International Journal of Networking and Computing – www.ijnc.org
ISSN 2185-2839 (print) ISSN 2185-2847 (online)
Volume 3, Number 2, pages 228–243, July 2013

Place Recommendation Based on Users Check-in History for Location-Based Services

Hongbo Chen, Mohammad Shamsul Arefin, Zhiming Chen, Yasuhiko Morimoto
Graduate School of Engineering, Hiroshima University, Kagamiyama 1-7-1,
Higashi-Hiroshima, Hiroshima, 739-8521, Japan

Received: February 20, 2013
Revised: May 14, 2013
Accepted: June 18, 2013
Communicated by Yasuaki Ito

Abstract

With rapid growth of the GPS enabled mobile devices, location-based online social network services become very popular, and allow their users to share life experiences with location information. In this paper, we considered a method for recommending places to a user based on spatial databases of location-based online social network services. We used a user-based collaborative filtering method to make a set of recommend places. In the proposed method, we calculate similarity of users' check-in activities based on not only their positions but also their semantics such as "shopping", "eating", "drinking", and so forth. We empirically evaluated our method in a real database and found that the proposed method outperforms the naive singular value decomposition collaborative filtering in its recommendation accuracy.

Keywords: Recommender systems, Collaborative filtering, Place recommendation

1 Introduction

During past few years, online social network services such as Facebook and Twitter have become very popular. Around half a billion users around the world are using these services [5]. At the same time, with the growing prevalence of GPS enabled mobile devices, such as smart phones like Apple iPhones and Google Android phones; Location-based Online Social Network Services (LOSNS) have become extremely popular, which help users to create, share, and communicate with other users by attaching their geographical location information. By using "check-in" function of LOSNS, users can easily publish their feelings and expressions with spatio-temporal information. For instance, "Facebook Places" is one of these kinds of check-in services.

Through the LOSNSs, we can collect a spatio-temporal database containing hundreds of millions of users' check-in records. By analyzing the database, we can extract information such as what the users are interested in, where the users like to go, and what the users often do. In this paper, we considered a method for recommending places to a user based on the spatial database of LOSNS. We used a user-based collaborative filtering method to make a set of recommend places for a user, in which we calculate similarity of users and use the similar users' records to predict places the user likes. In the proposed method, we calculate similarity of users' check-in activities based on both their positions and semantics.

1.1 Motivating Example

Most of the existing recommender systems simply perform collaborative filtering for finding similar users and perform recommendation accordingly. However, simple collaborative filtering method is not suitable for proper recommendation, since it does not use any mechanism to get the semantics of the users' activities. As a result, the recommendation for the user contains many irrelevant places that the user may not be interested

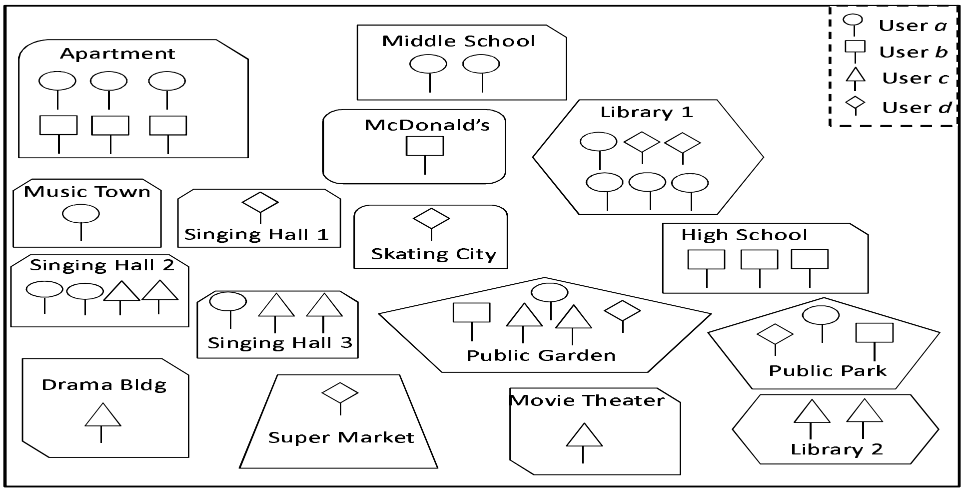

Figure 1: Users check-in spots

in. In our proposed system, we associate users' check-in locations with a Points Of Interest (POIs) database to get semantics of each check-in location such as "shopping", "eating", "drinking", and so forth.

Now, consider four users' check-in history as shown in the map of Figure 1. We use "pin" to stead for user's check-in spot. Each user has different shape of "pin" in the figure. From the map, we can find that the users have visited in sixteen different places as represented in the map. Table 1 summarizes the detail history of their visiting information.

At first, we use an existing recommender system like Collaborative Filtering (CF) systems to deal with this problem. The CF systems predict the utility of items for a particular user based on the items that are rated by other users who are similar to the user. In the place recommendation problem, the utility $t(u, p)$ of place p for user u is estimated based on the utilities $t(u_j, p)$ assigned to place p by those users $u_j \in U$, who are similar to user u. Let us consier the place recommendation for user a. From Table 1, we can find that user b is similar to user a since the number of check-in in "Public Park", "Public Garden", and "Apartment" are the same. CF system recommends "High School" and "McDonald's" for user a based on the check-in information of the similar user b. However, from the check-in information of user a in Table 1, we can observe that the provided recommendation is not well suited for user a, since user a is interested in entertainment facilities such as "Music Town", and "Singing Hall."

On the other hand, our proposed framework performs several intelligent calculation and recommends "Drama Bldg" and "Movie Theater" by analyzing data in Table 1 to user a who seems to like entertainment facilities by analysing semantics of user's check-in.

By taking into account of semantics of the check-in information proposed method can recommend places that conventional CF systems cannot.

