

# A Social Search Scheme Considering User Preferences and Popularities in Mobile Environments

**Kyoungsoo Bok<sup>1</sup>, Jongtae Lim<sup>1</sup>, Minje Ahn<sup>2</sup> and Jaesoo Yoo<sup>1</sup>**

<sup>1</sup>School of Information and Communication Engineering, Chungbuk National University  
1 Chungdai-ro, Seowon-gu, Cheongju 362-763, Korea

[e-mail: ksbok@chungbuk.ac.kr, efzotz@gmail.com, yjs@chungbuk.ac.kr]

<sup>2</sup>S/W Platform Team, Samsung Electronics Co., Ltd.

129, Samsung-ro, Yeongtong-gu, Suwon-si 443-742, Korea

[e-mail: minje.ahn@samsung.com]

\*Corresponding author: Jaesoo Yoo

*Received March 10, 2015; revised October 10, 2015; revised December 15, 2015; accepted January 10, 2016;  
published February 29, 2016*

---

## Abstract

As various pieces of information can be provided through the web, schemes that provide search results optimized for individual users are required in consideration of user preference. Since the existing social search schemes use users' profiles, the accuracy of the search deteriorates. They also decrease the reliability of a search result because they do not consider a search time. Therefore, a new social search scheme that considers temporal information as well as popularities and user preferences is required. In this paper, we propose a new mobile social search scheme considering popularities and user preferences based on temporal information. Popularity is calculated by collecting the visiting records of users, while user preference is generated by the actual visiting information among the search results. In order to extract meaningful information from the search target objects that have multiple attributes, a skyline processing method is used, and rank is given to the search results by combining the user preference and the popularity with the skyline processing result. To show the superiority of the proposed scheme, we conduct performance evaluations of the existing scheme and the proposed scheme.

---

**Keywords:** Social networks, user preference, popularity, social search, ranking

---

This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2015-H8501-15-1013) supervised by the IITP(Institute for Information & communication Technology Promotion), by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2015R1D1A3A01015962), and by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No.2013R1A2A2A01015710)

## 1. Introduction

**D**ue to advances in mobile networks and smartphone popularity in recent years, users can now connect to the internet anytime from anywhere. In addition, as users' locations can be found through a Global Positioning System (GPS), wireless internet, and mobile communication networks, several studies on mobile social network services (SNSs) combining location-based services have been conducted [1, 2, 3, 4, 5, 22, 23]. The importance of social searches that consider user preferences has arisen along with the development of the mobile network environments [6, 14, 15, 24]. The social search is a method that provides users preferred search results by analyzing their individual preferences from various SNSs that they visit frequently and considering them in contrast with existing web searches [6, 7, 19]. In the mobile environment, the ability of the social search to determine how to maintain up-to-date data status and how to assign the rank of the results is a critical issue. In contrast with general computing environments, portable terminals are highly personalized, and the amount of information that can be contained on a screen is limited. Therefore, an efficient search result layout is more convenient for users [8, 9, 10, 13, 16].

The most general social search method is to collect user behavior information from SNSs or emails and to search various sets of information provided through SNSs or common interests. That is, user preferences are analyzed by collecting keywords and internet links from main bodies of emails or comments over SNSs, and they are reflected in search results to assign ranks. Most social searches use implicit information-collection methods in order to collect information for analyzing user preferences without the direct participation of users. In other words, users do not enter their preferences or profiles themselves. Rather, their preferences or interests are extracted through the collection of their recent SNS activity information. However, during implicit information-collection, information should be collected for a certain period of time to analyze user preferences [11, 12]. If sufficient information is not collected, it could degrade search accuracy, since user preferences may not be understood correctly.

There are two methods of reducing the collection period of user activity information for the analysis of recent user preferences. The first is a scheme based on public popularity or expert assessment [1]. Such a scheme calculates popularity from the activities of SNS users and analyzes core keywords of query content transferred by users. Through the analyzed content, experts are selected for a corresponding keyword, thereby reflecting their assessment information and assigning priority to the search results. Therefore, objective search results can be provided through public popularity and expertise assessment. However, this scheme has a drawback, in that it cannot reflect individual personal preferences directly. The second method is a scheme that uses profiles of similar users [10]. This scheme performs searches using the preferences of users that have similar profiles to a user while collecting profiles. In general, user preferences change from time to time, but few users change their preferences explicitly. Therefore, if the preferences of users whose profiles are similar to that of a particular user are used for a search, it may generate search results that are different from the particular user's preference. In addition, since this method searches similar users using a profile that a user explicitly enters, the reliability of the profiles may be degraded. Since both of the two methods above determine the priority of search results based on user preferences, public popularity, or expert assessments, they are limited in their ability to provide search results suitable for users who move based on the search results. Therefore, a scheme that can provide user-preferred

search results considering temporal characteristics and the preference and popularity of a specific place is required.

In this paper, we propose a mobile social search scheme that considers popularity and user preference based on temporal information. Since the proposed method selects candidates appropriate for a time slot to be searched utilizing temporal information, it can reduce the amount of computation, as there is no need for an exhaustive assessment of all places within a search radius. Popularity is calculated by collecting the actual visit records of service users. The existing implicit information-collection methods determine user preferences by analyzing the activities of users over SNSs, such as registration of posts, comments, or sent and received emails. However, such methods require long-term data collection to create reliable user preferences. To overcome this limitation, this paper creates user preferences by collecting information on places a user visits through social search results. This process can shorten the data-collection period required for the analysis of user preferences. A skyline query processing technique is used to extract meaningful information from search target objects that have multiple attributes. The proposed scheme assigns priority of the search results based on the skyline query processing results as well as user preferences and popularity.

This paper is organized as follows. In Section 2, the characteristics and issues of the existing schemes in related work are described, while the proposed social search scheme is described in Section 3. In Section 4, the results of the performance evaluation are presented to verify the performance of the proposed scheme. Finally, in Section 5, conclusions and suggestions for future research are presented.

## 2. Related Work

T. Vu proposed Odin that finds targeted answers about who most likely helps answer the question with a high level of confidence in mobile social networks [18]. Odin uses the latent variable model proposed in [21] to generate the relationship strength between all users from social network profiles and identifies user expertise via mining social network data as well as sensing data from mobile devices. To enhance the probability of getting an answer with high quality, a modified page rank-like scheme considering the relationship strength and the user expertise is used.

P. Shankar proposed a location based service called SocialTelescope based on user interactions about locations to accomplish a location query in mobile social networks [1]. Since user interactions act as implicit feedback about locations in SocialTelescope, they are used to maintain and index information about locations. To process location queries from users, SocialTelescope first generates candidate locations based on matching user tags and then sorts these results by popularity. SocialTelescope assigns expertise score to users based on the query keywords. The ranking engine uses the location index and the user expertise score to rank the candidate locations by popularity weighted by user expertise.

A. Nagpal proposed SLANT using online social chatter to provide personalized web search [11]. SLANT uses the Google custom search engine to search sites with links extracted from email and Twitter. SLANT mined a user's email and Twitter feeds to improve the quality of web search and developed four indices such as email links index, friends' names index, Twitter links index, and toptweets index. SLANT combines results from the different indices for improving user satisfaction.

A. Kashyap developed SonetRank to provide the personalized web search based on the aggregate relevance feedback of the users in similar groups [17]. SonetRank combines the

three factors such as user's preference, group preference, and query preference and builds a Social-Aware Search (SAS) graph model that captures users, queries, groups, documents, and their associations. A personalized ranking is generated based on the SAS graph via the authority-flow based algorithm SonetRank calculates a confidence factor to decide the quality of the SonetRank ranking and merges the results returned by the SonetRank and the results returned by the search engine via confidence factor.

Y. A. Kim proposed the Topic-Driven SocialRank scheme based on social information such as the user profile and connectivity that are varied by topics [16]. Topic-Driven SocialRank scheme provides interest-driven search results with relevant web content from friends using social contacts online by identifying similar, credible users. The users having high social relationship values issue more relevant search results than other users.

