Location-based and Preference-Aware Recommendation Using Sparse Geo-Social Networking Data *

Jie Bao[1] Yu Zheng[2] Mohamed F. Mokbel[1]

[1]Department of Computer Science & Engineering, University of Minnesota, Minneapolis, USA
[2]Microsoft Research Asia, No. 5 Danling Street, Haidian District, Beijing, China
{baojie,mokbel}@cs.umn.edu, yuzheng@microsoft.com

ABSTRACT

The popularity of location-based social networks provide us with a new platform to understand users' behavior and preferences based on their location histories. In this paper, we present a location-based and preference-aware recommender system that offers a particular user a set of venues (such as restaurants and shopping malls) within a geospatial range with the consideration of both: 1) User personal preferences, which are automatically learned from her location history and 2) Social opinions, which are mined from the location histories of the *local experts*. This recommender system can facilitate people's travel not only near their living areas but also to a city that is new to them. As a user can only visit a limited number of locations, the user-locations matrix is very sparse, leading to a big challenge to traditional collaborative filtering-based location recommender systems. The problem becomes even more challenging when people travel to a new city where they could have not visited. To this end, we propose a novel location recommender system, which consists of two main parts: *offline modeling* and *online recommendation*. The *offline modeling* part models each individual's personal preferences with a weighted category hierarchy (WCH) and infers the expertise of each user in a city with respect to different category of locations according to their location histories using an iterative learning model. The *online recommendation* part selects candidate *local experts* in a user specified geospatial range that matches the user's preferences using a preference-aware candidate selection algorithm and then infers a score of the candidate locations based on the opinions of the selected *local experts*. Finally, the top-k ranked locations are returned as the recommendations for the user. We evaluated our system with a large-scale real dataset collected from Foursquare. The results confirm that our method offers more effective recommendations than baselines, while having a good efficiency of providing location recommendations.

Keywords

location-based social networks, location-based services, user preferences, recommendation systems.

*The work was done when the first author was performing an internship in Microsoft Research Asia.

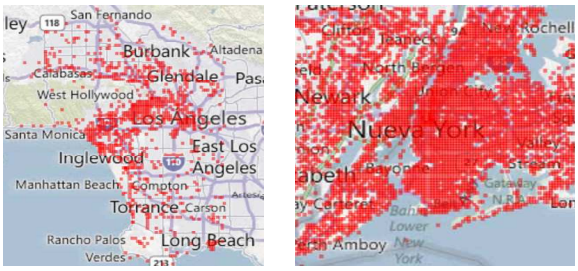
ACM SIGSPATIAL GIS '12, November 6-9, 2012. Redondo Beach, CA, USA

1. INTRODUCTION

The advances in location-acquisition and wireless communication technologies enable people to add a location dimension to traditional social networks, fostering a bunch of location-based social networking services (or LBSNs) [25], e.g., Foursquare, Loopt, and GeoLife [27], where users can easily share life experiences in the physical world via mobile devices. For example, a user can leave comments with respect to a restaurant in a LBSN site, so that the people from her social structure can refer to the comments when they visit the restaurant in a later time. Location as one of the most important components of user context implies extensive knowledge about an individual's interests and behavior, thereby providing us with opportunities to better understand users in a social structure according to not only online user behavior but also the user mobility and activities in the physical world. For instance, people often visiting gyms might like physical exercises and users who usually have dinner in the same restaurant may share a similar taste. Sometimes, individuals who do not have overlaps of physical locations can still be linked, as long as the categories of their visited locations are indicative of a similar interest, such as beaches or museums.

Under such a circumstance, a location recommender system is a valuable but unique application in location-based social networking services, in terms of what a recommendation is and where a recommendation is to be made [16, 25]. Specifically, location recommendations provide a user with some venues (e.g., an Italian restaurant or a fancy movie theater) that match her personal interests within a geospatial [25]. This application becomes more worthy when people travel to an unfamiliar area, where they have little knowledge about the neighborhoods. Nevertheless, a high-quality location recommendation has to simultaneously consider the following three factors. 1) *User preferences*: For example, food hunters maybe more interested in the high quality restaurants, while the shoppingaholics would pay more attentions to nearby shopping malls [17]. 2) *The current location of a user*: As the users prefer the nearby locations, this location indicates the spatial range of the recommended venues and may affect the ratings of these recommendations [14]. 3) *The opinions of a location given by the other users*: Social opinions from the nearby users is a valuable resource for making a recommendation [9]. But, the most popular venue may not always fit a particular user given her distinct preferences.

Inferring the rating for a location is very challenging using a user's location history in a LBSN. First, a user can only visit a limited number of physical locations. This results in a sparse user-location matrix for most existing location recommendation systems, e.g., [14, 9], which directly play a collaborative filtering-based model [8, 12] over physical locations. Second, the task becomes even more difficult when an individual travels to a new place



(a) New York users in Los Angels (b) New York users in New York City.

Figure 1: User Location History Distributions.

where she has visited few locations (though we believe people need the location recommendation service most at this moment). For example, Figure 1 a) and b) plot the locations (according to the *tips* in Foursquare) visited by people from New York City, in Los Angles (LA) and New York City (NYC) respectively. Clearly, the *tip* records generated by NYC people are very few in LA, which are only 0.47% of the records they left in NYC and 0.75% of the records generated by local users in LA. This phenomenon is quite common in the real world [20], aggravating the data sparse problem to location rating inference (if we want to provide people from NYC with location recommendations in LA). In this case, solely using a CF model is not feasible any more. First, we cannot simply put together the location histories of users from different cities in to a user-location matrix, which is neither efficient nor scalable. Second, performing collaborative inference in each city separately cannot cope with the *new city* problem demonstrated in Figure 1 a) very well, as a user usually has not enough location history in a city that is new to her.

To this end, we report on a location-based and preference-aware recommender system that offers a particular user a set of venues (such as restaurants and shopping malls) within a user specified geospatial range with the consideration of the three factors mentioned in the third paragraph. By modeling a user's preferences based on the category information of her location history (instead of physical locations) in a LBSN, our recommender system can facilitate people's travel not only near their living areas but also to a city that is new to them. Generating such a location recommendation is challenging because of two reasons:

1) *Learning a user's preferences.* First of all, a user's preferences are usually comprised of multiple kinds of interests, such as shopping, watching movies, cycling, and arts. By the meantime, a user's preferences are not generally binary decisions, e.g., like or dislike something, and have a variety of granularities, such as "*Food* $\rightarrow$ *Italian food* $\rightarrow$ *Italian noodles*". In addition, a user's preferences are evolving from time to time. Manually specifying an individual's preferences with some words is impractical. As a result, unobtrusively modeling a user's preferences with her location history is non-trivial.

2) *Inferring the rating to an unvisited location for an individual.* The rating inference needs to consider both an individual's preferences, the opinions given by the other users, especially the *local experts* [2, 13], and the similarity between them. This inference demands three aspects of computing: a) estimating the expertise of a user, b) computing the similarity between users, and c) collaborative social opinion inference for a location incorporating the results of the former two computation, e.g., using collaborative filtering (CF) model [8, 12]. None of them are trivial.

