A User Similarity Calculation Based on the Location for Social Network Services

Min-Joong Lee and Chin-Wan Chung

Department of Computer Science,
Korea Advanced Institute of Science and Technology(KAIST)
335 Gwahangno, Yuseong-gu, Daejeon, Republic of Korea
mjlee@islab.kaist.ac.kr, chungcw@kaist.edu

Abstract. The online social network services have been growing rapidly over the past few years, and the social network services can easily obtain the locations of users with the recent increasing popularity of the GPS enabled mobile device. In the social network, calculating the similarity between users is an important issue. The user similarity has significant impacts to users, communities and service providers by helping them acquire suitable information effectively.

There are numerous factors such as the location, the interest and the gender to calculate the user similarity. The location becomes a very important factor among them, since nowadays the social network services are highly coupled with the mobile device which the user holds all the time. There have been several researches on calculating the user similarity. However, most of them did not consider the location. Even if some methods consider the location, they only consider the physical location of the user which cannot be used for capturing the user's intention.

We propose an effective method to calculate the user similarity using the semantics of the location. By using the semantics of the location, we can capture the user's intention and interest. Moreover, we can calculate the similarity between different locations using the hierarchical location category. To the best of our knowledge, this is the first research that uses the semantics of the location in order to calculate the user similarity. We evaluate the proposed method with a real-world use case: finding the most similar user of a user. We collected more than 251,000 visited locations over 591 users from foursquare. The experimental results show that the proposed method outperforms a popular existing method calculating the user similarity.

Keywords: User similarity, Social network, Location based service.

1 Introduction

Over the past few years, the online social network services such as facebook, twitter and foursquare have been rapidly increasing their territory in the Internet world. With the help of recently growing prevalence of mobile devices, the online social network services have naturally permeated into the mobile devices

J.X. Yu, M.H. Kim, and R. Unland (Eds.): DASFAA 2011, Part I, LNCS 6587, pp. 38–52, 2011.

such as smartphones. Nowadays most of the smartphone users create, share and communicate with other users by using the online social network services at anytime, anywhere.

Moreover, the emergence of the Global Positioning System (GPS) enabled smartphone brings a great opportunity to the online social networks services. The GPS-enabled smartphones are able to acquire their current position through the GPS sensor and tag the acquired location of a device to user generated contents. For instance, when a user writes a post or takes a photo, a smartphone can tag the location of the user on the post or the photo automatically. Especially, if the post is regarding the user's current location or the photo is a landscape of the user's location, the location information will be a huge asset to the online social network. For example, for other online social network users trying to get information about the specific location, this tagged location information can be used to increase the quality of search results.

As the GPS-enabled smartphones become more and more popular, the location based social network service is getting into the spotlight as a new type of the online social network service. For instance, foursquare belongs to this category. foursquare lets a user record a place of one's current location and tell friends where he/she is and leave a short commentary about the place. We will describe the details of foursquare in section 5.1.

In social network services, finding similar users is a very important issue since we can recommend similar users as friends to a new user and recommend a lot of things such as products, search results and experiences. However there are numerous factors to calculate the user similarity. Among the various factors, we utilize the user's location to calculate the user similarity. Since the users carry mobile devices most of the day, especially smartphones, the location of a smartphone has more meaning then just a specific point of the earth. The location of a smartphone is the user's current position and it implies user's interest and life style.

When calculating the user similarity by using the user's location information, the physical location cannot capture the user's real intention why the user visits there. In the real world, one specific physical location is related to many places such as a coffee shop and a theater, and we cannot determine the exact place by using the physical location. This problem is worsened if the user is in a building in a downtown. Therefore, we use the semantics of the location such as the name of the place, and the type of the place to determine the exact place and capture the user's intention. For example, if a user visits a theater frequently, it is reasonable to infer that the user likes watching movies and if a user visits a university regularly, we can infer the user is a student or a faculty member of the university.

In this paper, we propose a new method to calculate the user similarity by using the semantics of the location. We only consider the top-k visited locations of each user. Infrequently visited locations incur incorrect results since people occasionally visit some locations against their will. To take advantage of using the semantics of the location we utilize a location category hierarchy. Also, we

devise the human sense imitated similarity calculation which is able to calculate the interest for another location by using a user's current interest in a certain location.