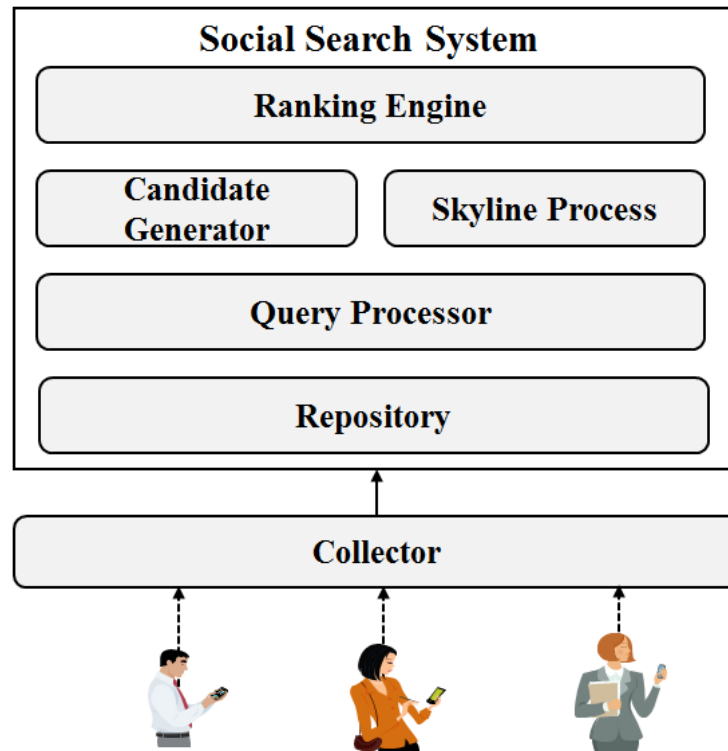
A. Jatowt proposed the raking scheme based on temporal analysis to retrieve fresh and relevant contents [20]. A search engine first generates candidate results via web search engine. Since content changes are more important to users than other types of changes, content changes are considered. To determine the relevance of the changes to the query topic, cosine similarities between the query vector and the change vectors for each candidate result are calculated. The candidate results returned by a search engine are re-ordered to give a high priority to the web pages with fresher and relevant information higher.

### 3. The proposed social search scheme

#### 3.1 System architecture

In this paper, we propose a novel social search scheme to overcome the problems of the existing schemes and to process a query efficiently for social searches in a mobile environment. The proposed location-based social search scheme is composed of four steps. First, we create a candidate group that is provided as a final search result by excluding non-operating places that have no visitors during the query time slot among place candidates including keywords by analyzing the keyword and time information included in the query. Second, we calculate the popularity of each candidate through the user's visit records previously collected from SNSs. Third, we select meaningful places by performing skyline processing with respect to the candidate group and assign weights to attributes based on user preference information to calculate the user preference scores of the candidates. Finally, we sum the popularity score and the user preference score and assign priority of the search results based on the final score, thereby providing the search result to users.

**Fig. 1** shows the overall system structure of the proposed social search scheme. A server continuously collects and analyzes SNS posts, including various pieces of location information uploaded by all users, using a collector. A repository stores user information, location information, and visit records. A user requests a query to a social search system in a mobile environment. The proposed social search system consists of five modules such as repository, query processor, candidate generator, skyline process, and ranking engine. The repository stores SNS information and location information collected from mobile users, and user preferences generated by our system. A query processor extracts core keywords, the time, and the user's current location from the user's query. A candidate generator creates appropriate candidates and calculates a popularity score, while a skyline process assigns a weight based on the user preference information. A ranking engine sums the popularity and user preference scores obtained from the candidate generator and the skyline process and assigns rankings.



**Fig. 1.** System structure of the social search scheme

An information-collection module (collector) gathers user feedback information regarding location information, user visiting information, and search results. **Fig. 2** shows the information-collection process of the proposed scheme. Location information is newly created or updated if the location information is not available in the server due to a lack of user visits or if information update is required. When a visitor performs a location registration for the first time, the server assigns a unique ID for the location and stores the location information along with additional location-related information, such as business trade name, business sector, and price information in the database. If the location information requires an update, then all information other than the unique ID of the location is updated collectively. If the server already contains the visit information of a user, it only records the visiting information of the user. The user's visiting record consists of a unique user ID, the visit location, and time information. The feedback of the search results stores information about the location once a user performs a search of the surrounding area and an actual visit to the place is conducted. Each of the characteristic values regarding the visited place is reflected in the user's preference through the feedback. Based on the accumulated visit information, a user's preference information is updated in the user table.

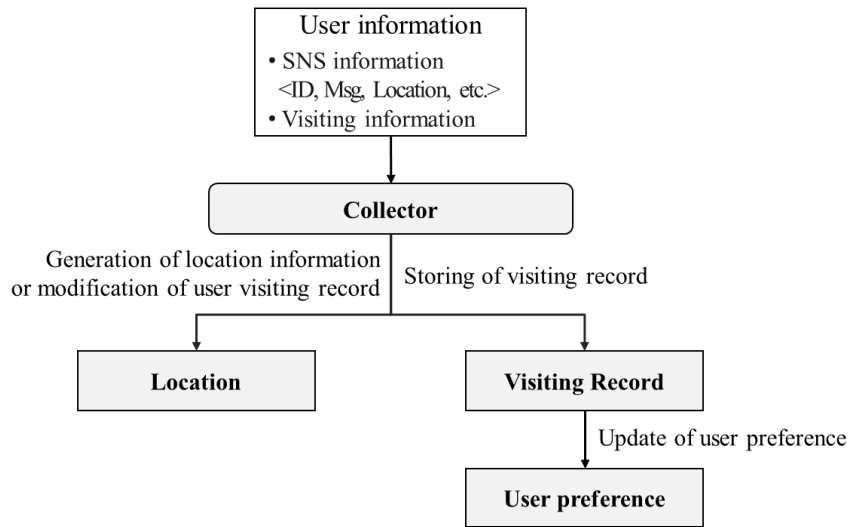


Fig. 2. Information-collection process

The proposed social search scheme continuously collects and stores visiting records of all locations of users through location-based services during the query processing. A user sends a query including his/her location information and expected visiting time to receive social search services, while the server provides a user with appropriate search results. Fig. 3 shows the query processing procedure of our system. The query processor analyzes user queries, while the candidate generator selects appropriate candidate results using user location and core keywords and excludes candidates that are not operating through time information. All location information included in candidate results calculates popularity scores through users’ visiting frequencies. The skyline process assigns weights to locations that satisfy user preferences through the skyline among locations in the candidate group created by the candidate generator. Finally, the ranking engine sums scores calculated in the candidate generator and the skyline process respectively to assign rankings and return the top-k result values requested by a user.

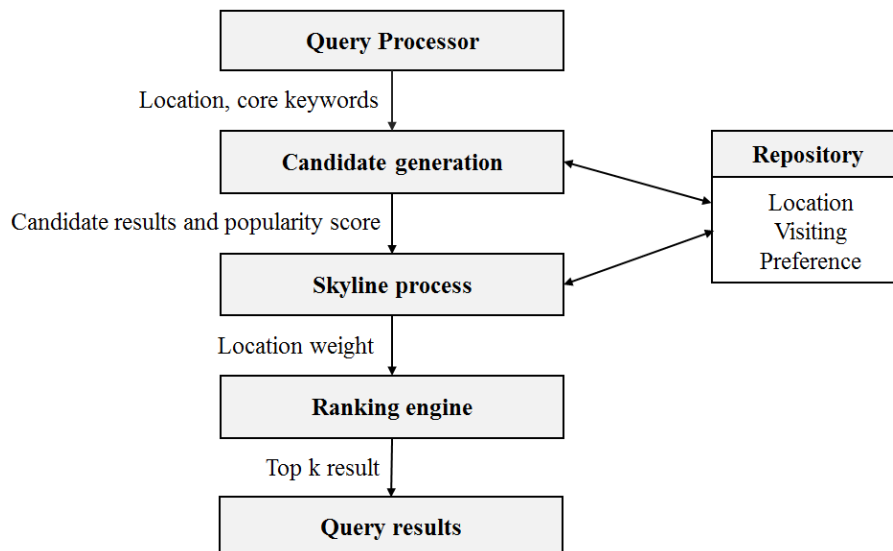


Fig. 3. The query processing procedure

### 3.2 Repository structure

In order to identify user preferences by analyzing information collected from users, a repository is required to store detailed information about visiting records and locations. The preference table stores user preference information, while the visiting table stores user's visiting records. If a user visits a certain location via the previous search result of our system, a visiting record is stored. The location table contains detailed information about locations.

**Table 1** shows a repository structure. Data collected from users is stored in the three tables to calculate a fast search in response to various queries by users. The visiting record table stores the visiting records of all users. Visiting times and locations are recorded in the time and Location\_ID fields in the visiting record table. If a certain location actually is visited via previous search results, the visiting information is stored in the feedback field. The preference table stores user preference information. In preference table,  $w_i$  is the weight value, which is calculated by visiting record and the existing weight value. The location table consists of the unique ID of the location, name, location coordinates, category, and so on.