Specifically, our contributions can be summarized as:

- We learn a user's preferences from her location history and model the preferences with a weighted category hierarchy (WCH). We further estimate the similarity between two users' preferences by computing the similarity between the two users' WCHs. This method contributes to user preference modeling and handling the data sparseness problem for location recommendations.
- We pre-compute and extract the *local expert* for each location category in a city using an iterative inference model over the users' location histories there, which improves the efficiency of our online recommendation process.
- We online infer the rating to a venue with the *local experts* selected by a preference-aware candidate selection algorithm and a CF-based model. This approach enables a real-time location recommendation simultaneously considering an individual's location, preferences granularities, and opinions from *local experts*.
- We evaluated our system with a real-world dataset collected from Foursquare including 221,128 tips generated by 49,062 users in NYC and 104,478 tips generated by 31,544 users in LA. The extensive experimental results show that our method provide users with location recommendations more effectively and efficiently beyond the existing baselines.

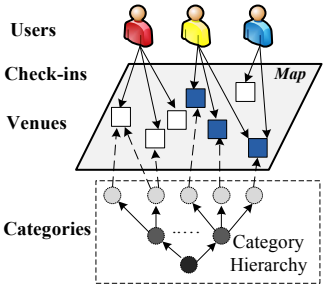
The rest of the paper is organized as follows: Section 2 gives an overview of our system. Section 3 and Section 4 present two major parts of our system: 1) *offline modeling* and 2) *online recommendation*. Extensive experimental results based on the real dataset are provided in Section 5 with some discussions. Section 6 summarizes the related works. Finally, Section 7 concludes the paper.

2. SYSTEM OVERVIEW

This section first introduces the key data structures we will use in the paper, and then presents the application scenario and overall architecture of the proposed location recommender system.

2.1 Preliminary

Figure 2 illustrates the relations of five key data structures: 1) *user*, 2) *venue*, 3) *check-in*, 4) *user location history* and 5) *category hierarchy*. In a location-based social network, a user u maintains her profile information, such as ID, name, age, gender, and home town. Moreover, the user can also mark a venue (e.g., a restaurant) and leave some comments, when she arrives there, which is also known as *check-in* in a LBSN. A user can visit multiple locations and may generate a *check-in* for each of the visit, shown as the solid arrows in Figure 2 a). All of the user's *check-ins* reflect her *location history* in the real world. Depicted as squares on the map, a venue is a location associated with a pair of coordinates indicating

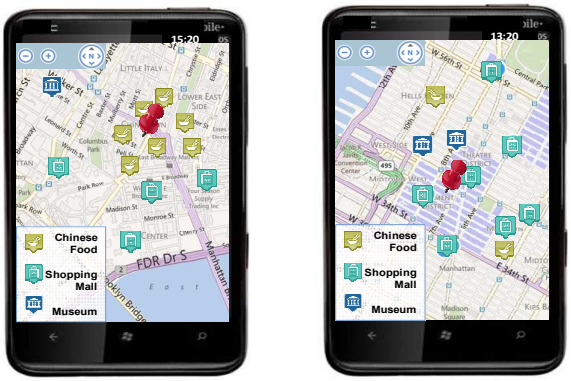


(a) Overview of a location-based social network

Category Name	Number of sub-categories
Arts & Entertainment	17
College & University	23
Food	78
Great Outdoors	28
Home, Work, Other	15
Nightlife Spot	20
Shop	45
Travel Spot	14

(b) Detailed location category hierarchy in FourSquare

Figure 2: Data Structures in Location-Based Social Networks.



(a) Recommendations near Chinatown (b) Recommendations near 7th Ave

Figure 3: Example of An Application Scenario in NYC.

its geographical position and a set of categories denoting its functionalities. The categories of venues have different granularities, which are usually represented by a *category hierarchy* shown in the bottom part of Figure 2 a). For example, "Food" category includes "Chinese restaurant" and "Italian restaurant" and etc. In our system, we focus on a two-level category hierarchy obtained from Foursquare, as shown in Figure 2 b).

2.2 Application Scenario

Figure 3 demonstrates an application scenario of our system, where the top N (N=10 here) venues matching a user's preferences are recommended based on the geo-region of the present view. Here, the number of recommendations and scale of the geo-region are determined by a user (e.g., by zooming in/out and panning a map in Figure 3, while the ranking of the locations are calculated in our backend system, based on the location history of the user and the opinions from the other people. Generally, the number of locations belonging to a category in the recommendations follows the distribution of the categories in the user's preferences. For example, the user (whose location is represented by the push-pin in Figure 3) has "Chinese restaurants" as her most preferred location category and "Shopping malls" as the second. Then, as demonstrated in Figure 3 a), "Chinese restaurants" have the biggest presence and shopping malls are the second in the recommendations, when she is near the Chinatown. However, when we change the map view to the 7th Ave, as shown in Figure 3 b), the presence of malls could become the majority of the recommendations though Chinese restaurants is her first interest. The reason is that the malls have much higher quality than the Chinese restaurants, according to people's location histories in that particular area. This is a trade-off between the user preferences and social opinions.

2.3 System Architecture

Offline Modeling. The offline modeling part is comprised of two major components: 1) *social knowledge learning* and 2) *personal preference discovery*, as illustrated in the lower half of Figure 4. The first component infers each user's expertise in each category city-by-city according to their location histories. Given a pre-defined category hierarchy (e.g., Figure 2 b), we break a user's location history in a city into groups of different location categories. Then, we model each category group of location histories using a user-location matrix, in which each entry denotes a user's number of visits to a physical location. By applying an iterative inference model to each user-location matrices, we calculate a score w.r.t. a category for each user, indicating a user's expertise in that category in that

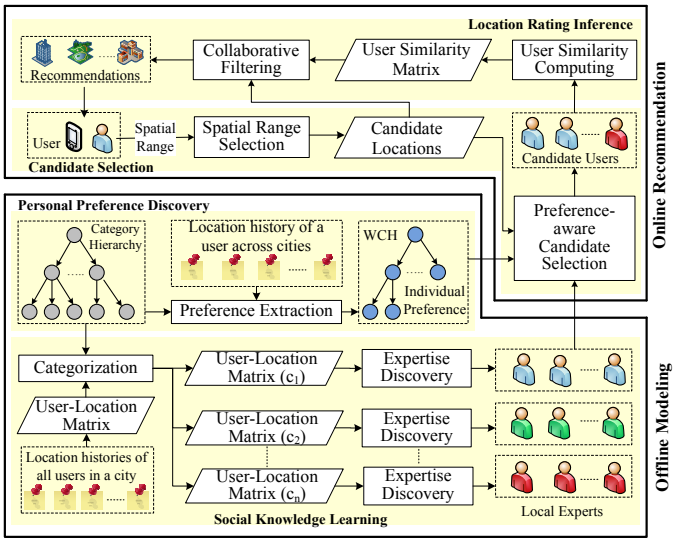


Figure 4: System Architecture.

city. By ranking the users in terms of the score corresponding to a category, we can discover the *local experts* of different categories in the city. The inferred expertise of a user will be used in later preference-aware candidate selection algorithm and help the online part generate quality recommendations with fewer computational loads. The second component models each user's personal preferences using a WCH by taking advantage of the location category information lying her location history, which help us to overcome the data sparsity problem. Specifically, a WCH is a sub-tree of the predefined category hierarchy, where each node carries a value denoting the user's number of visits to a category. These values are further normalized on each layer of a WCH using TF-IDF (term frequency- inverse document frequency) [19].