The proposed method works as follows:

- We, first, make clusters of the check-in information based proximity.
- Next, for each cluster, we calculate the gravity center of each cluster to represent the position of the cluster. We, then, annotate each cluster by using POIs database.
- Next, we perform semantic analysis of the users' interests and measure the similarity score among the users.
- Finally, we select Top-N similar users and use the Top-N users' records for the recommendation.

In this paper we propose an efficient method of recommendation that can overcome the limitations of CF based recommender systems and can provide more accurate recommendation.

The remaining paper is structured as follows:

Table 1: Users' check-in information

Place	User *a*	User *b*	User *c*	User *d*
Music Town	1	0	0	0
Singing Hall 1	0	0	0	1
Singing Hall 2	2	0	2	0
Singing Hall 3	1	0	2	0
Library 1	4	0	0	2
Library 2	0	0	2	0
Public Park	1	1	0	1
Public Garden	1	1	2	1
Super Market	0	0	0	1
Drama Bldg	0	0	1	0
Movie Theater	0	0	1	0
McDonald's	0	1	0	0
Skating City	0	0	0	1
Apartment	3	3	0	0
Middle School	2	0	0	0
High School	0	3	0	0

In section 2, we discuss some related works on recommender systems. We provide some preliminary concepts related to our work in section 3. Section 4 briefly describes our method of recommendation. Section 5 gives experimental analysis of our method. Finally, we conclude and sketch future research direction in section 6.

2 Related Work

Location recommendation is a popular service in social network and at present, there are many systems for providing location recommendation.

Adomavicius et al. [1] first introduced the idea of collaborative filtering based recommendation. In collaborative filtering based approach, an item is recommended to a user based on past information of the people with similar tastes and preference.

Zheng et al. [7, 8, 9, 10] proposed several techniques for providing recommendations to the users. In [7], they introduced a system by using the location data based on GPS and users' comments to discover interesting locations for recommendations. They proposed two types of travel recommendations by mining multiple users' GPS traces in [8]. The first type tried to recommend a user with top interesting locations in a given geospatial region, and the second type is a personalized one that provides an individual with locations matching the user's travel preferences. Their work in [9] introduced a social network service, called GeoLife, which can understand trajectories, locations and users, and mine the correlation between users and locations in terms of user-generated GPS trajectories. In [10], they moved their direction towards on a personalized friend and location recommender for the geographical information systems on the web. In their paper, they built a hierarchical graph to model each user's location history and measure the similarity among users.

Although the works have some similarities with our work, the major difference is that their frameworks paid more attention at users' physical location, while our proposed framework focused on both users' locations and interests.

Berjani et al. [4] proposed a personalized recommender for location-based online social networks, based on the check-in of the users. They crawled check-in information to generate a user / spot rating matrix. By predicting the interest of users in certain spots, their technique recommends places they have not been visited previously. This approach is useful to find the interest of users in certain spots. Takeuchi et al. [6] proposed a novel real-world recommender system based on users individual preferences and needs. This system automatically figures out each user's frequently visited spots and makes recommendations by using the spots as input to the item-based collaborative filtering algorithm. McDonald et al. [12] proposed an expert locating system that recommends people for possible collaboration within a work place. Chow et

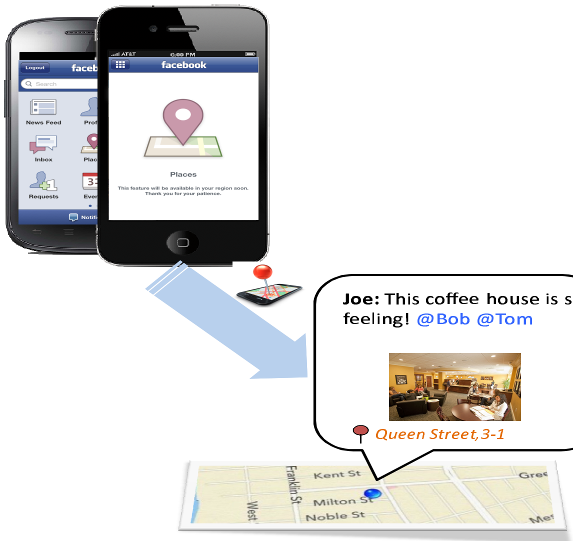

Figure 2: Social check-in Example

Table 2: Check-in spots records

UserID	Name	MessageID	Content	Coordinate	Location	PublishTime
187003160	Joe	201110308...	This coffee...	<23.116, 113.315>	Queen Street..	2011.10.30 8:23:15
1596423831	Tim	201110211...	I have a...	<31.093, 114.189>	Central Road..	2011.10.21 15:32:11
1886906022	Rose	201110201...	Tom and I...	<30.545, 114.312>	NL Street..	2011.10.20 17:33:55

al. [5] presented GeoSocialDB, a holistic system, providing location-based news feed, location-based news ranking, and location-based recommendation. Within their framework, they discussed the research challenges and directions towards the realization of location based social network services.

None of the above works considered the semantics of data. However, considering the semantics of data can provide more accurate recommendations. Considering this fact, in this paper, we provide a framework that considers the semantics of data along with the locations of users.

3 Preliminaries

3.1 Check-in Information of Social Network Services

Many social network services, such as Foursquare, Google Latitude and Facebook Places allow users to ”check in” to a physical place with their feeling and share their check-in with their friends.

Users can check in to a specific location by text messaging or by a mobile application on a smart phone. The application uses the phone’s GPS to find the current location. For example, in Figure 2, Joe checked in a coffee house with a picture and leaved a message to explain his feeling. The check-in information are stored in the database as shown in Table 2.

3.2 Recommender Systems

Recommender systems are software tools and techniques those try to provide suggestions to the users to help them in their decision-making processes. Recommender systems play an important role in highly rated Internet sites such as Amazon.com, YouTube, Netflix, and Yahoo. For example, Amazon.com employs a recommender system to assist the customers to select appropriate books while they are searching for some other books.