**Table 1.** Repository structure

(a) Preference table				
User_ID	W1	W2		
1402954	0.38	0.62		
1302540	0.71	0.29		
1102450	0.91	0.09		

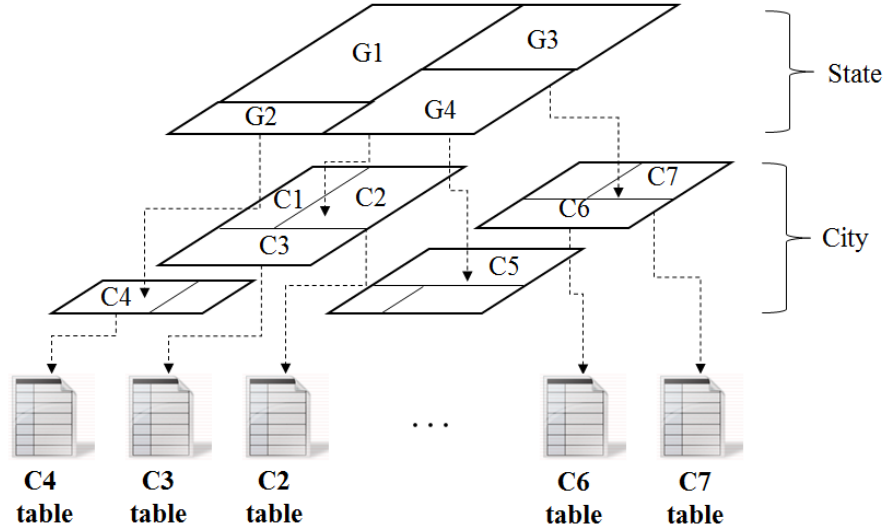
  

(b) Visiting record table			
User_ID	Time	Location_ID	Feedback
1402954	20121009T131523	274	0.43, 0.57
1302540	20121008T153010	455	-
1102450	20120931T050312	423	-

(c) Location table				
Location_ID	Name	X	Y	Category
274	Dunkin	30	-97	Restaurant, seafood
455	BergerKing	27	-95	Restaurant, bakery
423	Napoli	-50	42	Restaurant, seafood

The location information is very large, so it is stored through a hierarchical structure up to the detailed regional unit based on the actual address system, as shown in **Fig. 4**. For example, if a search location is within a city called C7 in a state called G3, only locations in a list within the C7 table among city lists inside a unit region of G3 are searched. Generally, a user tends to mainly visit locations in his/her administrative region (state, city, etc.). That is, a user is likely to visit facilities in a city where she/he is currently located rather than using facilities of other cities that are nearby. Therefore, if a hierarchical structure is employed to manage usable facilities, candidates can be searched efficiently based on location. In addition, search criteria to generate an appropriate candidate group can be minimized accordingly. Distance information out of the additional pieces of information regarding locations is not stored as well, because it changes with the user's search location.



**Fig. 4.** Hierarchical structure of location information

General information-collection methods require long-term information collection of activities of users over SNSs to create reliable user preferences. Data on whether a user actually visits a location from existing search results is collected to create user preferences to shorten the implicit information-collection period. The final weight  $w_i$  of each parameter in the User table represents a user preference of each attribute. A user preference is calculated by collecting information about whether a user visited a location from existing search results and storing an average value of the visits, as shown in Equation (1). Here,  $k$  is the total number of user feedback entries in a recent period  $T$ , while  $d_i$  is an  $i$ -th attribute value, which is calculated as shown in Equation (2). In Equation (2),  $n_i$  is a normalized value of  $v_i$  collected through the user feedback, which can be calculated by  $\alpha_i v_i$ . Since the range of values in each attribute  $i$  of  $\alpha_i$  differs, it is a parameter to normalize them between 0 and 1.  $v_i$  is an attribute value of location based on a user, such as distance and price when a user visits a specific location from the search results.

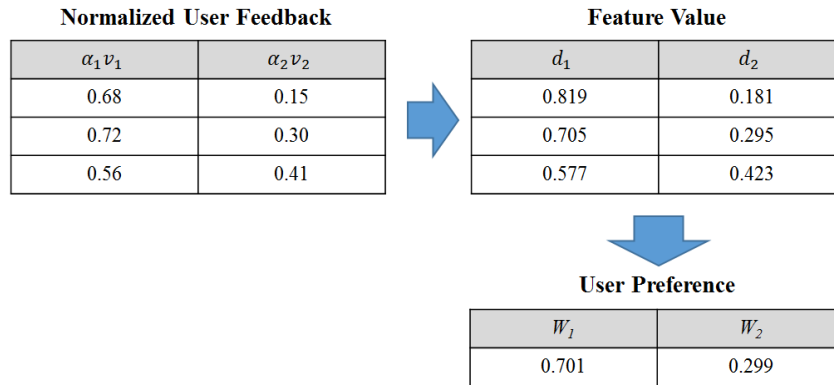
$$w_i = \frac{1}{k} \sum_{i=1}^k d_i \tag{1}$$

$$d_i = 1 - \frac{n_i}{\sum_{j=1}^l n_j} \tag{2}$$

For example, let us assume that three pieces of feedback information are collected from a user, as shown in Fig. 5. A normalized user feedback is a normalized value of user feedback information. Through the normalization, attribute units that differ from one another can be unified. When this is calculated via Equation (2), a feature value of each attribute is created.



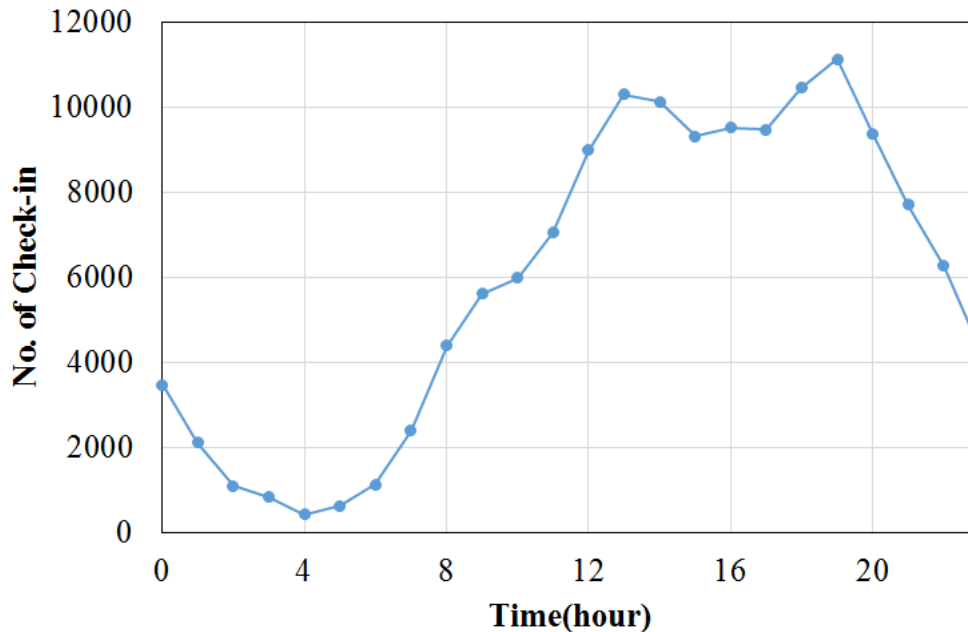
Through this, a ratio of each attribute can be calculated. The calculated value represents how important each value is to a user so that a mean value of feature values is calculated as a user preference using Equation (1).



**Fig. 5.** User preference information

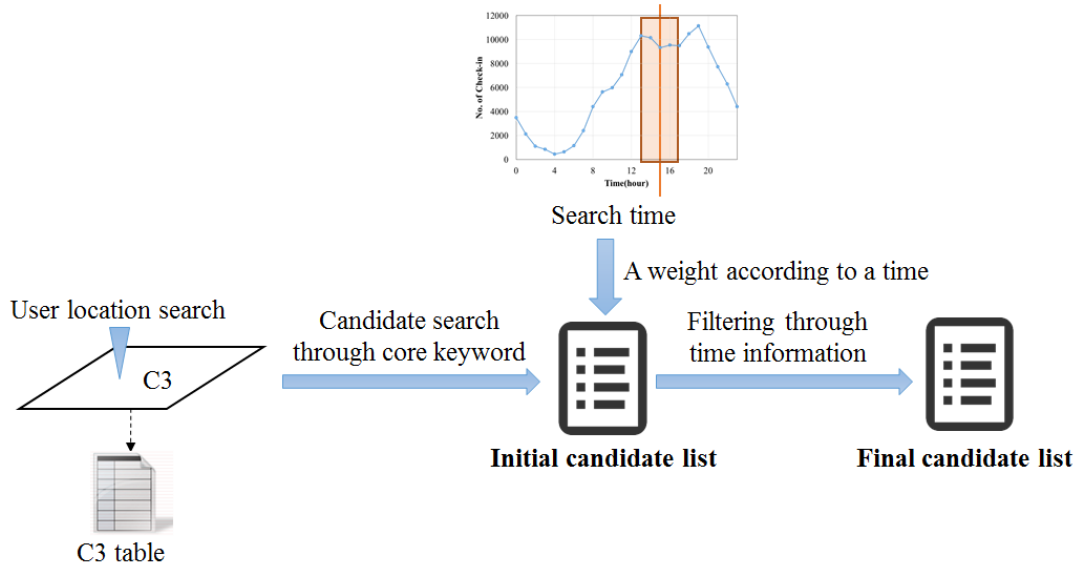
### 3.3 Candidate result generation

In the proposed scheme, a visit time is included and transferred with the search information. Therefore, the search efficiency can be increased by applying a method of assigning a high weight to publicly popular locations in the search time slot utilizing a mixed model. The utilization of time information can produce three positive effects. First, the hours of operation of the search location can be utilized in the search process. These can be inferred through the visit records of a corresponding time slot by checking user visit records within a valid time range in the search time. Therefore, locations that a user cannot visit within the expected visit time slot are excluded from the candidate group based on operating hours. Second, effective processing results in response to comprehensive queries can be provided. As shown in [Fig. 6](#), the most active check-in records occur between 12:00 and 20:00. The expected value of a comprehensive keyword, such as Restaurant, differs slot by slot when the time is divided into 12:00 to 14:00, 14:00 to 18:00, and after 18:00. It may be a simple location search for regular meals or a location search including Cafe or Bar depending on the search time. That is, if a query requires a wide range of search results, the number of locations to be searched can be reduced. Third, unnecessary computations can be decreased. The amount of accumulated check-in information collected continuously within a specific period is enormous. Therefore, it would take a great deal of time to sort out the rankings of candidates based on the assignment of user preference and public popularity scores in each module. Therefore, the utilization of time information only exploits check-in information within the valid range so that it can reduce the number of candidates, thereby increasing the search efficiency and computation speed.