The contributions of this paper are as follows:

- We address an importance of the semantics of the location than the physical location, and use the semantics of the location to calculate the user similarity. This is the first research that uses the semantics of the location in order to calculate the user similarity.
- We devise an efficient method to calculate the user similarity by using the semantics of the location. In our method, locations and their categories form a hierarchical graph structure. By considering only relevant nodes and computing the similarity at necessary nodes, the proposed method generates the result quickly.
- Our proposed method can also be used to efficiently calculate the similarity between the two objects other than users when the object can be associated with hierarchical categories of elements where each category has a weight.
- We experimentally evaluate the proposed method with a real-world use case: finding the most similar user of a user. Experimental results show that our proposed method is 84% higher in precision, 61% in recall and 72% in f-measure than Jaccard index.

The rest of the paper is organized as follows. In Section 2, we discuss the related works on user similarity calculation. In Section 3, we explain basic concepts and derive basic equations of our proposed method. In Section 4, we describe the details of our proposed method. The experimental results are shown in Section 5. Finally, in Section 6, we make conclusions.

2 Related Work

There have been numerous efforts to calculate the user similarity for different objectives. Recommending people is one of the popular objectives. Guy et al. [4] proposed a method based on the various aggregated information about people relationships but it focused on the people that the user is already familiar with. Therefore, this method cannot be used for calculating the similarity with an unknown user and finding a new friend in the online social network. Terveen et al. [9] proposed a framework called *socialmatching*. The *socialmatching* framework aims to match people mainly using the physical locations of people, while we focus on the semantics of the location.

Some methods recommend experts. McDonald et al. [7] proposed an expert locating system that recommends people for possible collaboration within a work place. Also an expert search engine is described in [2]. The expert search engine finds relevance people according to query keywords. Those approaches are useful to find co-workers or experts but their life style can be varying since the authors focused on a domain to find experts of that domain. Therefore, this approach can not used for finding similar users in general.

Nisgav et al. [8] proposed a method to find the user similarity in the social network. They utilize the user's typed queries to calculate the user similarity. However, since the location based social network is mainly accessed by using the smartphone, typed queries are not much used. In addition, considering the importance of the location information in the location based social network, using the user's queries is not suitable for the location based social network.

The increasing pervasiveness of the location-acquisition technologies such as GPS and WiFi has produced a large amount of location data, and there have been numerous attempts to utilize these location data. Several researchers manipulated and extracted valuable information by using the individual's location data. Chen et al. [1] proposed the raw-GPS trajectory simplification method for the location based social network service. They considered both the shape skeleton and the semantic meanings of a GPS trajectory but they did not report about the semantic meanings of a GPS trajectory.

Also, multiple users' locations have been used to extract meaningful information by several researchers. Krumm et al. [5] described a method that uses a history of the GPS driving data to predict the destination as a trip progresses, and Gonotti et al. [3] developed an extension of the sequential pattern mining paradigm that analyzes the trajectories of moving objects. In contrast to these techniques, Li et al. [6] proposed a framework for mining the user similarity based on the location history. Li et al. extended the paradigm of mining multiple users' location histories, from exploring user's behaviors to exploring a correlation between two users.

The purpose of [6] is similar to ours as finding the similarity between users by using the location history. First, they identified the stay points from the GPS-trajectories and clustered stay points. Then, they matched clustered sequences of two users. The higher user similarity in their framework means two users are physically close such as a family, roommates and lovers, since their framework is based on physical locations. In general, people do not think roommates are similar to each other. On the other hands, people intuitively think two users are similar to each other, if their life styles are alike such as two users both often go coffee shops even if coffee shops are different. Our proposed method efficiently finds the similarity of two users who have similar life styles, since we use the semantics of the location.

3 Preliminary

We explain basic concepts and derive basic equations.

3.1 Location Category

As we mentioned in previous sections, we are using the semantics of the location instead of the physical location. For the use of the semantics of the location, we construct and utilize the location category hierarchy graph. We extract the location category from foursquare since we use a foursquare dataset. A part of the location category hierarchy graph is shown in Fig. 1.