Online recommendation. The online recommendation part provides a user with a list of venues, considering the user's preferences, current location, and social opinions from the selected *local experts*, detailed in the following two components: 1) *Preference-aware candidate selection.* This component selects a set of *local experts* who visited the venues within a user's recommendation range R and have a high expertise in the categories preferred by the user. A preference-aware candidate selection algorithm is designed to properly choose these *local experts* from different categories according to a user's different preference weights in her WCH. Meanwhile, this algorithm improves the efficiency of our approach significantly while maintaining the effectiveness, making our system really location-aware. 2) *Location rating calculation.* This component first computes the similarity between each selected *local expert* and the user using a similarity function based on their WCHs. The calculated similarity score is further fed into a CF-based model to infer the rating that the user would give to an unvisited candidate venue. Later, the venues with relative high predict ratings are returned as the location recommendations.

3. OFFLINE MODELING

In this section, we present the *offline modeling* part of our system, which is comprised of: 1) *Social knowledge learning*, which evaluates a user's experiences and discovers the *local experts* in each city, and 2) *Personal preference discovery*, which extracts a user's preferences from her location history.

3.1 Social Knowledge Learning

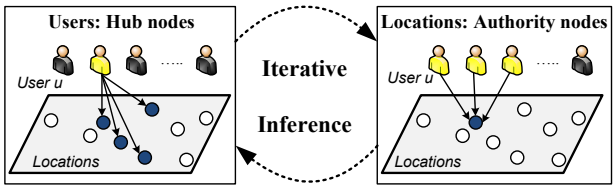


Figure 5: The Iterative Model for Social Knowledge Learning.

To identify the *local experts* of a location category like "Chinese food" and "shopping mall", this component computes a user's expertise in each category in different cities based on category information encapsulated in the user's location history. Intuitively, *local experts* of a category can find high quality venues of the category as compared with the regular users, resulting in more valuable location histories for a reference. In addition, using the *local experts* we are able to ignore some random users who have little data (and knowledge) in a category of locations, thereby reducing unnecessary computation during the online recommendation.

In our method, we first partition all users' location histories by cities as a user's knowledge usually varies in terms of geographic spaces, e.g., a travel expert of New York City may have no idea about the interesting venues in Beijing. Moreover, users may have different expertise in different location categories, e.g., a user likes "Chinese food" in the city does not necessary have much knowledge about "Italian food" there. Thus, we further divide users' location histories in a city into groups according to the categories of their visited venues. As a result, a city has n user-location matrices (n is the number of predefined categories) where an entry denotes the number of visits of a user to a venue. Later, we apply a HITS (or Hypertext Induced Topic Search)-based inference model [4, 10] to each category-based user-location matrix, inferring the expertise of each user in that category. As shown in Figure 5, this model regards an individual's visit to a venue as a directed link from the user to that venue. Each user has a hub score denoting its knowledge and each location is associated with an authority score indicating its interest level. The insight supporting this model is the mutual reinforcement relationship between a user's knowledge and the interest level of a venue [29]. That is, people who have visited many high quality venues in a region are more likely to have rich knowledge about that region. In turn, a venue visited by many people with rich knowledge is more likely to be a quality venue. As a result, as shown in Equation 1 and 2, a user's knowledge can be represented by the sum of the authority scores (i.e., interest levels) of the venues visited by the user, and the interest level of a venue can be represented by the sum of the hub scores (or knowledge) of the users who have visited this venue. Using a powerful iteration inference method, we generate the final scores for each user and each venue. The users with a relatively high authority score are regarded as the *local experts* in that category.

$$v_c.a = \sum_{u \in \mathcal{U}} u_c.h \tag{1}$$

$$u_c.h = \sum_{u.v \in c} v_c.a \tag{2}$$

where $u_c.h$ is user u's hub score in category c and $v_c.a$ denotes venue v's authority score.

If we use $\mathcal{A}_n$ and $\mathcal{H}_n$ to denote authority and hub scores at the nth iteration and $\boldsymbol{M}$ as the user-category matrix, the iterative processes for generating the final results are:

$$\mathcal{A}_n = \boldsymbol{M}^T \cdot \boldsymbol{M} \cdot \mathcal{A}_{n-1} \tag{3}$$

$$\mathcal{H}_n = \boldsymbol{M} \cdot \boldsymbol{M}^T \cdot \mathcal{H}_{n-1}, \tag{4}$$

as we set the initial authority and hub scores as the number of a user's visits, we are able to calculate the authority and hub scores using the power iteration method and identify the *local experts*.

3.2 Personal Preference Discovery

We extract a user's preferences from the category of her visited locations. As illustrated in Figure 6, we first project a user's location history across all the cities onto a predefined category hierarchy, where nodes occurring on a deeper layer denote the categories of a finer granularity. As a result, each node is associated with a value representing the number of visits (of the user) to a category. This is motivated by the fact that an individual's preferences are usually made up of multiple interests (such as shopping and hiking), which further have different granularities, e.g., "Food" $\rightarrow$ "Chinese food". Second, we calculate the TF-IDF value of each node in the hierarchy, where a user's location history is regarded as a document and categories are considered as terms in the document. Intuitively, a user would visit more locations belonging to a category if the user likes it. Further, if a user visits locations of a category that is rarely visited by other people, the user could like this category more prominently. For example, the number of visits to restaurants is generally more than other categories like museums in people location histories. It does not mean food is the first interest of all the people. However, if we find a user visits museums very frequently, the user may be truly interested in arts or history.

Overall, a user's preference weight ($u.w_{c'}$) is calculated by Equation 5, where the first part of the equation is the TF value of category c in user u's location history and the second part denotes the IDF value of the category.

$$u.w_{c'} = \frac{|\{u.v_i : v_i.c = c'\}|}{|u.\mathcal{V}|} \times \lg \frac{|\mathcal{U}|}{|\{u_j{:}c' \in u_j.\mathcal{C}\}|}, \tag{5}$$

where $|\{u.v_i : v_i.c = c'\}|$ is user u's number of visits in category c', $u.\mathcal{V}$ is the total number of the user's visits, and $|\{u_j{:}c' \in u_j.\mathcal{C}\}|$ counts the number of users who have visited category c' among all the users $\mathcal{U}$ in the system. Clearly, after applying IDF to the user's WCH, Chinese restaurant is no longer the first preference (i.e., with lighter color). The WCH well captures a user's interests, having the following advantages: 1) reduce the concern raised by the different data scales of different users, 2) handle the data sparseness problem and reduce the computational loads for further user similarity computing (from physical locations to categories), and 3) enable the computing of similarity between users who do not share any physical location histories, e.g., living in different cities.

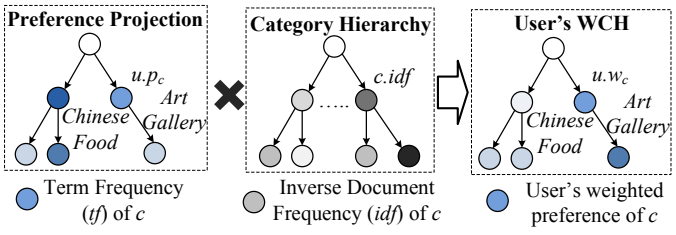


Figure 6: User WCH Construction.