3.2.1 Place Recommender Systems

Due to recent increase in location-acquisition technologies, like GPS, the collection of large spatio-temporal datasets becomes easier. This brings the opportunity of discovering valuable knowledge from large spatio-temporal datasets and received considerable attention. From GPS data of an user, we can easily find his interests and preferences. For instance, if a person goes to stadiums and gyms frequently, it might imply that the person likes sports. Likewise, if a user frequently travels to some mountains, it might imply that the user is interested in hiking. People who have similar location histories might share similar interests and preferences. Based on similar location histories of the users, place recommender systems can provide suggestions about unvisited places to a user.

3.2.2 Collaborative Filtering

Collaborative Filtering (CF) [18] is one of the most popular techniques used in recommender systems. The CF methods produce user specific recommendations of items based on patterns of ratings or usage (e.g., purchases) without need for exogenous information about either items or users. In general, collaborative filtering is the process of filtering for information or patterns using techniques involving collaboration among multiple agents, viewpoints, data sources, etc.

CF systems try to predict the utility of items for a particular user based on the items previously rated by other users. More formally, the utility $u(c,s)$ of item s for user c is estimated based on the utilities $u(c_j,s)$ assigned to item s by those users $c_j \in C$, who are similar to user c. For example, in a movie recommender application, in order to recommend movies to user c, the collaborative recommender system tries to find the "peers" of user c, i.e., other users those have similar tastes in movies (rate the same movies similarly). Then, only the movies that are most liked by the "peers" of user c would be recommended.

Algorithms of CF essentially are heuristics that make rating predictions based on the entire collection of previously rated items by the users. That is, the value of the unknown rating $r_{c,s}$ for user c and item s is usually computed as an aggregate of the ratings of some other (usually, the N most similar) users for the same item s:

$$r_{c,s} = aggr_{c' \in \hat{C}} r_{c',s} \tag{1}$$

where $\hat{C}$ denotes the set of N users that are the most similar to user c who have rated item s (N can range anywhere from 1 to the number of all users).

3.2.3 SVD-CF

Singular Value Decomposition (SVD) [19] is a factorization of a real or complex matrix in linear algebra. It has many useful applications in signal processing and statistics.

SVD is a well-known matrix factorization technique that factors an $n \times m$ matrix A into three matrices as $A = USV^T$ where U and V are two orthogonal matrices of size $n \times y$ and $m \times y$, respectively; y is the rank of the matrix A. S is a diagonal matrix of size $y \times y$ having all singular values of matrix A as its diagonal entries. It is possible to reduce the $y \times y$ matrix S to have only k largest diagonal values to obtain a matrix S_k, $k \leq y$.

If the number of customers and products increase, the performance of CF systems become worse. SVD can reduce the dimensionality of filtering databases and can prevent the performance decrease of CF in the following way.

First, SVD captures relationships between customers and products that can compute the likeliness of a certain product by a customer. Second, it produces a low-dimensional representation of the original customer-product space and then computes neighborhood in the reduced space. Finally, SVD generates a list of top-N products as recommendations for the customers.

In SVD-CF, the sparse user-item ratings matrix (A) is filled using the ratings for users to capture a meaningful latent relationship. The filled matrix is normalized by converting ratings to z-scores. The normalized

Table 3: Point of interest database

GPS	Location	Name	Category	Father Category
< 23.116, 113.315>	Queen Street, 1-123, New York	Queen Coffee House	Coffee	Food
< 31.093, 114.189>	Central Road, 3-456, Tokyo	Super Market	Market	Shopping
< 30.545, 114.312>	NL Street, 5-32, Beijing	Shoudu Theater	Theater	Entertainment

matrix (A_{norm}) is factored into U, S, and V using SVD. Then the matrix S_k is obtained by retaining only k largest singular values. Accordingly, the dimensions of matrices U and V are also reduced. Then, $U_k\sqrt{S_k}$ and $\sqrt{S_k}V_k{}^T$ are computed. These resultant matrices can be used to compute the prediction for any user u on item q. To compute the prediction, the scalar product of the u^{th} row of $U_k\sqrt{S_k}$ (denoted as $U_k\sqrt{S_k}(u)$) and the q^{th} column of $\sqrt{S_k}V_k{}^T(q)$ is calculated and the result is denormalized as follows:

$$p_{uq} = \bar{v}_u + \sigma_u[U_k\sqrt{S_k}(u).\ \sqrt{S_k}V_k{}^T(q)] \tag{2}$$

where $\bar{v}_u$ and σ_u are mean rating and standard deviation for user u, respectively. p_{uq} is the prediction for any user u on item q.

3.3 Point of Interest

A Point of Interest (POI), is a specific point location that someone may find useful or interesting. POI is widely used in cartography, especially in electronic variants including GIS, and GPS navigation software. A GPS point of interest specifies, at minimum, the latitude and longitude of the POI, assuming a certain map datum. A name or description for the POI is usually included, and other information such as altitude or a telephone number may also be attached.

For instance, in Table 3, it gives some details of POI database information. By using these kinds of GPS detail data, some useful information like “category” and “father category” can be extracted for learning users’ interest that makes it easier to find similar users in our proposed method.

4 Development of the Place Recommender System

In this section, we explain detailed framework of our place recommender system. Proposed framework comprises three main components: interest discovery, similarity calculation, and recommendation generation.

4.1 Interests Discovery Using Semantic Hierarchical Category-Graph

To learn users’ interests, we need to analyze the check-in spots dataset.

There are variations of check-in spots and people visit these spots for different purposes. Some spots, such as entertainments, are frequently visited by people with interest, others, such as schools, are visited by people who have to attend class regardless of their interest almost daily. On the other hand, even if a person is interested in the place, he/she rarely visits there due to accessibility.

Similarly, there are different representations of check-in spot data based on the area of spots and the distribution of surrounding spots. In areas where spots are crowded like urban areas, even if the distance is very close to each other, people are likely stay separate places. On the other hand, in large spots, such as airport, even if the distance is very far, people are likely stay the same spots.