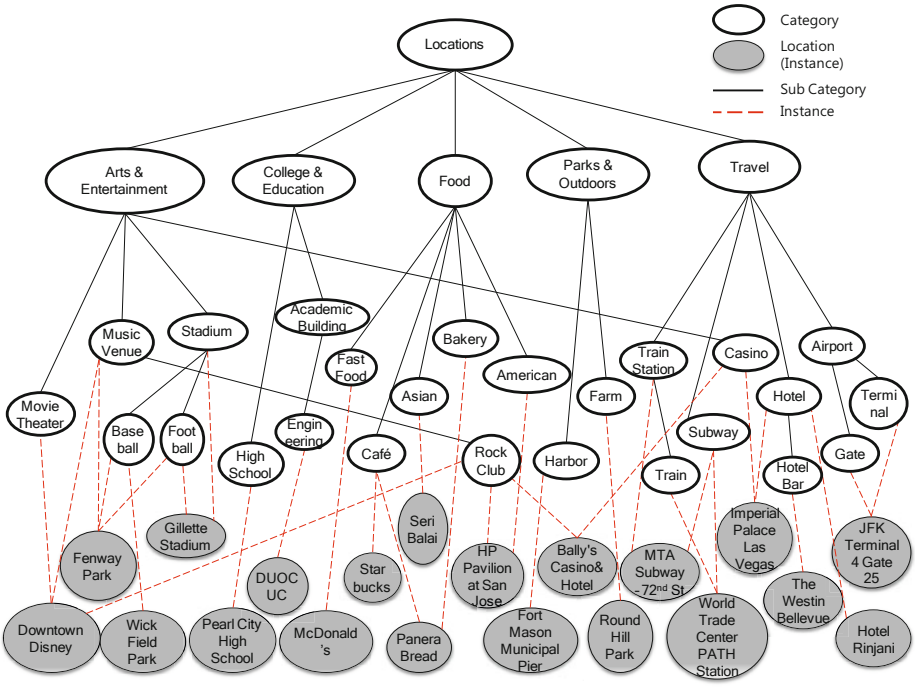


Fig. 1. The location category hierarchy graph (selected)

The location category hierarchy graph consists of two kinds of nodes, *location nodes* and *category nodes*. A *location node* represents the corresponding unique location such as Downtown Disney, Hotel Rinjanis and Gillette Stadium. A *category node* represents a location category such as a movie theater, a hotel and a stadium.

3.2 Significant Score

$SigS_n(u)$ denotes the significant score of node n of user u and it is calculated as follows:

$$SigS_n(u) = \frac{Visit_n(u)}{TotalVisit(u)} \tag{1}$$

where $Visit_n(u)$ is the number of visits at *location node* n of user u, $TotalVisit(u)$ is the total number of visits of user u.

We denote a user's most frequently visited k locations as the $top-k$ *locations* of the user. There are many locations that users visit, but users visit only a few locations frequently. We can consider that the location visited more by a user represents the user's characteristic better than the location visited less, and we experimentally show that the visits to $top-k$ *locations* take up great part of total visits in Section 5.4. To avoid a time-consuming process, we consider only $top-k$ *locations* of a user to calculate the similarity.

3.3 Similarity Score

If two users have their own significant scores at the same *node*, we can compute a similarity score at that node (denote as a *match node*). Let $SimS_n(u, v)$ be the similarity score between user u and user v at *match node* n. To compute $SimS_n(u, v)$, we take the minimum significant score of user u and user v at *match node* n as follows:

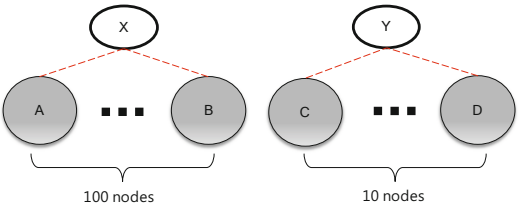
$$SimS_n(u, v) = min(SigS_n(u), SigS_n(v)) \tag{2}$$

We take the minimum value of two users' significant scores since the minimum value intuitively represents two users' common interest at the *match node*.

3.4 Significant Score Propagation

We should take into account miss-matching nodes between two users to get more accurate similarity. For example, consider that two users like watching movies, where one user often goes to 'theater A' and the other user often goes to 'theater B'. In such a case, people intuitively think two users are similar. Furthermore, we can infer that their hobby is watching movies. The human intuition also tells us two users in different theaters are less similar than two users in the same theater.