Algorithm 1: Preference Aware Candidate Selection

Input: (1) Spatial Region R, (2) A user's *u.wch*, and (3) Total number of location recommendations N.
Output: (1) A set of selected local experts E and (2) A set of candidate locations V

1. Retrieve venues V' in R
2. $U \leftarrow$ users who have visited V'
3. **while** *True* **do**
4. **for** level l from bottom to the root-1 in *u.wch* **do**
5. $w_{min} \leftarrow$ minimum preference weight at l
6. **for** ***each*** category c in user's *u.wch* at level l **do**
7. $k \leftarrow |u.w_c/w_{min}|$ *//Calculate the number of users*
8. $e \leftarrow$Top(k, U, c) *// Select top-k users based on u'_c.h*
9. **for** ***each*** $u' \in e$ **do**
10. $V \leftarrow V \cup u'.V$ located in R
11. $E \leftarrow E \cup e$
12. **if** enough candidate venues $|V| \geq N$ ***or*** $E == U$ **then**
13. **Return** local experts E and candidate locations V

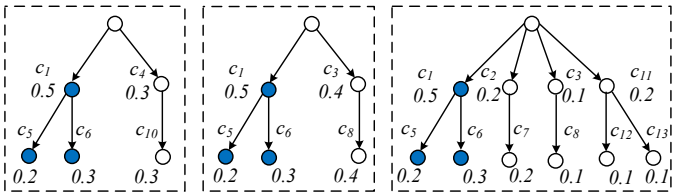
4. ONLINE RECOMMENDATION

In this section, we present *online recommendation* part of our system, which consists of: 1) *preference-aware candidate selection*, which selects the candidate*local expert* based on the user's preferences and 2) *location rating calculation*, which infers a predication score of the candidate locations the user would give based on CF-based inference model using the similarity comparison between the user and selected *local experts*.

4.1 Preference-Aware Candidate Selection

This component selects a set of candidate *local experts* and venues in the user specified geospatial range using our preference-aware candidate selection algorithm (i.e., demonstrated as Algorithm 1), which guarantees the number of selected venues exceeds the individual's requirement k and the category distribution of the selected *local experts* fits the individual's preferences. The algorithm significantly improves the efficiency of the online recommendations process as we do not need to compute the similarity between the individual and all the users in the area any more. Meanwhile, the location history of users with very little knowledge about the region can be excluded, as they may have limited contributions to the final score inference. The experiments show that the candidate selection increases the efficiency significantly while maintaining the effectiveness.

Specifically, given a geospatial range R specified by the individual, this algorithm first retrieves the venues V' located in the range and users U who have visited these venues (Line 1 and 2).The candidate *local experts* selection process initiates from the bottom level of the individual's WCH (which has a finer granularity) and moves up to the next higher level if the number of venues cannot meet the required number of recommendations. When selecting venues at one level of WCH, we choose the node (a category) having the minimum value w_{min}. Later, we calculate a k value using $|\frac{u.wc}{w_{min}}|$ to decide the number of *local experts* we select in this category, and then top-k users with a relatively high expertise (hub score) in category c are selected as candidate experts e (Line 7-8). The venues (located in R) visited by the users in e will be retrieved and deposited into V. After that, candidate experts e are merged with E (Line 9-11). The algorithm will stop once we obtain enough number of venues or all the users who have visited region R have been scanned. As a result, a set of venues V and a set of *local experts* E are returned.



(a) WCH of u_1 (b) WCH of u_2 (c) WCH of u_3

Figure 7: Diversities of Users' Preferences.

4.2 Location Rating Inference

Step 1. User Similarity Computing. In this step, we compute a similarity score between an individual (who issues the recommendation request) and each *local expert* (selected by Algorithm 1) according to their WCHs. Since a WCH is essentially a tree, we measure the similarity between the two WCHs in terms of both their structures and the preference weights associated with each overlapped node. Specifically, we decompose the similarity between two WCHs as a weighted sum of the similarities between each corresponding level of the WCHs (i.e., $u.wch.l_1$ vs. $u'.wch.l_1$). The deeper levels are given a bigger weight as they represent a finer granularity of an individual's preferences. Further, the similarity between the same levels of two different WCHs is measured by the following two aspects:

The first one is the number of overlapped nodes at the level and their values, as shown in Equation 6. The more overlapped nodes two WCHs have the more similar the two users could be. The minimum preference weight of an overlapped node c is selected to represent two users' common interests.

$$\textit{LevelSim}(u, u', l) = \sum_{c \in \mathcal{C}^l} \textit{min}(u.w_c, u'.w_c), \tag{6}$$

The other is the entropy of each level, which can effectively capture the diversity of a user's preferences [7], as shown in Equation 7, where $H(u, l)$ is user u's entropy at level l and $P(c)$ is the probability that u visited category c in her historical data.

$$H(u, l) = -\sum_{c \in \mathcal{C}^l} u.P(c) \times \lg u.P(c), \tag{7}$$

Figure 7 illustrates the importance of this entropy using an example, where three users share some same preferences (marked blue in WCHs) and the values represent the weights. Without considering the entropy of each level, the similarity scores *Sim*(u_1, u_2) and *Sim*(u_1, u_3) are identical. However, we can clearly observe that u_1 is more similar to u_2 who is relatively focused than u_3 who has a variety of interests. Or, we can say u_3 is more different from u_1 as compared with u_2 since u_3 has more different categories. We validated the effectiveness of the entropy in later experiments.

Finally, the similarity between two WCHs can be calculated as Equation 8, where β is a weight varying in the depth of the level of the location category (the depth of a root is 0) in the hierarchy. In the experiment we choose β=2^l as we found the overlapped nodes decreased exponentially as the depth of levels increases.

$$\textit{Sim}(u, u') = \sum_{l=1}^{|l|} \beta \times \frac{\textit{LevelSim}(u, u', l)}{1 + |H(u, l) - H(u', l)|} \tag{8}$$

That is, two users are more likely to be similar if 1) they share more nodes with a bigger preference weight, 2) the difference be-

tween each level's entropy is small, and 3) these nodes located in a lower level in their own WCHs.

Step 2. Location Rating Calculation. In this step, we place the *local experts* and candidate venues selected by Algorithm 1 back into a user-location matrix, which is fed into a user-based CF model to infer a user's rating of a candidate venue. The general intuition behind a CF model is that similar users rate the same items similarly. As users usually do not offer explicit ratings to a venue in a LBSN, we regard a user's number of visits to the venue as an implicit rating (of the venue). Formally, the rating that user u would give to venue v is calculated as Equation 9.

$$R_u(v) = \sum_{u' \in \mathcal{E} \& v \in \mathcal{V}} Sim(u, u') \times v(u', v), \quad (9)$$

where $v(u', v)$ denotes the number of visits of user u' at venue v. Note that the user similarity $Sim(u, u')$ is computed in the Step 1 based on WCHs rather than the simple Cosine similarity between two users's location vectors. That is, we can still make recommendations for a user even if the user has not visited any locations in a new city. Finally, the system returns the top-N venues with the highest scores to the user as the location recommendations.

5. EXPERIMENTAL EVALUATION

In this section, we first describe the settings of experiments including the dataset, baseline approaches, and the evaluation method. After that, we report on major results on both the effectiveness and efficiency of our system followed by some discussions.

5.1 Experiment Settings

Datasets. We study the top two largest cities in USA, obtaining 221,128 *tips* generated by 49,062 users in New York City (NYC) and 104,478 *tips* generated by 31,544 users in Los Angeles (LA) from Foursquare. At the meantime, we collect these users' *tips* in other cities so as to model a user's preferences thoroughly. Foursquare blocked the API for crawling a user's *check-in* data due to the privacy concern, but leaving tips open to download. Our method could be more effective if using *check-in* data (though it is not bad using the *tips*). On the other hand, *tips* have their own advantages in reflecting a user's real interests. Some-times, people check in at a venue without doing anything at the venue. But, leaving a *tip* in a venue usually means a user has carried out some essential activities (like dinning and shopping) at the venue.