Considering above two facts, we make clusters of users’ check-in spots using a well-known clustering algorithm known as DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [20]. DBSCAN can obtain a number of clusters starting from the estimated density distribution of corresponding nodes. Examples of the clusters are shown in bottom layer of Figure 3. In the figure, C_1, C_2, ..., C_{15} are the clusters of the check-in spots.

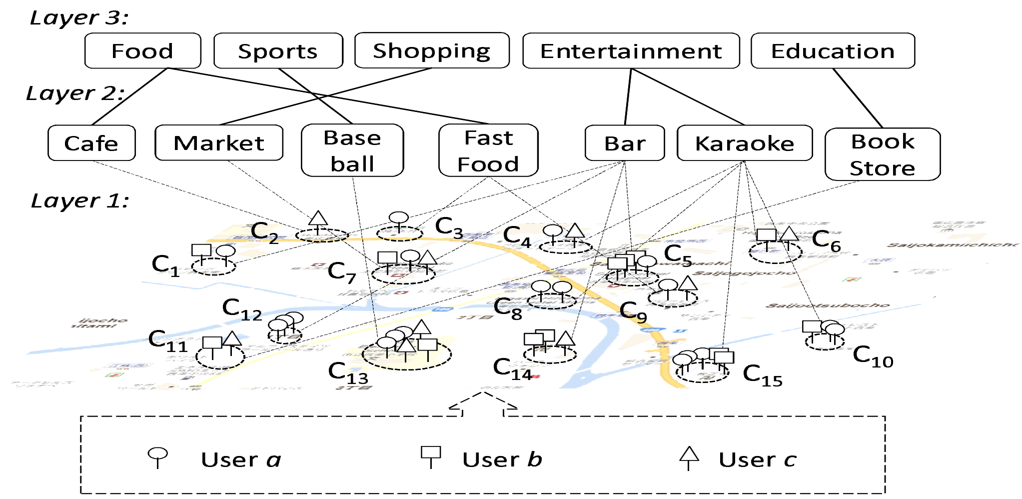

Figure 3: Semantic hierarchical category-graph shared framework

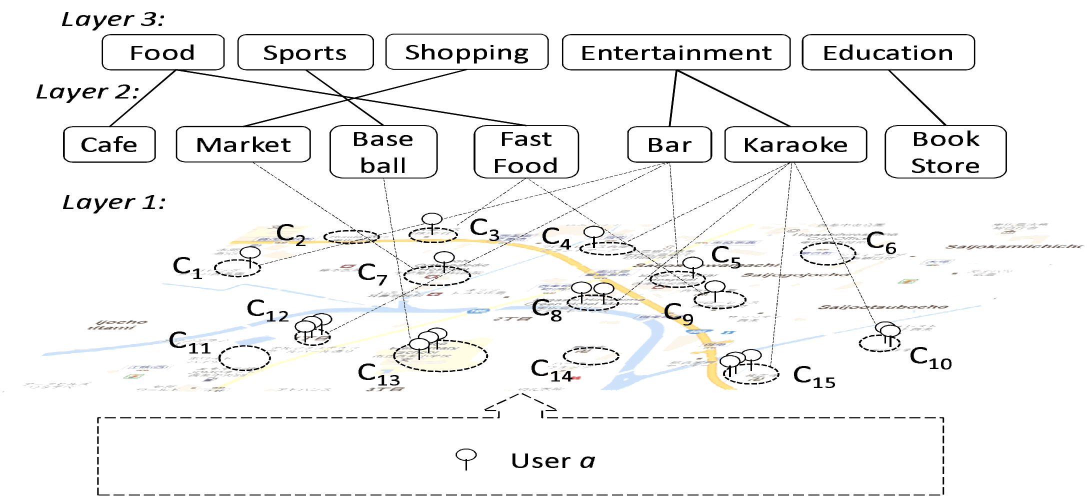

Figure 4: Semantic hierarchical category-graph personal framework

In the next step, for each cluster, we calculate the gravity center to represent the position of the cluster. We, then, annotate each cluster by using POIs database. If the nearest POI from the center of a cluster is a movie theater, which belong to entertainment category in the POI database, we annotate "movie theater" and "entertainment" to the cluster.

Then, we construct a Semantic Hierarchical Category-graph Framework (SHCF) as in Figure 3. As shown in Figure 3, this hierarchical framework contains three layers: In the bottom layer, which we call "layer 1", there are clusters of check-in spots. In the middle layer, which we call "layer 2", there are nodes of the annotated words. Each node contains corresponding clusters in layer 1. And, in the top layer, which is "layer 3", there are nodes of the POI categories. Each category node contains corresponding POI nodes of the layer 2.

The SHCF aggregates every users' check-in records. We call it a shared framework. After making the shared framework, we make a Semantic Hierarchical Category-graph by using the shared framework for each user. We call each user's framework "Personal Semantic Hierarchical Category-graph (PSHC)". Figure 4 shows such a graph for user a, whose check-in spots are represented by oval pins. In the Figure 4, clusters and categories are same as in Figure 3. By using POI database, we can find that clusters C_4, C_8, C_{10}, and C_{15} belong to Karaoke category and clusters C_1, C_5, C_{12} are within Bar category as shown in Figure 4. Therefore, we can imagine that user a is interested in Karaoke and Bar, and he / she might like entertainment very much.

Those hierarchical features are used to differentiate users in different degrees of similarity. For example, the users who share the check-in spots in same clusters on the bottom layer might be more correlated than those who share interests in same categories on a higher layer.

4.2 Similarity Calculation

Similarity calculation is an important step for predicting users' potential interests. In this step, we use SHCF as explained in previous section to calculate the similarity weight between users in each layer. Finally, we add the similarity weight of the layers to obtain similarity score.

4.2.1 Inverse Document Frequency

In order to normalize the popularity of different locations, we use the idea of "inverse document frequency" (IDF) [21] that is used for document analysis. If the location "A" is not popular and visited by a few people and the location "B" is popular and many people visited, then the co-occurrence of "A" is stronger than that of "B" for calculating similarity in the idea of "inverse document frequency".