**Fig. 6.** Gowalla user check-in record by time

It is vital to create an appropriate candidate group in order to provide search results required by users. **Fig. 7** shows a candidate-generation process. First, it extracts location information that includes a core keyword in the location category information in the unit-region table that corresponds to a user's current location. Locations that have no visit records within the search time are removed from the temporarily generated candidate group based on the time information specified by a user. This is because the businesses in the removed locations are regarded as not operating in the specified time. Through this process, a final candidate list is generated to provide search results. For example, assuming that a user called User34 arrives at a meeting place one and a half hours earlier than the appointment time, which was 16:00, and searches for a suitable café nearby to wait until the appointment time, he will send a query to the server in the following form: <User34, cafe, 36.002453, 145.356363, 20131102T123124>. The server then identifies his current location through geocoding based on the latitude and longitude sent by the user. Through the location hierarchy structure, the city where the user is currently located is searched, and the locations that have cafés in the sub-category field are searched from the corresponding city table. Once an initial candidate list is generated via the search keyword, the time information of 14:00 is extracted from the search time "20131102T143124." Locations for which no check-in records are found within the valid time period based on 14:00 are removed from the candidate list. In this way, locations that are currently operating can be checked, and nearby cafés that can be visited are included in the final candidate list.



**Fig. 7.** Candidate group generation process

In the proposed scheme, the time information  $v$  used to check the operating hours can be calculated via Equation (3). A weight is assigned within a valid range of  $u_1$  and  $u_2$  with respect to the past visit times  $t(O_i)$  of each location  $O_i$  based on an expected visit time  $t(r)$ . Here,  $u_1$  and  $u_2$  are values specified by repetitive experiences. If there are no visit records within a valid range, locations are removed from the candidate group. They are taken into consideration to calculate a popularity score  $P(O_i)$  through weights by utilizing time information  $v$  only when visit records are found.

$$v = \begin{cases} 0, & u_1 < t(O_i) \\ 0.5, & t(r) < t(O_i) \leq u_1 \\ 1, & t(r) = t(O_i) \\ 0.5, & u_2 \leq t(O_i) < t(r) \\ 0, & t(O_i) < u_2 \end{cases} \quad (3)$$

### 3.4 Ranking

The proposed scheme performs a four-step processing procedure to return appropriate results to users based on data collected continuously. Once an appropriate candidate group is selected based on the user's current location, popularity and location preference scores are calculated via the candidate generator and the skyline process, respectively. The location preference means how much a user prefer a particular location. The ranking engine assigns a final score  $S_i$  through Equation (4) by taking the popularity and location preference. Here,  $L(O_i)$  is the location preference,  $P(O_i)$  is the popularity, and  $\beta$  is the weight for each location in a candidate group  $O_i$ . The weight  $\beta$  is determined by the search frequency. The candidate group is rearranged based on the final score, and the final result is returned to the user.

$$S_i = \beta \frac{L(O_i)}{\text{Max}(L(O_i))} + (1 - \beta) \frac{P(O_i)}{\text{Max}(P(O_i))} \quad (4)$$

The Query Processor extracts core keywords and coordinates information from the user search query. In this way, a candidate group is selected based on the user's current location. The information of the selected candidate locations is calculated as a score in which public popularity is considered during the recent period  $T$  based on a search period  $t(Q)$ . Equation (5) calculates popularity, where  $C$  is the number of total visitors of total candidate locations during the recent period  $T$ ,  $m$  is the number of visitors of the corresponding candidate location  $O_i$ ,  $t_j(O_i)$  is the time that a visitor visits a candidate location  $O_i$ . The popularity is calculated by assigning a higher weight to a more recent visit record  $t_j(O_i)$  of location  $O_i$  from the query time. In addition, a higher weight is given to visit records that have times closer to the search time through the time information weight  $\nu$  according to the search time.

$$P(O_i) = \frac{1}{C} \sum_{j=1}^m \nu \left( \frac{1}{t(Q) - t_j(O_i) + 1} \right) \quad (5)$$

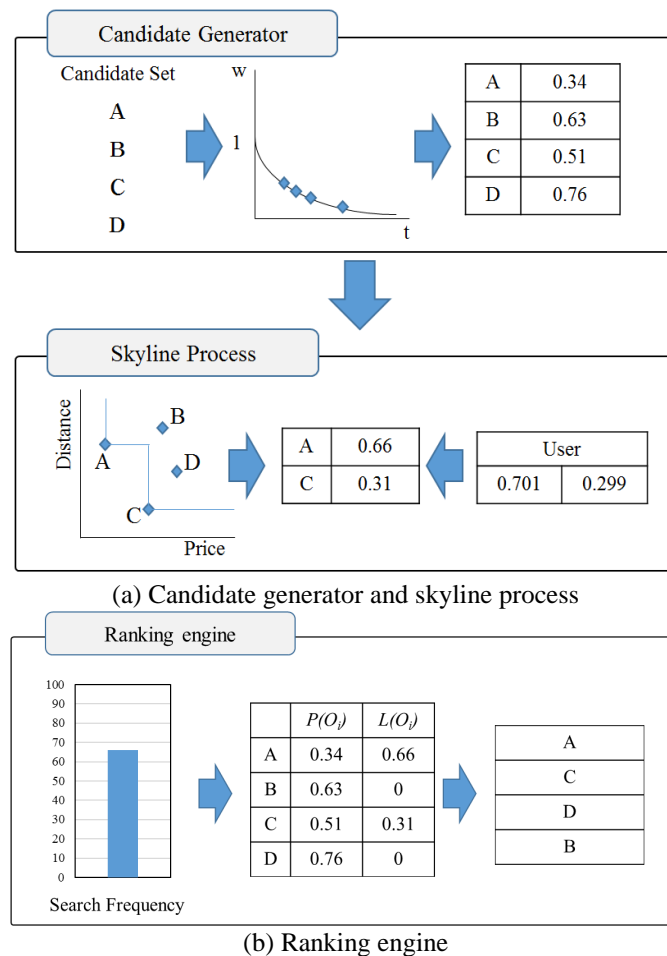
The Skyline Module selects objects that are not dependent on specific attribute values among search target objects that have multiple attributes. Through this process, unnecessary locations can be removed in advance, and priority is given by extracting only meaningful locations to users. Therefore, a location preference is calculated by considering user preference information with respect to objects selected through the skyline. Equation (6) calculates  $L(O_i)$ . Here,  $w_k$  is user preference information, while  $n_k$  is a normalized attribute value of locations collected through feedback.

$$L(O_i) = \sum_{k=1}^n w_k n_k \quad (6)$$

The Ranking Engine sums popularity scores and location preferences, thereby assigning rankings and returning the results. Here, a ratio  $\beta$  of popularity score and location preference is reflected through the actual search frequency  $f_a$  divided by the search frequency threshold  $f_t$ , as shown in Equation (7). The constant  $\delta$  that is applied to the calculation of  $\beta$  is a value calculated via performance evaluation in various environments. If user preference information exceeds the constant value  $\delta$  of the equation, search results in which only user preference information is considered (i.e., not general popularity) are provided. Therefore, to adjust this, the actual search frequency  $f_a$  should not be accumulated such that it exceeds the search frequency threshold  $f_t$ , and the applied proportion should be limited by the constant  $\delta$ .

$$\beta = \frac{f_a}{f_t} \delta \quad (7)$$

**Fig. 8** shows a Top-3 processing procedure in the social search. Generally, search results provide all results related to a keyword. As shown in **Fig. 8(a)**, nearby information is searched based on the location of a user who requests a search from the Candidate Generator. Here, locations that have no visit records within a corresponding time slot are removed from the candidate list based on the time information transferred by a user. Based on the search keyword and time and location information, A, B, C, and D are extracted. Then, the weights and popularity of each location considering the search time are calculated. The more recent the visit, the higher the weight assigned to calculate public popularity. Similarly, the closer the time of the visit to the time requested by a user, the higher the weight assigned. The Skyline Module selects locations A and C that are not dependent on specific attributes among extracted locations and calculates a location preference by considering user preferences. In the Ranking Engine, popularity  $P(O_i)$  and location preference  $L(O_i)$  are summed to calculate a final score, as shown in **Fig. 8(b)**. The proportions of popularity and location preference vary depending on the user's search frequency. If the user's search frequency shows a 65% utilization rate during a unit period, the proportion of location preference exceeds that of public popularity. Based on this result, a final rank is given for each location, thereby providing A, C, and D to the user. If the user requests additional information using a scroll, locations with rankings lower than B are provided.