The following information is recorded when collecting the data: 1) user profile information, including the user ID, name, and home city; 2) venue profile information, consisting of a venue's ID, name, address, GPS coordinates, and its categories; and 3) user location histories, represented by all the *tips* a user left in the system. Each *tip* is associated with a venue ID, comments and a timestamp. From the dataset we collected, we choose the users whose home city is located in New Jersey (NJ) state and study the location recommendations made for these users in NYC and LA respectively. To guarantee the validity of the experimental results, we further select the user who has over 8 *tips* in a city as a candidate query user. Table 1 shows the details about these NJ users, where the footprint range denotes the average diagonal distance of the minimal bounding box of the locations visited by the user in the querying city. The data presented in Table 1 tells two stories. First, users have more opportunities traveling to nearby locations, thereby generating more *tips* in total in a nearby city than a distant one. Second, users who visit LA traveled in a large range than those visiting NYC. This is in line with the fact that LA is larger than NYC geographically.

Home City	Querying City	Total Users	Tips in City	Tips /User	Footprint (miles)	All Tips
NJ	LA	228	2,553	11.20	5.31	9,836
NJ	NYC	2,886	72,170	25.01	3.93	106,870

Table 1: Statistics of Experimental Data Set.

Baseline approaches. We compare our method with the following three baseline approaches, detailed in Table 2, where the first three baseline approaches are the existing recommender systems and the fourth one (ours w/o CS) means our method without using the preference-aware candidate selection algorithm.

1) *Most-Preferred-Category-based (MPC) recommendation.* Given a user-specified geospatial range and the user's WCH, this approach chooses the top-N venues as the final recommendations based on an iterative inference model, which is similar to [29]. As compared with our method, this approach does not consider local users' opinions on the recommended locations.

2) *Location-based Collaborative Filtering (LCF).* Location-based Collaborative Filtering (LCF) is the most common way that people would come up with [24], which applies the collaborative filtering method directly over the venues. This baseline utilizes the users' location histories in a city with a user-venue matrix (an entry denotes the number of visits of a user to a venue) and applies the traditional user-based CF method to make recommendations. The Cosine similarity between two users' location vector is employed as the similarity between the two users, and the inference is performed offline. Finally, the locations in the user-specified range and having a relatively inference score will be recommended.

3) *Preference-based Collaborative Filtering (PCF).* This baseline first retrieves all the users and venues in the user-specified range, formulates a user-venue matrix online, and then applies a user-based CF model to predict a user's rating of a venue. This approach starts considering the opinions from other users. However, the similarity between two users is represented by the Cosine similarity between the category vectors corresponding to the two users (without considering the category hierarchy).

Method	Social Opinion	Category of Location	Preference Hierarchy	Candidate Selection
MPC		√	√	√
LCF	√			
PCF	√	√		
Ours w/o CS	√	√	√	
Ours	√	√	√	√

Table 2: Comparison Between Baseline Methods and Ours.

Evaluation methods. We evaluate both the effectiveness of the suggested recommendations and the efficiency for generating online recommendations with the baseline solutions.

1) *Recommendation effectiveness.* It is very difficult to carry out a large-scale in-the-field study for evaluating the effectiveness of the location recommendations. To make the effectiveness evaluation, we divide a user's location history into two parts: 1) we select the location history generated in a querying city as a test set and 2) we use the rest of the user's location history as a training set for us to learn the user's preferences. We regard the venues that a user has visited in the querying city as the ground truths and match the recommended locations against these venues. The more recommended locations truly visited by a user in the test city, the more effective the recommendation method is. Specifically, as shown in the left part of Figure 8, the black dots are the venues the user actually visited, and we regard the minimum bounding box of all the

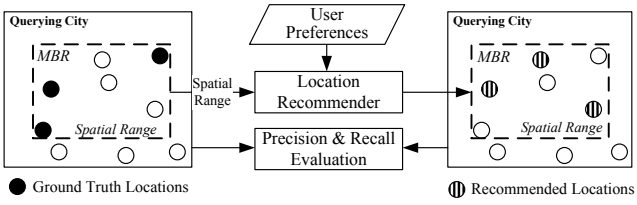


Figure 8: Recommendation Effectiveness Evaluation Method.

visited venues in the querying city to simulate the geospatial range that would be specified in the user's recommendation request. Remember that our recommendation system is location-aware, i.e., a spatial range is needed here to evaluate the effectiveness. Then, based on the given geospatial range and the user's location history, some venues will be recommended by our system, as illustrated by the striped dots in the right part of Figure 8. Based on the ground truth and recommendations, we are able to compute a precision and recall according to Equation 8 and 9.

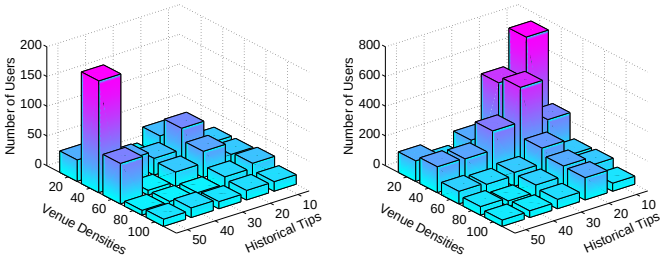
$$precision = \frac{number\ of\ recovered\ ground\ truths}{total\ number\ of\ recommendations} \quad (10)$$

$$recall = \frac{number\ of\ recovered\ ground\ truths}{total\ number\ of\ ground\ truths}. \quad (11)$$

In fact, this is a very strict evaluation measurement as a user may still like a venue even if the user did not visit the venue. Or, a user has visited a location while the user forgot to leave *tips*. In other words, our method is actually more effective than the number shown in the following experimental results. Meanwhile, the results still reveal the advantages of our method beyond baselines from the perspective of a relative comparison.

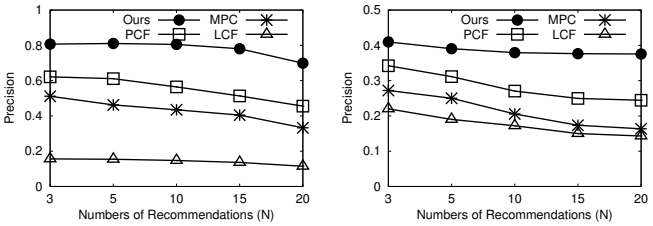
The precision and recall are affected by the following three major factors: 1) the number of requested recommendations N, 2) the scale of a user's location history (i.e., the number of visited locations, including locations outside a querying city), and 3) the density of venues with *tips* in a user's query range (for simplicity termed as venue density). For example, the venue density shown in the left part of Figure 8 is 6 (if the size of the bounding box is 1 $mile^2$). Therefore, in the rest of the paper, we study the effectiveness of our system changing over these three factors, using the NJ users' data shown in Table 1. Figure 9 respectively illustrates the distributions of the NJ users in LA and NYC with respect to the scale of location history and the venue density (the number of venues with *tips* per $mile^2$).