A cluster can be regard as a document, while the users who have checked in this cluster can be regarded as terms. If a large number of users (n_i) that checked in a cluster (c_j), the $IDF_j = log\frac{|U|}{n_i}$ of this cluster would become very small. Therefore, this cluster will not offer many contributions to the similarity score of these two users on the cluster layer.

4.2.2 Similarity Weight Measure of Bottom Layer

To measure the similarity between two users in layer 1, we use a Pearson Correlation approach [22], which is popularly used in similarity calculation in recommender systems.

Suppose $N = |C_j|$ clusters are generated on the bottom layer (cluster layer) in our SHCF.

Assume that two users u_a and u_b visited m_j and m'_j times in a cluster C_j. Let w_j and w'_j be a weight value of u_a and u_b to C_j. We can compute the weight w_j and w'_j with the equations (3) and (4).

$$w_j = m_j * IDF_j, \tag{3}$$
$$w'_j = m'_j * IDF'_j, \tag{4}$$

So, two vectors can be respectively constructed for U_a and U_b as follows,

$$U_a = < w_1, w_2, \ldots, w_j, \ldots, w_N >, \tag{5}$$
$$and$$
$$U_b = < w'_1, w'_2, \ldots, w'_j, \ldots, w'_N > . \tag{6}$$

The Pearson similarity is computed as:

$$\begin{aligned} SimWeightCluster_{a,b} &= sim_{pearson}(u_a, u_b) \\ &= \frac{\sum_j (w_j - \bar{U_a})(w'_j - \bar{U_b})}{\sqrt{\sum_j (w_j - \bar{U_a})^2 (w'_j - \bar{U_b})^2}}, \end{aligned} \tag{7}$$

4.2.3 Significance Score of Category Propagation

In general, if we find that a user frequently visits different facilities under a category such as coffee shops and fast food restaurants and seldom visits the facilities like Karaoke houses and Bars, we can consider that the person likes eating rather than entertainment. Considering this fact we have calculated significance score of each category as follows:

Let us consider the information of Figure 5. From Figure 5, we can see that there are many clusters i.e. $C_m \cdots C_n$ belong to $Category\ X$, while there are few clusters $C_p \cdots C_q$ belong to $Category\ Y$. We can say if a

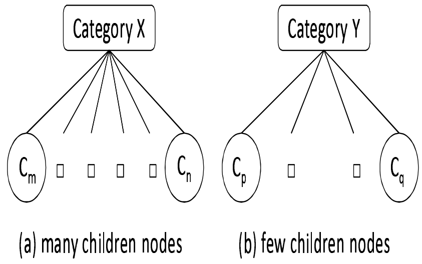

Figure 5: The significance score of different categories

Table 4: Similarity matrix

SimScore	u_a	u_b	$\ldots$	u_k	$\ldots$	u_u
u_a	1	$SimScore_{a,b}$	$\ldots$	$SimScore_{a,k}$	$\ldots$	$SimScore_{a,u}$
u_b	$SimScore_{b,a}$	1	$\ldots$	$SimScore_{b,k}$	$\ldots$	$SimScore_{b,u}$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
u_j	$SimScore_{j,a}$	$SimScore_{j,b}$	$\ldots$	$SimScore_{j,k}$	$\ldots$	$SimScore_{j,u}$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
u_u	$SimScore_{u,a}$	$SimScore_{u,b}$	$\ldots$	$SimScore_{u,k}$	$\ldots$	1

user checked at many clusters of $Category\ X$ and few clusters of $Category\ Y$, then the significance score of $Category\ X$ is bigger than the significance score of $Category\ Y$.

To formalize this, we let $SS_a(x)$ be the significance score of category node x of a user a. It is calculated as follows:

$$SS_a(x) = \frac{Category_a(x)}{totalNumberofCheckinSpots_a}, \tag{8}$$

In (8), $Category_a(x)$ is the number of check-in spots belong to $Category\ X$, $totalNumberofCheckinSpots_a$ is the total number of check-in spots of user a.

4.2.4 Similarity Weight Measure of Upper Layers

Assume that a user u_a can be characterized as a vector $U_a \in R^{|i|}$, where $U_a^i = SS_a(i)$ is user u_a's category score of category i. Let k be the total number of category on layer h of the shared SHCF. Two vectors U_a and U_b can be respectively constructed for user u_a and u_b as follows:

$$U_a = < SS_a(1), SS_a(2), \ldots, SS_a(i), \ldots, SS_a(k) >, \tag{9}$$

$$and$$

$$U_b = < SS_b(1), SS_b(2), \ldots, SS_b(i), \ldots, SS_b(k) > . \tag{10}$$

The similarity weight of upper layers is computed as:

$$\begin{aligned} SimWeightCategory_{a,b} &= \sum_h \frac{1}{h^2} sim_{pearson}(u_a, u_b) \\ &= \sum_h \frac{\sum_i (SS_a(i) - \bar{U_a})(SS_b(i) - \bar{U_b})}{h^2 \sqrt{\sum_i (SS_a(i) - \bar{U_a})^2 (SS_b(i) - \bar{U_b})^2}}. \end{aligned} \tag{11}$$