**Fig. 8.** Social search processing procedure

#### 4. Performance evaluation

To show the superiority of the proposed social search scheme considering user preferences, we conducted performance evaluation in the same environment as the scheme proposed in [1]. The scheme proposed in [1] provides results through objective popularity and expert assessment. Therefore, it provides social search results that do not consider users' personal preferences. Through the comparison evaluation with [1], we verified the objectivity of the ranking. The extent to which user preferences were reflected in the results was also verified through the average values of all attributes of locations included in the ranking. The performance evaluation was implemented by Java and conducted on a system with an Intel core i5-3570K CPU 3.4GHz and 8GB memory. We employed MySQL as a database. **Table 2** shows the data characteristics used in the performance evaluation. The experimental data was check-in data collected via the Gowalla API by Stanford University. This data consists of coordinates, user IDs, location IDs, and time location (i.e., no price information). As a result, price information was given arbitrarily by assigning prices as a mean value without detailed menu differentiation. Registered locations in New York City referred to restaurants where users had at least one check-in record rather than all restaurants in New York City. Gowalla provides location-based social services in smartphones or mobile terminals. It is a mobile application used to share and query information about who visits which locations with how many people along with location information between users. For this experiment,  $\delta$  was set to 0.7 in Equation (7). This was set through iterative experiments in the environment described in **Table 2**.

**Table 2.** Gowalla check-in data

Attribute	Value
No. of total users	196,591
No. of total check-ins	6,442,890
Collection duration	Feb. 2009–Oct. 2010 (20 months)
Registered locations	24,533

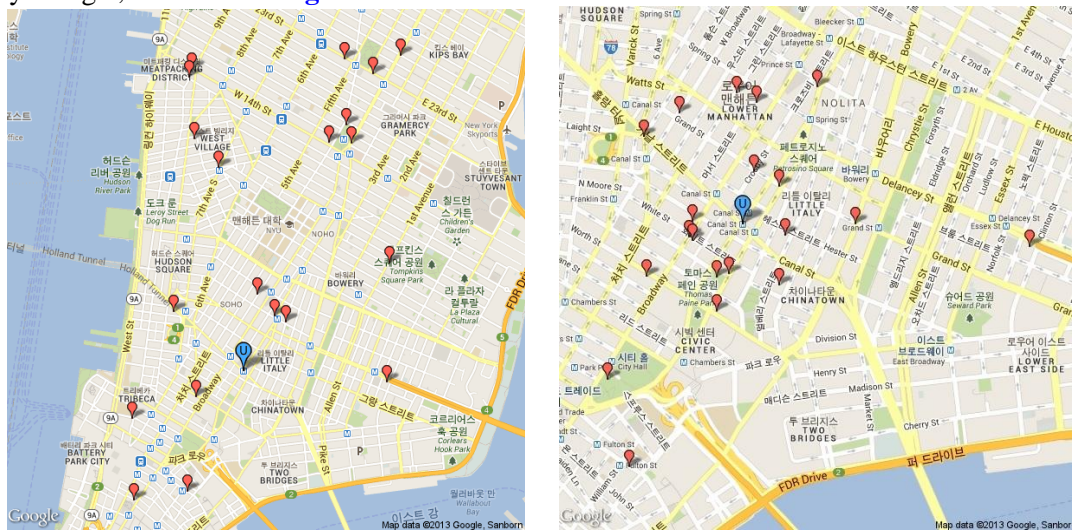
Initial values for the performance evaluation based on preference information were set as shown in Table 3. Since the ability to obtain user location information at the time of the search from the existing collected dataset was limited, user location information was set arbitrarily. Preference is divided into two parameters such as distance and price. A user's preference is computed by Equation (1) by using the visiting record of the user. When a user visits a particular location, the price preference and the distance preference mean weights assigned to a price and a distance, respectively. In other words, the price preference and the distance preference mean that the reason that a user visits a particular location is a price or a distance. For example, when type 1 with the distance preference 0.7 and the price preference 0.3 visits a location, it prefers distance over price as a choice criterion of the location. The search frequency represents the number of search requests about a particular location and is used to compute ranking scores in Equation (4). As shown in Equation (4), the proposed scheme assigns high weights to the location preference when the search frequency is high. It assigns high weights to the popularity when the search frequency is low. The user types are classified into seven types according to the price preference, the distance preference, and the search frequency. In order to conduct the performance evaluation according to the preference and the search frequency, types 1, 2, 3 and 4 set a high price preference value to 0.7 and a low price

preference value to 0.3 based on data collected via the Gowalla API. The distance preference was applied in the same way as the price preference. We also set the high search frequency value to 0.8 and the low search frequency value to 0.2. For the performance evaluation of the extreme cases, we used type 5 and type 6. We used type 7 to compare the characteristics of a user not considering particular types of preferences. To verify the flexibility of the proposed scheme, we conducted experiments with a variety of setup criteria. The search location of the user was 14th Street, New York, NY 10011, United States, and the search radius was within Downtown, New York City.

**Table 3.** Performance evaluation setup values

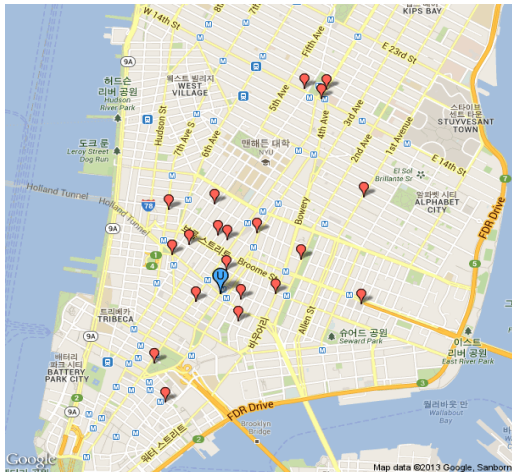
Attribute Type	Price preference	Distance preference	Search frequency
Type 1	0.3	0.7	0.8
Type 2	0.7	0.3	0.8
Type 3	0.3	0.7	0.2
Type 4	0.7	0.3	0.2
Type 5	0	1	0.6
Type 6	1	0	0.6
Type 7	0.5	0.5	0.6

As shown in **Table 3**, types 1 to 4 had different search frequencies but the same preferences, while types 5 to 7 were set to have neutral to extreme preferences. Types 5 and 6 tended to select either distance or price one-sidedly. On the other hand, type 7 was not biased to either price or distance. The result of the nearby information search from the search location showed that a total of 707 candidate locations were searched. The existing scheme and the proposed scheme were applied to these 707 locations, respectively, to produce the top 20 ranking locations. The attribute information of the two results was compared and evaluated. We performed performance evaluation based on the setup shown in **Table 3**, and marked the actual search results to verify the search result intuitively using Google Places API provided by Google, as shown in **Fig. 9**.