2) *Recommendation efficiency.* The efficiency of the online recommendation mainly depends on the following two aspects: a) the size of the user-specified geospatial range and b) the number of



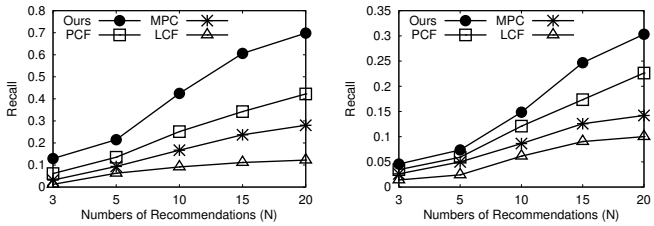
(a) New Jersey Users in LA. (b) New Jersey Users in NYC.

Figure 9: User Location History Distributions.



(a) New Jersey Users in LA. (b) New Jersey Users in NYC.

Figure 10: Precision w.r.t Recommendation Numbers.



(a) New Jersey Users in LA. (b) New Jersey Users in NYC.

Figure 11: Recall w.r.t Recommendation Numbers.

venues recommended. Therefore, we test the efficiency of our system changing over these two factors. At the same, we explore the benefit the candidate selection component brings to the system.

5.2 Experimental Results

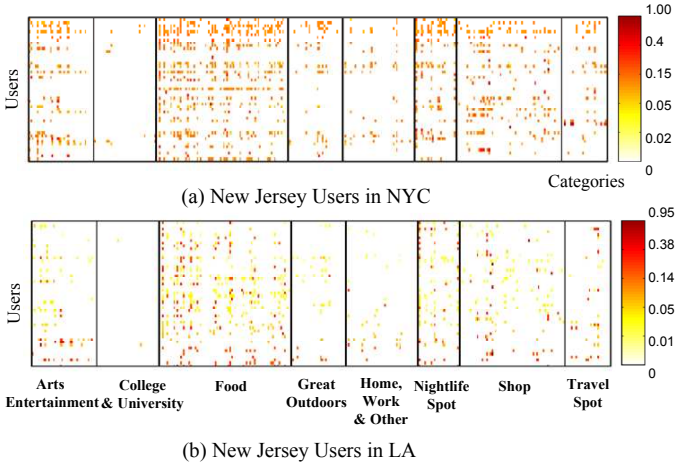
5.2.1 *Effectiveness of Recommendations*

Figure 10 and 11 show the average precision and recall of different methods varying in the number of recommended locations (N). Clearly, our method outperforms baseline approaches significantly. First, LCF drops behind other three methods, showing the advantage of using location categories to model a user's location history and carrying a location-dependent inference. Second, PCF and our method outperform MPC, justifying the benefit brought by considering social opinions. Third, our method exceeds PCF due to the advantages of WCH, which is more capable of modeling a user's preferences. Finally, our method has a very similar performance between using and without using the candidate select algorithm, as shown in Table 3 (we did not plot it on Figure 10 and 11, as the difference is minor). This is a good result as the candidate selection improves the efficiency of our method (see later results) significantly while having the same (or even better) effectiveness as (or than) using the full set of locations falling in a user-specified geospatial range.

As shown in Figure 10 and 11, the recall of our method increases quickly though the precision drops slightly as the number of recommendation increases. Our method achieves the best performance when N=15 in LA (*F-measure*=0.771), and N=20 in NYC (*F-measure* =0.385), where F-measure=$2\times\frac{precision\times recall}{(precision+recall)}$. In addition, the precision in LA is higher than that of NYC though NJ users have more location histories in NYC beyond LA. In other words, the venues to be visited by a user are more predicable when the

Method	Precision			Recall		
	N=**5**	N=**10**	N=**20**	N=**5**	N=**10**	N=**20**
Ours	0.80	0.79	0.71	0.21	0.42	0.70
Ours w/o CS	0.81	0.80	0.70	0.21	0.42	0.68

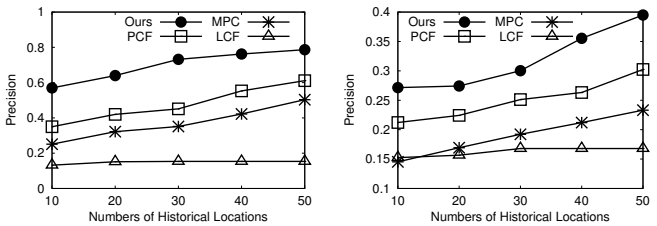
Table 3: Comparison of *Ours* & *Ours w/o CS* (NJ users in LA).



(a) New Jersey Users in NYC

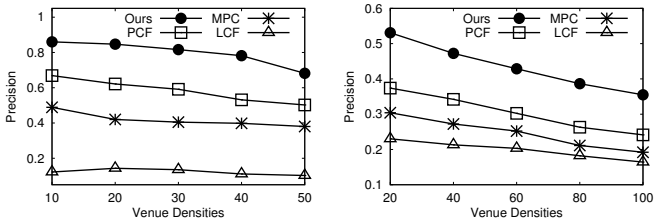
(b) New Jersey Users in LA

Figure 12: Category distributions of Top-50 NJ users.



(a) New Jersey Users in LA. (b) New Jersey Users in NYC.

Figure 13: Precision w.r.t Scales of Location Histories.



(a) New Jersey Users in LA. (b) New Jersey Users in NYC.

Figure 14: Precision w.r.t Venue Densities.

user travels to a new city. This seems somehow surprising at first glance. However, we found it is true given the following fact: People usually visit some well-known places (e.g., tourist attractions or restaurants introduced in a travel guide book) in a new city to them, while would travel to any venues in a city they are very familiar with (e.g., hometown). This is also one of the reasons leading to a lower recall in NYC. Besides that, NJ users have visited more locations in NYC, causing a bigger denominator in Equation 9 which further reduces the recall. Figure 12 further justifies this claim by visualizing the distribution of a user's location history in different categories (in LA and NYC respectively). Here, each row (line) represents a user and each column (line) denotes a category. We select the top-50 users with the largest scale of location history, ranking them from the top to the bottom in the figure. Meanwhile, we group the sub-categories belonging to the same category by a set of separators on the horizontal axis (refer to Figure 2 b)). Clearly, these users' location histories are more focused in LA than in NYC (as NYC is much closer to New Jersey than LA), therefore easy to predict. It is similar to the discovery in [5] that a long-distance travel is more influenced by the social network ties.

To further explore the performance of our method, Figure 13 presents the precision of different methods changing over the scale of a user's locations history (where a user requests 10 recommendations, i.e., N=10). As a result, the more locations that a user has visited the more accurate we can model a user's preferences, thereby leading to a better performance. Additionally, the precision of the other three methods increases faster beyond LCF as the number of visited location increases, showing the advantage of location category in dealing with data sparseness problem. Similar to Figure 10, the precision in LA is still higher than NYC.

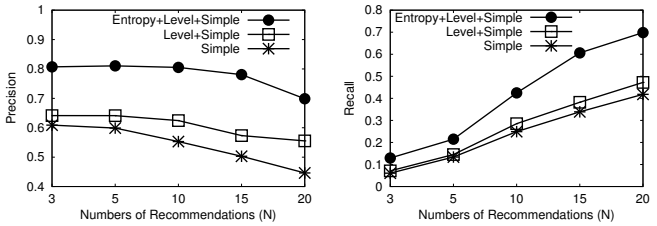
Figure 14 plots the precision of different methods changing over the venue density. The results match our intuition that the denser venues located around a user the more location candidates can be recommended. Therefore, the prediction becomes harder and then the precision decreases. Actually, to guarantee the quality of recommendations, our system can help a user smartly determine the number of venues that should be recommended based on the scale of her location history and the venue density around. In this way, a user does not need to do anything when using our system.