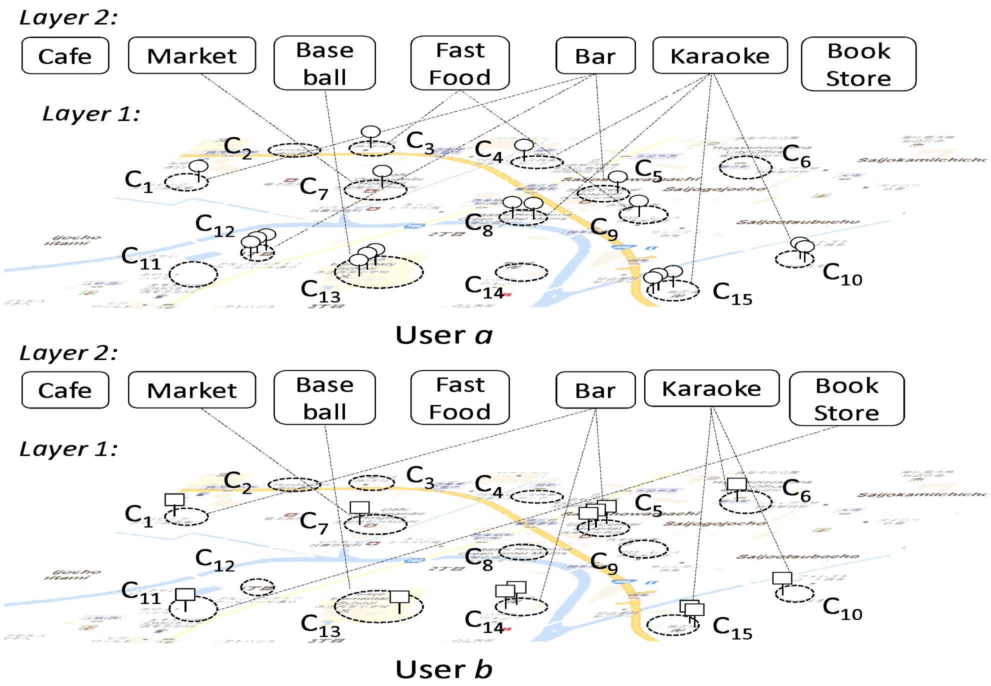

Figure 6: Personal interests distribution hierarchical graph of user *a* and *b*

4.2.5 Similarity Score Measure

$SimScore_{a,b}$ denotes the similarity score between user a and b on all of the shared SHCF layers, it is calculated as follows:

$$SimScore_{a,b} = \alpha \times SimWeightCategory_{a,b} + (1-\alpha) \times SimWeightCluster_{a,b}. \tag{12}$$

In (12), α is an influence parameter. In general, the system accuracy will decrease with an increase in the value of α and vice versa. This is because the users who share the check-in spots in same clusters on the bottom layer are more correlated than those who share interests in same categories on a higher layer.

4.3 Recommendation Generation

After we estimate the similarity between users, a user-based CF method is employed to infer users' interests in unvisited places. We use the profiles of a cluster by exploring the knowledge from POIs. We can recommend different types of places by matching the user's preferences. In addition, by using the profiles of users and the semantics of places, we combine the content-based method with collaborative filtering.

For providing efficient recommendation to the users, we perform following tasks.

4.3.1 Similarity Matrix Generation

After measuring the similarity score between two users, we then formulate a similarity matrix (SM) for all users. The (j,k)-th element of SM, $SimScore_{j,k}$, contains the similarity score of two different users j and k $(1 \leq j \leq |U|, 1 \leq k \leq |U|, j \neq k)$ where $|U|$ is the total number of all users.

From Table 4, given a user (u_j) as a query, we can rank other people in this community $(u_k \in U, j \neq k)$ according to their similarity score $(SimScore_{j,k})$ to u_j. Then, a group of people (U') with relatively high similarity scores can be retrieved as "similar users" for u_j.

Table 5: The Schema of Dataset

parameter	type
userID	varchar(10)
blogID	varchar(15)
latitude	float
longitude	float

Table 6: Examples of check-in spots dataset

userID	blogID	latitude	longitude
1879003160	20111030823..	23.116938	113.315515
1596423831	20111041532..	31.09352647...	114.1893663...
1886906022	20111041635..	30.59464	114.268715
1742520175	20111041737..	30.553574	114.336205
1886906022	20111042045..	40.06715	116.589264
1758921435	20111021821..	30.545422	114.3128

4.3.2 Similar Users and Unvisited Places Discovering

When a query u_j is given, we retrieve vector v_j (here, $v_j = SimScore_{j,k}, k \in |U|, j \neq k$), from the similarity matrix that contains similarity scores between u_j and others. Then, $SimScore_{j,k}$ will be normalized by following equation:

$$SimScore_{j,k} = \frac{SimScore_{j,k} - min(v_j)}{Max(v_j) - min(v_j)}. \tag{13}$$

Thus, the top N users with high $SimScore_{j,k}$ are retrieved as u_j's "similar users" (U'_j).

After we retrieve u_j's similar users U'_j, we discover some places that might interest u_j but have not been visited by u_j. And we define this set of places discovered from U'_j as candidate places C'_j of u_j.

4.3.3 CF-based Recommender

We can say that a user likes a category very much if the user has many check-in in locations of the category. On the other hand, if the user does not check-in in a category, we can consider the user is not interested in the category. Based on this observation, we add the significance score for predicting the rating of each candidate place.

For instance, in Figure 6, we assume user u_b is one of the similar users of u_a. We can imagine that, u_a may be interested in regions (places) C_6, C_{11}, C_{14} which are predictd from u_b's data. However, u_a has never been there. Because we have already know the semantic information of these places, C_6 is a Karaoke house and C_{14} is a Bar. We may think that user u_a likes these kinds of places. On the other hand, C_{11} is a bookstore and u_a has never checked in this category. Although, it is reasonable to recommend C_{11} to u_a, he/she may not like to go there. Our system can efficiently deal with such issues and can provide better recommendation.

After candidate places C'_j are discovered for a particular user u_j, CF-based methods are employed to infer the recommend places to the user.

In SHCF, the following equation describes the process for predicting u_j 's rating r_j^q on a region q ($\in C'_j$):

$$r_j^q = SS_j^q + \hat{r}_j^q, \tag{14}$$

where SS_j^q is the significance score of category propagation of region q, as formula (8).

Table 7: Examples of clusters' information

Cluster	Spots	LocationName	Category	FatherCategory
1	6	Return97 Bar	Nightclub	Entertainment
2	15	KFC	Fast Food	Food
3	22	Starbucks	Cafe	Food
4	3	Wangfujing Bookstore	Bookstore	Education
5	13	Mcdonald's	Fast Food	Education
6	1168	Disneyland	Park	Entertainment
...	...	...	...	...