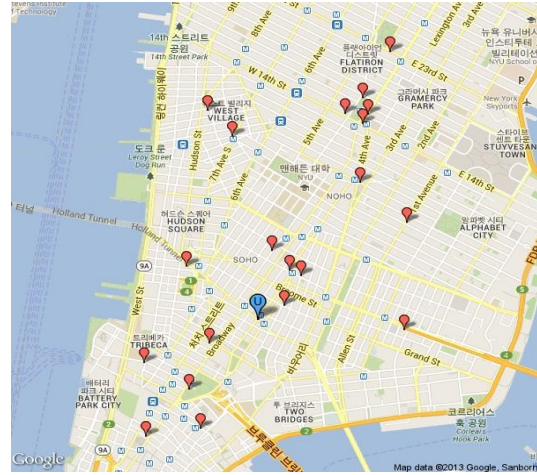


(a) Existing scheme

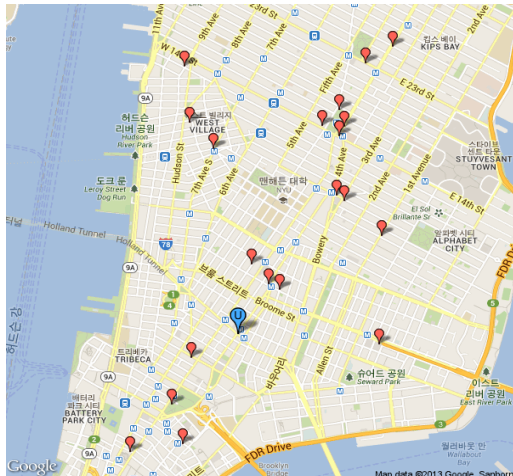
(b) Type 1



(c) Type 2



(d) Type 3



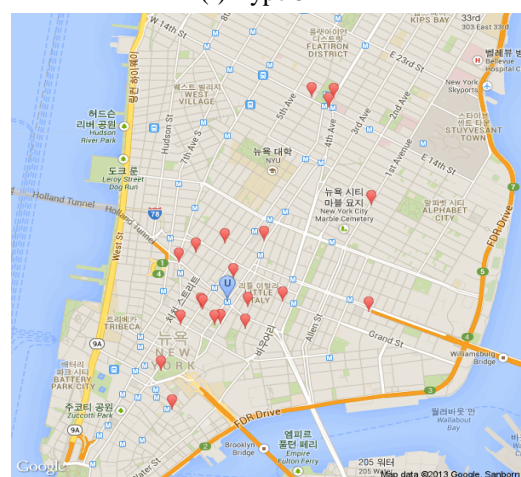
(e) Type 4



(f) Type 5



(g) Type 6



(h) Type 7

Fig. 9. Search result according to preference type



As shown in Equation (4), the proposed scheme generates a search result by applying the search frequency to location preference and popularity. The location preference means how much a user prefer a particular location by considering user preference properties such as distance and price. **Fig. 10** and **Fig. 11** show the top 20 ranking results of the social search. In general, a user prefers a location that is cheaper and closer among the same type of locations. Since the social search provides the personalized results when user preferences are reflected well, the average distance and price of the search result become low. **Fig. 10** shows top 20 average distances according to preference types. As shown in **Fig. 10**, the proposed scheme provides closer locations by 71% and 18% over the existing schemes when the distance preference is high and low, respectively. **Fig. 11** shows top 20 average prices according to the preference types. As shown in **Fig. 11**, the proposed scheme provides cheaper locations by 61% and 13% over the existing schemes when the price preference is high and low, respectively. As a result, the proposed scheme improves 32% average distance and 30% average price over the existing schemes. In **Fig. 10**, since the distance preferences of type 1, type 3, and type 5 are high, their average distances are small over type 2, type 4, and type 6. Especially, type 5 shows a short distance since it considers only the distance preference. The average distance of search results in type 7 is less than those of type 2, type 4, and type 6 that have the small distance preferences. Although the distance preferences of type 2, type 4, and type 6 are low, there is the difference of average distances of search results due to the difference of search frequencies. When only locations with high prices are around a user, faraway locations are provided to a user with only the price preference like type 6. As a result, faraway locations are included in the search result by reflecting the price preference than the distance preference. In **Fig. 11**, the average price of a search result of the proposed scheme changes according to the price preference and the search frequency. However, the average price of a search result of the existing scheme does not almost change. Since type 1, type 3, and type 5 are low over type 2, type 4, and type 6 in terms of the price preference, the average prices of their search results are relatively low. Especially, type 6 shows the cheapest price since it considers only the price preference. The search result of the existing scheme according to the preference does not almost change since it considers not the user distance and the price preference but popularities and expert ratings. Type 5 provides only the expensive locations since it considers only the distance preference. Type 3 also shows the high price preference over the distance preference but when the search frequency is low, it provides locations with high prices since the price preference is less reflected. It is shown that the search result that considers the distance preference reflects the price preference. The proposed scheme generates preferences through a user's visiting records and extracts meaningful information using skyline processing. Therefore, the average distance and price of a search result in the proposed scheme changes by reflecting the preference and search frequency of a user's price and distance. It was verified that preference information was well reflected when the search frequency was high and preference information was well reflected despite a relatively low search frequency. It was also verified that extreme user preferences caused no problems when producing the result values, as shown in types 5 and 6. As a result, it was shown that the proposed scheme reflects user preferences well and changes a search result according to the search frequency. It was also shown that as the search frequency increases, the user preference is reflected well.

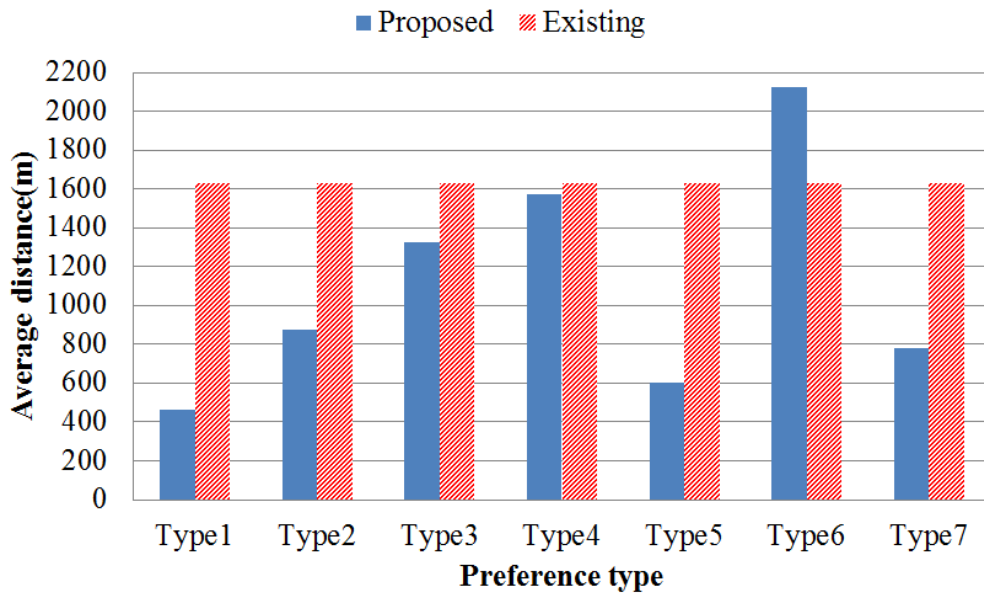


Fig. 10. Average distance of top 20 ranks according to preference type

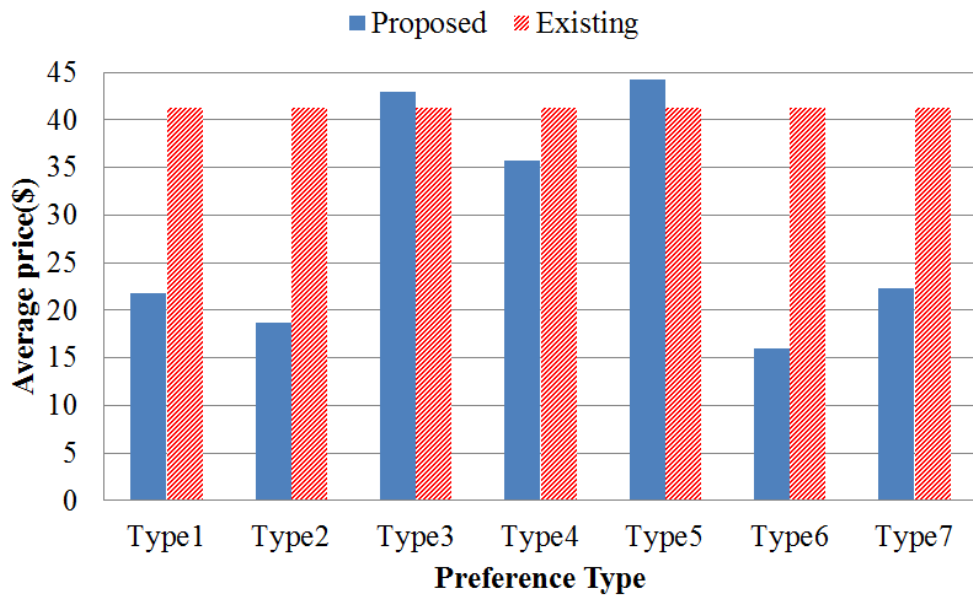


Fig. 11. Average price of top 20 ranks according to preference type

It was verified that extreme user preferences and very low search frequencies were well reflected in the search results through the top 20 results of the social search. As a result, we verified the flexibility of the proposed system. However, it is difficult to say that simply reflecting user information in search results guarantees satisfactory results. Therefore, we carried out an objective evaluation of the locations included in the top 20 ranks. For example, even if a user prefers a cheap restaurant once, it does not mean that s/he wants a cheap location always. In the preference information about restaurants, public reputation is also included

along with price information. If there are no visitors to a restaurant despite a cheap price, it may not produce appropriate search results in response to a user's request.

In order to measure the accuracy of the proposed scheme, we measured the ratio of inclusion of the top 20 locations to the top 50 locations produced by the existing scheme. The compared existing scheme excluded users' personal preference information and determined priority based on public popularity and assessment scores of experts, which is why the search results had high objectivity. Fig. 12 shows the ratio of inclusion of the top 20 locations, as shown in Fig. 10 and Fig. 11, to the top 50 locations produced by the existing scheme. It was shown through performance evaluation that when the preference is the same and the search frequency is different, the ratios that type 1, type 2, type 3, and type 4 can be included in the results of the existing scheme are relatively different by 71% on average. However, type 5, type 6, and type 7 are different within about 4%. Although type 1 is similar to type 2 in terms of the preferences of distance and price, the search frequency of type 1 is high. In addition, although type 2 is similar to type 4 in terms of the preferences of distance and price, the search frequency of type 2 is high. Therefore, the probability that type 3 and type 4 with low search frequency can be included in the results of the existing scheme over type 1 and type 2 is increased. That is, the lower their search frequencies are, the higher the weight of popularity than user preference is. Therefore, they have the similar results as the existing scheme. Type 5, type 6, and type 7 are the same in terms of search frequency although their preference types are different. So the ratios that they can be included in the results of the existing scheme are different each other. However, the difference is very small over other preferences. Through these results, it was concluded that the proposed scheme reflected both user preference information and public popularity sufficiently.