Figure 15 further studies the user similarity function (using the defined precision and recall criteria), justifying the advantage of each component we defined in Equation 8. Here, "Simple" denotes the user similarity solely considering the overlapped nodes between two users' WCHs (i.e., Equation 6). "Simple+Level" means the similarity taking into account both the overlapped nodes and the granularity of a WCH (nodes on a deeper level are assigned with a bigger β). Finally, "Simple + Level + Entropy" is the similarity we defined in Equation 8. The results show the benefit by adding each component to our similarity function. In addition, the entropy of a WCH brings a significant improvement.

5.2.2 *Efficiency of Recommendations*

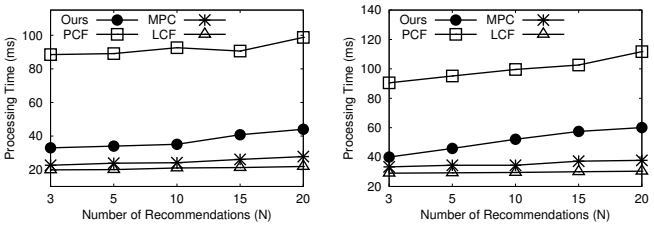
In the efficiency study, we test 200 users in LA and NYC respectively, randomly choosing a location in the city for the user. The experiments were evaluated on a computer running Windows 7 with an Intel Xeon CPU 2.80GHz processor and 24 GB RAM.

Figure 16 presents the average online efficiency of different methods varying in the number of recommendations, setting 10 miles as a query range. For example, on average our method can find top-10 location recommendations (that could interest a user most) within a distance of 10 mile (to a user's current position) in 40ms in LA and about 60ms in NYC. It is not surprising that our method is slower than MPC which does not consider the location history of other users. LCF achieves the best efficiency because we do not count the time for the CF-based inference (which is supposed to be carried out offline). Theoretically, no method can outperform LCF in efficiency as it only does an online selection (of course, the effectiveness of LCF is the worst among these approaches). But, our method is faster than PCF due to the candidate selection algorithm, and is not significantly slower than MPC and LCF. The processing time only increases slightly as the number of recommendations increases. Additionally, the online recommendation only costs a little bit more time in NYC (than LA) though the venue density in NYC



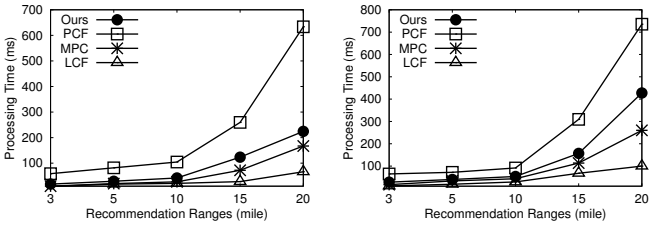
(a) Precision (NJ users in LA). (b) Recall (NJ users in LA).

Figure 15: Similarity Functions w.r.t Recommendations.



(a) Processing time in LA. (b) Processing time in NYC.

Figure 16: Efficiency w.r.t Recommendations (R=10 miles).



(a) Processing time in LA. (b) Processing time in NYC.

Figure 17: Efficiency w.r.t Spatial Ranges (N=10).

is higher than LA.

Figure 17 shows the average efficiency of different approaches changing over the geospatial range specified by a user, setting N=10. Intuitively, a larger range will incorporate more location and user candidates, leading to a heavier computational load. But, we find the similar trends as that shown in Figure 17 (LCF $>$ MPC $>$ ours $>$ PCF). As people would not request location recommendation far away from them, we only study the efficiency up to 20 miles. Overall, our method is efficient and scalable, besides the effectiveness we have justified before.

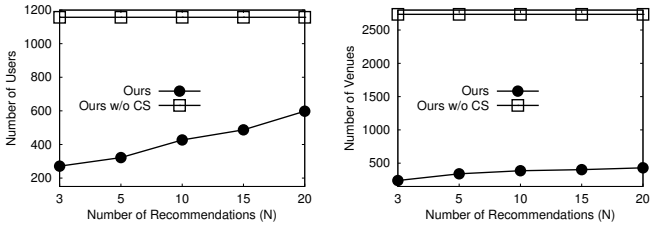
To explore the benefit brought by the candidate selection algorithm, we further study the difference between using and without using the candidate selection algorithm. Figure 18 a) and b) respectively present the number of users and that of locations chosen for the CF model, varying in number of recommendations (setting range R=10 miles) For instance, our method with candidate selection only employs 1/3 users and 1/5 location candidates for generating 10 location recommendations, which is as good as using the full set. In addition, the smaller number of recommendations requested, the more inexperienced users and low quality locations removed. Figure 19 a) and b) respectively plot the number of users and that of locations chosen for the CF model, changing over the size of the user-specified geospatial range. As a result, the larger range a user specifies, the more inexperienced users and low quality locations our candidate selection algorithm removes. In short, the candidate selection algorithm improves the efficiency of our system significantly while maintaining the effectiveness.

6. RELATED WORK

We summarize the existing location recommendations into two categories: 1) generic location recommendations and 2) personalized location recommendations.

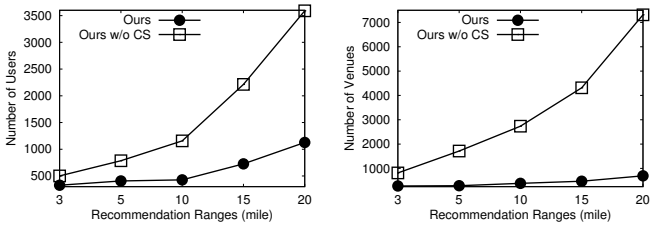
6.1 Generic Location Recommendations

Regardless of the preferences of an individual, generic location recommendation systems encapsulate the public opinions on locations to provide people with the most popular venues or travel routes in a city. For example, [29] mines the most interesting locations and travel sequences from a large number of user-generated GPS trajectories. Given a user-location matrix, a HITS-based inference model was also proposed to predict the interest level of a physical location and the knowledge of a user. [3] further extends this work by considering the correlation between locations when doing the inference. However, both of them do not differentiate the locations from different categories. Though these recommendation systems have their own applications, sometimes, it would be difficult to say which one is more interesting, a shopping mall or a museum, as different users may have different answers.



(a) Selected Users in LA. (b) Candidate Venues in LA.

Figure 18: Candidates w.r.t Recommendations (R=10 miles).



(a) Selected Users in LA. (b) Candidate Venues in LA.

Figure 19: Candidates w.r.t Spatial Ranges (N=10).

6.2 Personalized Location Recommendation

Some simple personalized recommendation systems request a user to manually specify her personal interests by categories (like restaurants and parks) [11, 18], which will be employed to determine the POIs (around the user) to be shown on a mobile interface. As a user's preferences are not actually binary decisions and have a certain granularity, manually specifying personal preferences is obtrusive and usually bring a user too many or too few recommendations. Meanwhile, such systems do not incorporate other users' opinions on a venue, losing a lot of valuable information.