Table 8: Possible recommendation results of places to a user

	Predicted Visited	**Predicted Unvisited**
Actual Visited	True-Positive (tp)	False-Negative (fn)
Actual Unvisited	False-Positive (fp)	True-Negative (tn)

$\hat{r}_j^q$ comes from the user-based prediction, it likes:

$$\hat{r}_j^q = \bar{r}_j + \frac{\sum_{k \in U'} SimScore_{j,k}(r_k^q - \bar{r}_k)}{\sum_{k \in U'} |SimScore_{j,k}|}, \tag{15}$$

$$\bar{r}_j = \frac{1}{|C|} \sum_{c \in C} r_j^c, r_j^c \neq 0. \tag{16}$$

Equation (15) shows the similarity score $SimScore_{j,k}$ between users u_j and u_k. Here, we use SHCF and normalized formula of (13). According to (15), if two users u_j and u_k are more similar, more weight r_k^q will carry in the prediction of $\hat{r}_j^q$.

The equations (15) and (16) shown above have been used widely in many recommender systems.

5 Experiment

We use the real-world check-in spots dataset to make an offline experiment to evaluate the quality of the recommendations of our framework.

5.1 Data Sample

We used SINA microblog [23] as data source to collect users check-in spots. SINA microblog is a Chinese microblogging website. It is one of the most popular LOSNS in China, which is used by over 30% of Internet users, and has more than 300 million registered users as of February 2012. Additionally, users are also allowed to insert emoticons or attach own image, music, video files, and check-in location information in every post.

We extracted the user's unique identification as attribute "userID", blogs' identification as "blogID", and geographical position as "<latitude, longitude>" to constitute our check-in spots dataset. The schema of our check-in spots dataset is shown in Table 5. Examples of the check-in spots dataset are shown in Table 6.

When we perform the check-in function, we can add some short sentences to express our feelings. The blogID of the check-in spots identifies the documents that was posted with the check-in. Though we do not use the blog information, we add it for future use.

5.2 Experimental Setup

In our experiment, we used 30,515 check-in spots of 10,827 users in a city from SINA microblog. Among them, we select 268 users who checked in more than 25 times. The number of check-in records of those 268 users is 15,713.

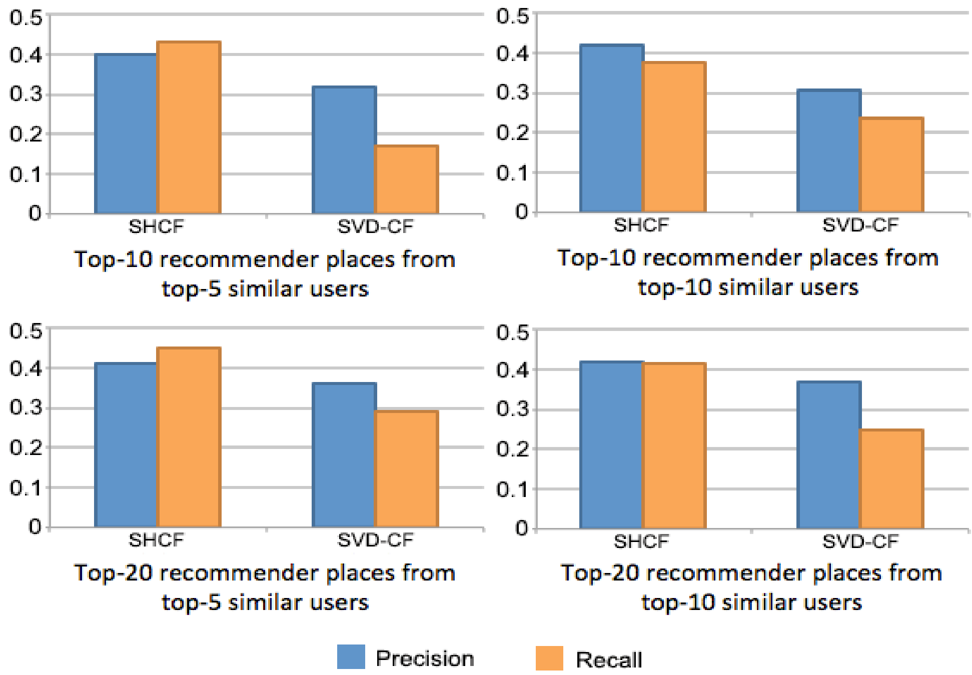

Figure 7: Experimental results with various recommender places from different top-N similar users

In our clustering analysis, we set the neighborhood radius ε equals actual distance 50 meters, and each cluster contains at least a minimum number, $MinPts$ equals 1. Using these parameters, DBSCAN algorithm, made 10,476 clusters on the bottom layer.

After the clustering, we use Foursquare [24] (one of most spotlighted location based social network services) database as our POIs data to annotate clusters. Examples of clusters' annotations, which are semantic information of the clusters, are shown in Table 7.

5.3 Evaluation

To evaluate prediction accuracy of the proposed method, we examined the precision and recall of recommended places and compare the accuracy with a naive Singular Value Decomposition Collaborative Filtering (SVD-CF) recommender, which is a popular method in recommender system.

In evaluation of prediction, firstly, we select 10 testing users, hide some of their already visited check-in places from the dataset that consists all places each user has visited. Then, we use our method (SHCF) and SVD-CF to recommend top-10 and top-20 places to each user based on top-5 and top-10 similar users. Here, we have four possible outcomes for the recommended and hidden places, as shown in Table 8. The visited value is considered the positive target value - the value we are interested in predicting. The non-visited value is considered the negative target value. We, then, compute precision and recall of SHCF and SVD-CF for each of these 10 users for each of the four cases using equations (17) and (18), respectively. Finally, we compute the average precision and recall score of the users for each case and obtain the results as shown in Figure 7.

$$Precision = \frac{tp}{tp + fp} \tag{17}$$

$$Recall = \frac{tp}{tp + fn} \tag{18}$$

From the results, we can see that our SHCF works effectively in recommendation, and it performs better than SVD-CF in each of the four cases.