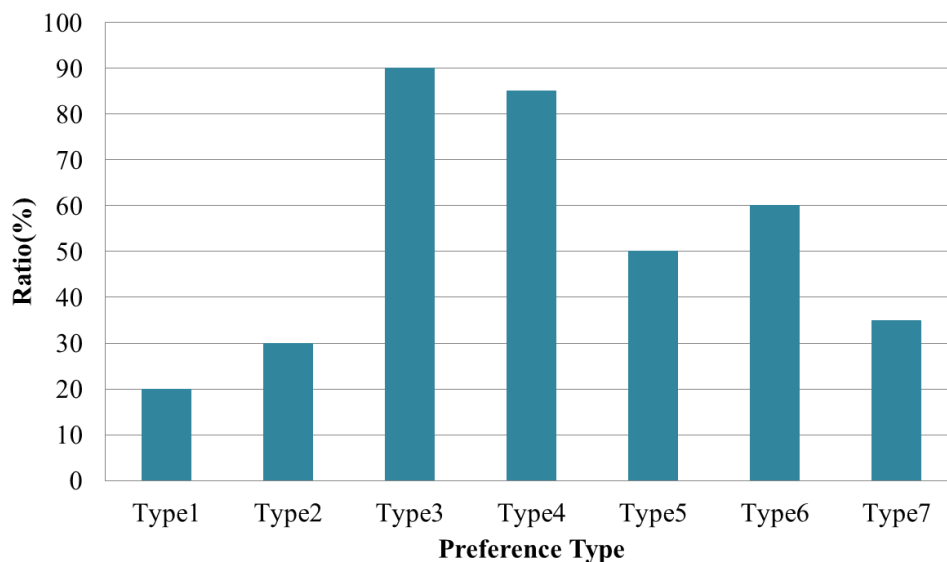


Fig. 12. Search ratio

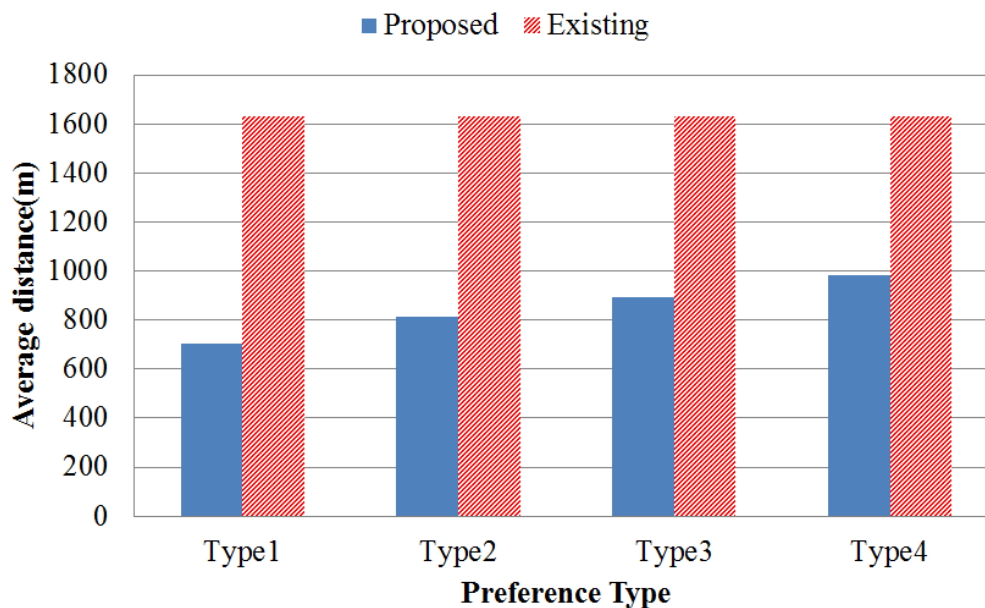
The proposed scheme improves the utilization of search results by using time information and increases the search efficiency by assigning high weights to locations with high popularities. In order to perform performance evaluation according to search times, we set up performance evaluation parameter values as shown in Table 4. We fix a distance, a price, and

a search frequency, and change a search time. The unit of the search time is hour. Type 1 removed time information, while types 2 and 4 reflected the three most active check-in time slots based on the Gowalla dataset.

**Table 4.** Performance evaluation setup values

Attribute	Type 1	Type 2	Type 3	Type 4
Search time(hour)	N/A	12	15	19
Price	0.4			
Distance	0.6			
Search frequency	0.6			
Search location	14th Street, New York, NY 10011, United States			
Keyword	restaurant			

**Fig. 13** and **Fig. 14** shows average distances and prices of top 20 locations according to search times in order to prove the efficiency of the search result. It was shown through performance evaluation that the average distance and price of the proposed scheme reduced by 47% and 44% over the existing scheme when we compare the search results according to search times. Since the existing scheme does not reflect a search time, the average distances according to search times almost do not change. However, the proposed scheme provides the different results according to search times since it considers the popularity at the search time. It was verified that the average values of the search results differed according to time despite the fact that the user attribute information was the same, since a candidate group changes according to time. Since the user preferred distance to price, as shown in **Table 4**, this was marked on a map utilizing the Google Place API, as shown in **Fig. 15**, to verify this intuitively. Type 1 was omitted on the map, because it had the same condition as type 7 in **Fig. 9(h)**.



**Fig. 13.** Average distance according to search time

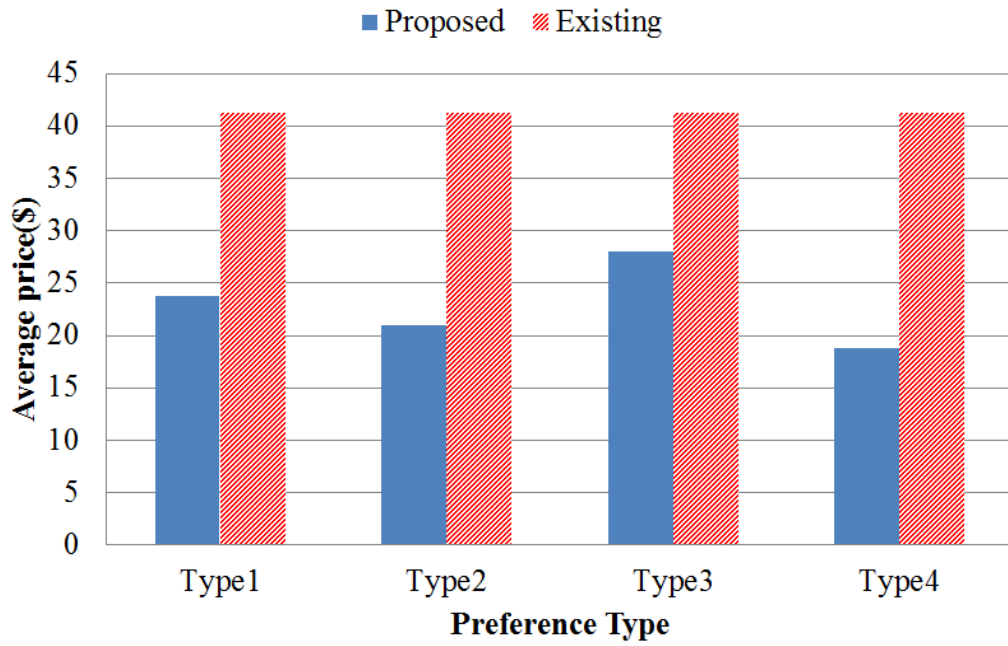


Fig. 14. Average price according to search time



(a) Type 2



(b) Type 3



(c) Type 4

Fig. 15. Change in search results according to time

Table 5 refers to the top four sub-category keywords excluding the search keyword in the top 20 ranks produced by the existing scheme and the proposed scheme. Through the sub-category distribution table of the top search ranks, the way in which time information was reflected in the search results could be determined. This table verified the sub-categories of search results according to a wide query such as restaurant. For example, type 3 showed search results at around 3PM, which verified that many cafés and bakeries were included in the search result. On the other hand, type 4, which was searched at 7PM, had many restaurants, including taverns and hotels unlike the result of type 3. This result verified that user preference, in consideration of user circumstances over time, was well reflected in the search result of the proposed scheme compared to the existing scheme. In addition, the proposed scheme reduced computation time by 20.2% due to time information utilization compared to type 1, which did not utilize time information.