A branch of recent research starts learning a user's interests from the user's location history and incorporates the social environment of the user to make recommendations. Specifically, [6, 14, 23, 22] deposit people's location histories into a user-location matrix where a row corresponds to a user's location history and each column denotes a venue like a restaurant. Each entry in the matrix represents the number of visits of a particular user to a physical venue. Then, a user-based CF model is employed to infer a user's interest to an unvisited venue. However, the similarity between two users is simply represented by the Cosine similarity between the two users' rows, overlooking the features of human mobility in geographic spaces, such as sequential and hierarchical properties of locations. To better estimate the similarity between users, Zheng et al. [28] proposed a hierarchical-graph-based similarity measurement taking the human mobility features into account. The location recommendation system using the user similarity outperforms those using the Cosine similarity. While the user-based CF model is able to capture people's mobility in the physical world, it has a poor scalability as adding a new user into a system will trigger a large number of similarity computing operations. To address the problem of scalability, [26] proposed a location-based CF model using

the location correlation mined from many users' GPS traces as a distance measure between two locations. The location-based CF model is slightly less effective than the user-based one while being much more efficient.

Unfortunately, solely using a CF model (no matter the user-based or the location-based) cannot handle the data sparseness problem very well if we directly formulate a user-location matrix. Though [15, 24] applied Single Value Decomposition to a user-location matrix so as to reduce the data sparseness problem to some extent, this method does not work well when there is no overlap between users' location histories. In fact, this is quite common when an individual travels to a city that is new to her.

Our recommendation system differs from the above-mentioned work in the following two aspects: 1) We project a user's location history into the category space and model a user's preferences using a WCH. This method handles the data sparseness problem and enables the computing of similarity between users who do not share any physical location histories, e.g., living in different cities. Unlike the traditional cold-start problem in the recommender system [21, 1], where the users or items come to the system with no ratings, a user is new only for the unfamiliar area in terms of the new city problem in location-based recommendation. As we take advantage of the category information of the user's historical location, we can recommend locations to a user in a city based on her location history in other cities. 2) Pervious CF-model based methods have to infer a user's interests in a venue offline due to the heavy computation and then present the locations with a high ranking around a user. Such methods cannot guarantee the quality of the recommended locations as a user's current location is not truly incorporated in the inference. But, our system chooses candidate venues according to a user's current location (or any location specified by a user) and carries out the inference online. So, the venues recommended by our system are not only preference-aware but also really location-based.

7. CONCLUSION

This paper presents a location-based and preference-aware recommender system, which provides a user with location recommendations around the specified geo-position based on 1) the user's personal preferences learnt from her location history and 2) social opinions mined from the *local experts* who could share similar interests. This recommender system can facilitate people's travel not only near their living areas but also to a city that is new to them (even if they have not visited any places there). By taking advantage of the category information of a user's location history, our system overcomes the data sparsity problem in the original user-location matrix. We evaluated our system using extensive experiments based on a real data set (221,128 *tips* generated by 49,062 users in NYC and 104,478 *tips* generated by 31,544 users in Los Angeles) collected from Foursquare. According to the experimental results, our approach significantly outperforms some major location recommendation methods (MPC, LCF, and PCF) in effectiveness (measured by precision and recall). The results also justify each component proposed in our system, e.g., taking into account location history of others, category-hierarchy based preference modeling, user similarity computing, and CF-based inference. Meanwhile, the proposed candidate selection algorithm improves the efficiency of our approach tremendously while maintaining the effectiveness, enabling an online recommendation scenario. In general, our system can provide 10 quality location recommendations within a 10-mile spatial range within 60ms. In the future, we are going to incorporate the temporal and weather features into the recommendation system.

8. REFERENCES

[1] H.J. Ahn. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences*, 178(1):37–51, 2008.

[2] X. Amatriain, N. Lathia, J.M. Pujol, H. Kwak, and N. Oliver. The wisdom of the few: a collaborative filtering approach based on expert opinions from the web. In *ACM SIGIR*, pages 532–539. ACM, 2009.

[3] X. Cao, G. Cong, and C.S. Jensen. Mining significant semantic locations from gps data. *Proceedings of the VLDB Endowment*, 3(1-2):1009–1020, 2010.

[4] S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and J. Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. *Computer Networks and ISDN Systems*, 30(1-7):65–74, 1998.

[5] E. Cho, S.A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD*, pages 1082–1090. ACM, 2011.

[6] Chi-Yin Chow, Jie Bao, and Mohamed F. Mokbel. Towards Location-based Social Networking Services. In *The 2nd Workshop on LBSN*, 2010.

[7] J. Cranshaw, E. Toch, J. Hong, A. Kittur, and N. Sadeh. Bridging the gap between physical location and online social networks. In *Ubicomp*, pages 119–128. ACM, 2010.

[8] J.L. Herlocker, J.A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR*, pages 230–237. ACM, 1999.

[9] Tzvetan Horozov, Nitya Narasimhan, and Venu Vasudevan. Using location for personalized poi recommendations in mobile environments. In *SAINT*, pages 124–129, 2006.

[10] J.M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[11] K. Kodama, Y. Iijima, X. Guo, and Y. Ishikawa. Skyline queries based on user locations and preferences for making location-based recommendations. In *LBSN*, pages 9–16. ACM, 2009.

[12] J.A. Konstan, B.N. Miller, D. Maltz, J.L. Herlocker, L.R. Gordon, and J. Riedl. Grouplens: applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

[13] C.H. Lee, Y.H. Kim, and P.K. Rhee. Web personalization expert with combining collaborative filtering and association rule mining technique. *Expert Systems with Applications*, 21(3):131–137, 2001.

[14] J. Levandoski, Mohamed Sarwat, Ahmed Eldawy, and Mohamed Mokbel. Lars: A location-aware recommender system. In *ICDE*, 2012.

[15] H. Ma, H. Yang, M.R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM*, pages 931–940. ACM, 2008.

[16] Mohamed Mokbel, Jie Bao, Ahmed Eldawy, Justin Levandoski, and Mohamed Sarwat. Personalization, Socialization, and Recommendations in Location-based Services 2.0. In *PersDB*. VLDB, 2011.

[17] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil. Exploiting semantic annotations for clustering geographic areas and users in location-based social networks. In *Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[18] M.H. Park, J.H. Hong, and S.B. Cho. Location-based recommendation system using bayesian user's preference model in mobile devices. *Ubiquitous Intelligence and Computing*, pages 1130–1139, 2007.

[19] S. Robertson. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of Documentation*, 60(5):503–520, 2004.

[20] S. Scellato, A. Noulas, R. Lambiotte, and C. Mascolo. Socio-spatial properties of online location-based social networks. *Proceedings of ICWSM*, 11, 2011.

[21] A.I. Schein, A. Popescul, L.H. Ungar, and D.M. Pennock. Methods and metrics for cold-start recommendations. In *SIGIR*, pages 253–260. ACM, 2002.

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

[23] M. Ye, P. Yin, W.C. Lee, and D.L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*, pages 325–334. ACM, 2011.

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

[25] Y Zheng. Location-based social networks: Users. In *Computing with Spatial Trajectories, Zheng, Y and Zhou, X, Ed.* Springer, 2011.

[26] Y. Zheng and X. Xie. Learning travel recommendations from user-generated gps traces. *ACM TIST*, 2(1):2, 2011.

[27] Y. Zheng, X. Xie, and W.Y. Ma. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Engineering Bulletin*, 33(2):32–40, 2010.

[28] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.Y. Ma. Recommending friends and locations based on individual location history. *ACM Transactions on the Web*, 5(1):5, 2011.

[29] Y. Zheng, L. Zhang, X. Xie, and W.Y. Ma. Mining interesting locations and travel sequences from gps trajectories. In *WWW*, pages 791–800. ACM, 2009.