Figure 8: Check-in location for users

Table 9: Detail check-in information

Places	User *a*	User *b*	User *c*	User *d*
Zhonghua Road Apartment	3	3	0	0
Wangjiang Yangxinyuan Music Town	1	0	0	0
Changhong Touying Singing Hall	2	0	2	0
Better Life Singing Hall	1	0	2	0
Wuchang Wenhua Middle School	2	0	0	0
Wuchang Library	4	0	0	2
Shouyi Park	1	1	0	1
Yellow Crane Tower	1	1	2	1
Xinshidai Concert Hall	0	0	0	1
Huanghe Drama Bldg	0	0	1	0
Dawu Changcheng	0	0	0	1
Hubei Theater	0	0	1	0
Wuhan Middle School	0	3	0	0
Hubei Library	0	0	2	0
McDonalds	0	1	0	0
Xinweiduo Skating City	0	0	0	1

5.4 Recommendation Example

In order to demonstrate our method, we picked up four real users, say *user a*, *user b*, *user c*, and *user d*, in the SINA micro blog and show our recommendation results. The map in Figure 8 shows the check-in spots of four users at Yuemachang area, Wuhan City, China. Table 9 shows detailed check-in information of the users. Using our approach, we found that *Huanghe Drama Bldg* has the highest predict rating and *Hubei Theater* has second highest predict rating. So, our system provides *Huanghe Drama Bldg* and *Hubei Theater* as top-2 recommendation to *user a*. From the information of Table 9, we can see that *user a* is highly interested in entertainment facilities. So, we can say that the recommendations provided by our system are appropriate.

6 Conclusion

In this paper, we have proposed a place recommender system that takes into account the semantics of users' check-in activities. In the system, to reflect users' semantics, we utilized semantic hierarchical category-graph framework that contains multiple layers which include physical location layer and semantic (category) layer. We calculated similarity of users in the semantic hierarchical category-graph framework and predicted each user's interest accurately.

So far, our work in this paper does not consider user's current location and the physical distances among the recommended places. In future, we will consider such a real time recommendation problem in which we have to get more accurate and meaningful results that is related to user's current location. Moreover, in this paper, we used the nearest POI from the center of the cluster. In future, we hope to consider the semantics based on the nearest *k* POIs by stochastic means to improve the recommendation accuracy.

Acknowledgment

This work is supported by KAKENHI (23500180) Japan. Mohammad Shamsul Arefin is supported by the scholarship of MEXT Japan.

References

[1] G. Adomavicius, and A. Tuzhilin. Toward the next generation of recommender systems: Asurvey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, pages 734-749, 2005.

[2] A. Gunawardana, and G. Shani. A survey of accuracy evaluation metrics of recommendation tasks. *The Journal of Machine Learning Research*, pages 2935-2962, Volume 10, 2009

[3] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, pages 5-53, Volume 22 Issue 1, January 2004.

[4] B.Berjani, and T. Strufe. A recommendation system for spots in location-based online social networks. *In Proc. of the 4th Workshop on Social Network Systems*, pages 1-6, 2011.

[5] C. Chow, J. Bao, and M. F. Mokbel. Towards location-based social networking services. *In Proc. of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*, pages 31-38, 2010.

[6] Y. Takeuchi, and M. Sugimoto. CityVoyager: An outdoor recommendation system based on user location history. *In Proc. of the 3rd international conference on Ubiquitous Intelligence and Computing*, pages 625-636, 2006.

[7] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with GPS history data. *In Proc. of the 19th international conference on World wide web*, pages 1029-1038, 2010.

[8] Y. Zheng and X. Xie. Learning travel recommendation from user-generated GPS trajectories. *In ACM Transaction on Intelligent Systems and Technologies*, pages 1-2, 2011.

[9] Y. Zheng, X. Xie, and W. Ma. GeoLife: A collaborative social networking service among user location and trajectory. *IEEE Database Engineering Bulletin*, vol. 33, pages 32-40, 2010.

[10] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W. Ma. Recommending friends and locations based on individual location history. *ACM Transaction on the Web*, pages 1-44, 2011.

[11] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, vol. 41, pages 391-407, 1990.

[12] D. W. McDonald. Recommending collaboration with social networks: a comparative evaluation. *In Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pages 593-600, 2003.

[13] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattcharjee. Measurement and analysis of online social networks. *In Proc. of the 7th ACM SIGCOMM Conference on Internet Measurement*, pages 29-42, 2007.

[14] M. Lee, and C. Chung. A user similarity calculation based on the location for social network services. *In Proc. of the 16th international conference on Database systems for advanced applications*, pages 38-52, 2011.

[15] N. Li and G. Chen. Sharing location in online social networks. *IEEE Network*, pages 20-25, 2010.

[16] D. Ashbrook and T. Starner, Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, pages 275-286, 2003.

[17] T. Horozov, N. Narasimhan, and V. Vasudevan, Using location for personalized POI recommendations in mobile environments. *Symposium on Applications and the Internet*, pages 124-129, 2006.

[18] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. *In Proc. of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 43-52, 1998.

[19] H. Polat, and W. Du. SVD-based collaborative filtering with privacy. *In Proc. of the ACM symposium on Applied computing*, pages 791-795, 2005.

[20] M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *In Proc. of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226-231, 1996.

[21] C. Kenneth, and W. Gale. Inverse document frequency (IDF): a measure of deviations from Poisson. *In Proc. of 3rd Workshop on Very Large Corpora*, pages 121-130, 1995.

[22] J. L. Rodgers and W. A. Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, vol. 42, no. 1, pages. 59-66, 1988.

[23] SINA Micro Bolg. *http://www.weibo.com/*

[24] Foursquare API. *https://developer.foursquare.com/*