Table 5. Frequencies of sub-categories within search results

Rank	Type 1	Type 2	Type 3	Type 4
1	takeaway	takeaway	cafe	bar
2	cafe	delivery	bakery	nightclub
3	grocery store/ supermarket	cafe	grocery store/ supermarket	convenience store
4	bar	grocery store/ supermarket	convenience store	hotel

### 5. Conclusion

In this paper, we have proposed a social search scheme to improve reliability through an implicit information-collection scheme utilizing skyline processing and reflecting user preference information on locations by receiving feedback. Furthermore, time information was included in queries, thereby providing search results appropriate for moving users

according to search results. In the performance evaluation, we compared the average values according to the attributes of the top 20 ranking locations. The proposed scheme reduced the average value of each attribute by about 15% according to search frequency, which verified that user preference was well reflected. The comparison of the top search results of the proposed scheme with those of the existing scheme showed at least 20% similarity, which guarantees objectivity according to search frequency. These results verified that the proposed scheme not only reflected user preference but also maintained a certain level of objectivity. In addition, it was verified that the proposed scheme not only provided suitable search results according to search time by utilizing time information but also reduced overall computation time by about 20% compared to the scheme that does not consider time information.

## Reference

- [1] P. Shankar, Y. Huang, P. Castro, B. Nath and L. Ifto, "Crowds Replace Experts : Building Better Location-based Services using Mobile Social Network Interactions," in *Proc. of IEEE International Conference on Pervasive Computing and Communications*, pp.20-29, 2012. [Article \(CrossRef Link\)](#).
- [2] A. Khodaei and C. Shahabi, "Social-Textual Search and Ranking," in *Proc. of International Workshop on Crowdsourcing Web Search*, pp.3-8, 2012. [Article \(CrossRef Link\)](#).
- [3] Y. Joung, S. M. Chen, C. Wu and T. H. Chiu, "A Comparative Study of Expert Search Strategies in Online Social Networks," in *Proc. of International Conference on Advanced Information Networking and Applications*, pp.960-967, 2013. [Article \(CrossRef Link\)](#).
- [4] B. Fan, S. Leng, K. Yang and Q. Liu, "GPS: A method for data sharing in Mobile Social Networks," in *Proc. of Networking Conference*, pp.1-9, 2014. [Article \(CrossRef Link\)](#).
- [5] O. Khalid, M. U. S. Khan, S. U. Khan and A. Y. Zomaya, "OmniSuggest: A Ubiquitous Cloud-Based Context-Aware Recommendation System for Mobile Social Networks," *IEEE Transactions on Services Computing*, vol.7, no.3, pp.401-414, 2014. [Article \(CrossRef Link\)](#).
- [6] Y. Wang, A. Nakao and J. Ma, "Socially inspired search and ranking in mobile social networking: concepts and challenges," *Frontiers of Computer Science in China*, vol.3, no.4, pp.435-444, 2009. [Article \(CrossRef Link\)](#).
- [7] P. S. Dodds, R. Muhamad and D. J. Watts, "An experimental study of search in global social networks," *Science*, vol.301, no.5634, pp.827-829, 2003. [Article \(CrossRef Link\)](#).
- [8] I. Guy, M. Jacovi, E. Shahar, N. Meshulam, V. Soroka and S. Farrell, "Harvesting with SONAR: The Value of Aggregating Social Networking Information," in *Proc. of SIGCHI Conference on Human Factors in Computing Systems*, pp.1017-1026, 2002. [Article \(CrossRef Link\)](#).
- [9] N. B. Ellison, C. SteinField and C. Lampe, "The Benefit of Facebook Friends: Social Capital and College Student's Use of Online Social Network Sites," *Journal of Computer Mediated Communication*, vol.12, no.4, pp.1143-1168, 2007. [Article \(CrossRef Link\)](#).
- [10] K. P. Tang, J. Lin, J. I. Hong, D. P. Siewiorek and N. Sadeh, "Rethinking Location Sharing : Exploring the Implications of Social-Driven vs. Purpose-Driven Location Sharing," in *Proc. of International Conference on Ubiquitous Computing*, pp.85-94, 2010. [Article \(CrossRef Link\)](#).
- [11] A. Nagpal, S. Hangal, R. R. Joyee and M. S. Lam, "Friends, Romans, Countrymen : Lend me your URLs. Using Social Chatter to Personalize Web Search," in *Proc. of International Conference on Computer Supported Cooperative Work*, pp.461-470, 2012. [Article \(CrossRef Link\)](#).
- [12] S. Hangal, M. S. Lam and J. Heer, "Muse: Reviving memories using email archives," in *Proc. of ACM symposium on User interface software and technology*, pp.75-84, 2011. [Article \(CrossRef Link\)](#).
- [13] K. Y. Lee and J. L. Hong, "A user survey on search ranking algorithm for social networking sites," in *Proc. of International Conference on Fuzzy Systems and Knowledge Discovery*, pp.995-999, 2012. [Article \(CrossRef Link\)](#).

- [14] J. V. del Campo, J. H. Serrano and J. Pegueroles, "Profile-based Searches on P2P Social Networks," in *Proc. of International Conference on Networks*, pp.98-103, 2010. [Article \(CrossRef Link\)](#).
- [15] H. Hu, J. Feng, S. Liu and X. Zhu, "Social-Aware KNN Search in Location-Based Social Networks," in *Proc. of International Conference on Web-Age Information Management*, pp.242-254, 2014. [Article \(CrossRef Link\)](#).
- [16] Y. A. Kim and G. W. Park, "Topic-Driven SocialRank: Personalized search result ranking by identifying similar, credible users in a social network," *Knowledge Based Systems*, vol.54, pp.230-242, 2013. [Article \(CrossRef Link\)](#).
- [17] A. Kashyap, R. Amini and V. Hristidis, "SonetRank-Leveraging Social Networks to Personalize Search," in *Proc. of International Conference on Information and Knowledge Management*, pp.2045-2049, 2012. [Article \(CrossRef Link\)](#).
- [18] T. Vu and A. Baid, "Ask, Don't Search: A Social Help Engine for Online Social Network Mobile Users," in *Proc. of IEEE Sarnoff Symposium*, pp.1-5, 2012. [Article \(CrossRef Link\)](#).
- [19] Z. Xu, T. Lukasiewicz and O. Tifrea-Marciuska, "Improving Personalized Search on the Social Web Based on Similarities between Users," in *Proc. of International Conference on Scalable Uncertainty Management*, pp.306-319, 2014. [Article \(CrossRef Link\)](#).
- [20] A. Jatowt, Y. Kawai and K. Tanaka. "Temporal ranking of search engine results," in *Proc. of International Conference on Web Information Systems Engineering*, pp.43-52, 2005. [Article \(CrossRef Link\)](#).
- [21] R. Xiang, J. Neville and M. Rogati, "Modeling relationship strength in online social networks," in *Proc. of International Conference on World Wide Web*, pp.981-990, 2010. [Article \(CrossRef Link\)](#).
- [22] R. Akhtar, S. Leng, I. Memon, M. Ali and L. Zhang, "Architecture of Hybrid Mobile Social Networks for Efficient Content Delivery," *Wireless Personal Communications*, vol.80, no.1, pp.85-96, 2015. [Article \(CrossRef Link\)](#).
- [23] J. Cao, Q. Hu and Q. Li, "A Study of Users' Movements Based on Check-In Data in Location-Based Social Networks," in *Proc. of International Symposium on Web and Wireless Geographical Information Systems*, pp.54-66, 2014. [Article \(CrossRef Link\)](#).
- [24] H. Wang, G. Li and J. Feng, "Group-Based Personalized Location Recommendation on Social Networks," in *Proc. of Asia-Pacific Web Conference*, pp.68-80, 2014. [Article \(CrossRef Link\)](#).





**Kyoungsoo Bok** received the B.S. in Mathematics from Chungbuk National University, Korea in 1998 and also received M.S. and Ph.D. in Information and Communication Engineering from Chungbuk National University, Korea in 2000 and 2005 respectively. He is now a research professor in Information and Communication Engineering, Chungbuk National University, Korea. His research interests are sensor network, location based service, mobile ad-hoc network, social network, and big data processing.



**Jongtae Lim** received the B.S., the M.S., and the Ph.D in Information and Communication Engineering from Chungbuk National University, Korea in 2009, 2011, and 2015 respectively. He is now a postdoc researcher in Information and Communication Engineering, Chungbuk National University, Korea. His research interests are Query Processing, Moving Object Database, Spatial Database, Location-based Services (LBS), P2P Networks, Big Data.



**Minje Ahn** received the B.S. and the M.S in Information and Communication Engineering from Chungbuk National University, Korea in 2012 and 2014 respectively. He is now an assistant engineer in S/W Platform Team, Samsung Electronics Co., Ltd, Korea. His research interests are Moving Object Database, Social Network Service, Location-based Services, Big Data Processing.



**Jaesoo Yoo** received the B.S. in Computer Engineering from Chunbuk National University, Korea in 1989 and also received M.S. and Ph.D. in Computer Science from Korea Advanced Institute of Science and Technology, Korea in 1991 and 1995 respectively. He is now a professor in Information and Communication Engineering, Chungbuk National University, Korea. His research interests are database system, storage management system, sensor network, distributed computing, and big data processing.