of the 19th International Conference on World Wide Web (WWW'10)*. ACM, New York, 1029–1038.

Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. 2009. Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of the 18th International Conference on World Wide Web (WWW'09)*. ACM, New York, 791–800.

Received June 2013; revised December 2013; accepted March 2014