To imitate this human intuition, we design a method to give the similarity score at the common nearest ancestor node when two nodes are different in the location category hierarchy graph. For instance, 'theater A' and 'theater B' belong to a movie theater category node.



(a) 100 children nodes (b) 10 children nodes

Fig. 2. Different propagation rate according to the number of children nodes

Consider the following two cases depicted in Fig 2. First, the similarity between two users when one user visits location A, corresponding to node A, and the other user visits location B, corresponding to node B, where node A and node B are children of node X, represents category X, which has 100 children (Fig. 2 (a)). Second, the similarity between two users when one user visits location C, corresponding to node C, and the other user visits location D, corresponding to node D, where node C and node D are children of node Y, represents category Y, which has 10 children (Fig. 2 (b)). The first case is more similar than the second case from a probabilistic perspective. The difference between location A and location B in the first case is less than that between location C and location D

in the second case because location A and location B are among 100 locations, while location C and location D are among 10 locations. Therefore, the similarity between two users in the first case is probabilistically higher than that in the second case.

Since each category has various numbers of children, we introduce the logarithm to lessen the effect of various numbers of children nodes. Let $PR(n)$ be the propagation rate of node n. It is calculated as follows:

$$PR(n) = \frac{\log(|Sibling(n)| + 1)}{\log(totalNumberofNodes)} \tag{3}$$

where $Sibling(n)$ is the node n's sibling node set, including node n. We add one to $|Sibling(n)|$ to prevent a case which a dividend becomes zero. $totalNumberof\ Nodes$ is used because $PR(n)$ should be a small number, and $totalNumberof\ Nodes$ is always much bigger than $|Sibiling(n)|$ and easy to obtain. The significant score at node n is multiplied by the propagation rate ($PR(n)$) when node n's significant score propagates to the parent node.

4 User Similarity Calculation

In this section, we first overview our proposed method, and then explain the details.

4.1 Overall Process

The procedure of our proposed method is as follows:

1. Compute the significant score (Equation 1) of each visited location of user u and user v
2. Find $top-k$ *locations* of user u and user v, and construct a top-k significant score table for each user.
3. Construct a location category hierarchy graph by using only visited *location nodes* of two user and visited *location nodes*' ancestor nodes.
4. Find the *match nodes* and its calculation order by using algorithm **MatchNodeOrder()** (Fig. 4).
5. Calculate the user similarity between user u and v by using algorithm **Similarity()** (Fig. 5).

The details of **MatchNodeOrder()** will be discussed in Section 4.2, and the details of **Similarity()** will be discussed in Section 4.3.

4.2 Order of Match Nodes

There are two difficulties to calculate the user similarity. First, as we showed in Fig. 1, the structure of *category nodes* is a tree structure. However, some of *location nodes* have multiple parent nodes since some locations belong to more than one category. If a *location node* has multiple parent nodes, we should select one of the parent nodes to propagate the significant score. Second, the diverse

depths of location nodes make it hard to find match nodes at which two users' significant scores are propagated to come across each other.

Regarding to first difficultly, to overcome the multiple parent nodes problem, we split a *location node* to the number of parent nodes and also the significant score is divided equally among split nodes. This step does not require too much workload since only some of *location nodes* belong to more than one parent node as shown in Fig. 1. Regarding the second difficulty, to efficiently calculate the similarity score at a match node, we find match nodes which need to calculate the similarity score, and the calculation order of match nodes. Without this match node order, we should calculate the similarity score recursively. The algorithm of splitting *location nodes* and finding the match nodes and the order of the match nodes is as follows:

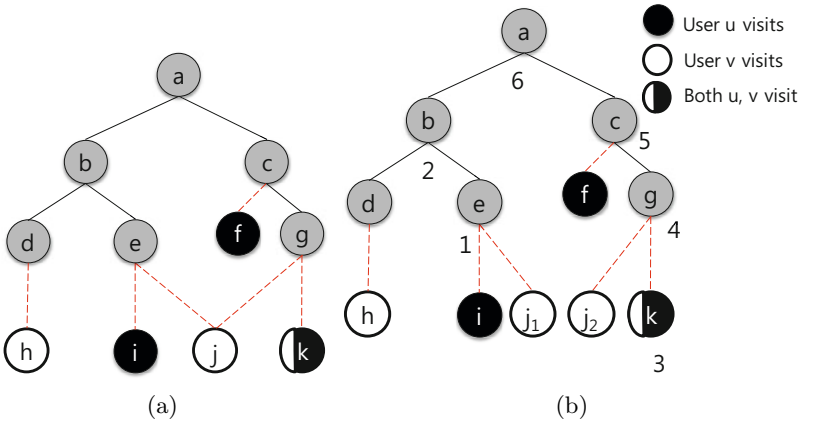


Fig. 3. Example of finding match nodes and its calculation order

Fig. 3 (b) shows an example of finding match nodes and its calculation order. Black colored nodes correspond to user u's visited locations and white colored nodes correspond to user v's corresponding visited locations. The nodes which have a number at the bottom are match nodes and the numbers indicate the calculation order which is the output of the algorithm. We explain the details of the algorithm with an example case in Fig. 3.

As shown in Fig. 3 (a), node j has two parent nodes, e and g. We split node j to j_1, j_2 and make them the children of node e and g, respectively in Fig. 3 (b) (Line(4) - Line(6)). Then, we equally divide the initial significant score of node j into split node j_1, j_2 and store split nodes in the ST_u table (Line(7) - Line(10)). We construct empty sets for each user, $ancestorU$ and $ancestorV$ (Line(13)). For user u, add all ancestor nodes of all visited nodes (a, b, c, e and g) and all visited nodes (f, i and k) to $ancestorU$. And for user v, ancestor nodes (a, b, c, d, e and g) and visited nodes (h, j_1, j_2 and k) are added to $ancestorV$ (Line(14) - Line(19)).

After that, let $MatchNodeSet$ be the intersection of $ancestorU$ and $ancestorV$ (Line(20)). In this example, $MatchNodeSet$ is $\{a, b, c, e, g, k\}$ node elements

```
Algorithm MatchNodeOrder()
Input location category hierarchy graph G
      user u's top-k significant score table ST_u
      user v's top-k significant score table ST_v
Output match node order list m
begin
1.   Let list m be an empty node list
2.   Let ST_u[n] be a significant score of user u at node n
     /* to handle multiple parent nodes problem */
3.   foreach leafnode l in graph G
4.      if location node l has more than one parent node
5.         pNum := location node l's parent nodes count
6.         Split node l to l_1, l_2, ..., l_pNum as each parent's child node
           /* Do following if step on ST_v */
7.         if ST_u contains score for node l
8.            Add l_1, l_2, ..., l_pNum to ST_u with ST_u[l]/pNum score
9.            Delete node l and its significant score from ST_u table
10.        endif
11.     endif
12.  endforeach
13.  Let MatchNodeSet, ancestorU and ancestorV be empty node sets
14.  foreach node n in ST_u
15.     Add node n's ancestor nodes to ancestorU
16.  endforeach
17.  foreach node n in ST_v
18.     Add node n's ancestor nodes to ancestorV
19.  endforeach
20.  MatchNodeSet := ancestorU ⋂ ancestorV
21.  m := Sort MatchNodeSet in post-order by using graph G structure
22.  return m
end
```

Fig. 4. Algorithm of finding match nodes order

which is intersection of $\{a, b, c, e, g, i, k, f\}$ and $\{a, b, c, d, e, g, j_1, j_2, h, k\}$. Finally, a post-order list of $MatchNodeSet$ elements are assigned in a list m and returned as an output of this algorithm (Line(21) - Line(22)). As shown as numbers in Fig. 3 (b), the calculation order list is $<e, b, k, g, c, a>$.

4.3 User Similarity Calculation

After we determine the match nodes and its calculation order, we can efficiently calculate the user similarity. The algorithm of calculating the user similarity is as follows:

For efficiency, we devise the multiple propagation rate based on Equation 3 for propagating a significant score of a node to its ancestor node through multiple levels. $MPR(n, v)$ denotes the multiple propagation rate, from node v to ancestor node n, and it is calculated as follows:

$$MPR(n,v) = \frac{\log(|Sibling(k_1)|+1) \times \log(|Sibling(k_2)|+1) \times ... \times \log(|Sibling(k_n)|+1)}{depthDiff \times \log(totalNumberofNodes)} \tag{4}$$

where $depthDiff$ is the depth difference between node n and node v, $Sibling(n)$ is the number of node n's sibling nodes. k_1 is node v, k_2 is the parent node of v, ... , and node $k_{depthDiff}$ is the node n since $depthDiff$ is the depth difference between node n and node v.

```
Algorithm Similarity()
Input location category hierarchy graph G
        user u's top-k significant score table ST_u
        user v's top-k significant score table ST_v
        match node order list m
Output User similarity score SimScore
begin
1.    Let ST_u[n] be a significant score of user u at node n
2.    SimScore := 0.0
      /* Calculate user similarity at each node in list m in order*/
3.    foreach node n in list m
4.       descendants := the set of descendant of node n (use graph G structure)
5.       foreach node d in descendants
            /* Do following if step on ST_v */
6.          if ST_u has node v
7.             if ST_u does not have node n
8.                Add node n to ST_u
9.                ST_u[n] : = 0.0
10.            endif
               /* propagate all descendants significant score to node n */
11.            ST_u[n] := ST_u[n] + ST_u[d] * MPR(n, d) (Equation 4)
12.            Delete node d from ST_u table
13.         endif
14.      endforeach
15.      SimScore := SimScore + SimS_n(u, v) (Equation 2)
16.      ST_u[n] := ST_u[n] - SimS_n(u, v)
17.      ST_v[n] := ST_u[n] - SimS_n(u, v)
18.   endforeach
19.   return SimScore
end
```

Fig. 5. Algorithm of the user similarity calculation

The user similarity calculation algorithm (Fig. 5) utilizes match node order list m (output of Fig. 4 algorithm) as one of the inputs. At the beginning of the algorithm, we initialize $SimScore$ with zero (Line (2)) and enumerate each node n in list m (Line (3) - Line (18)). At the beginning of enumeration steps, let $descendants$ be the descendant node set of node n (Line (4)). For user u, if ST_u has node v (Line (6)), and if ST_u does not have node n we make a empty table entry for node n to store the propagated score of descendant nodes (Line(7) - (10)). Then we propagate user u's significant scores at node v to the match node n using Equation 4 (Line (11) - Line(12)). We calculate the similarity score at node n and add the similarity score to $SimScore$ (Line (15)), and we subtract similarity score $SimS_n(u, v)$ from two user's significant scores since the similarity score $SimS_n(u, v)$ is already added to the user similarity score $SimScore$. (Line (16) - Line (17)). Finally, the algorithm returns $SimScore$ as the output (Line(19)).

5 Experiment

We evaluate the proposed method with a real-world use case. We use foursquare user's data as a dataset. Before discussing about the experimental results, we briefly introduce our dataset.

5.1 Dataset

foursquare is one of the most spotlighted location based social network services. We briefly introduce foursquare since we collect data from foursquare for evaluating our method.

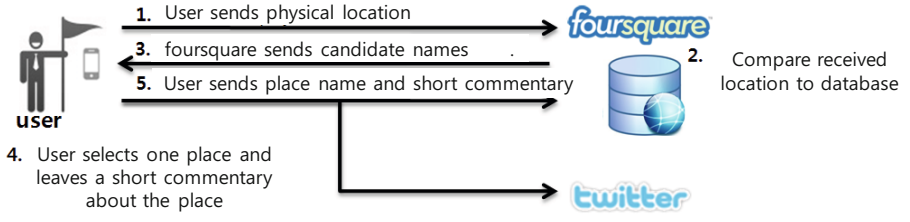


Fig. 6. Diagram of foursquare service

Fig. 6 shows a diagram of foursquare's 'check-in'[1], feature. When a user tries to 'check-in' to a certain place, the user sends an exact physical location of the user (Step 1). Then foursquare compares the received location with their huge venue database (Step 2) and suggests a few names of places in distance order (Step 3). After that, the user selects one place name for his/her current locations, also the user can make a short commentary about the place (Step 4). After that, the user sends the selected place name and a commentary to foursquare and twitter (Step 5). Therefore, by using foursquare APIs[2] (Step 1-3), we can convert the physical location to the semantics of the location.

We collect users' visited locations through users' twitter pages since users also post 'check-in' information to their twitter accounts (Step 5 in Fig. 6). At first, we collected 1,358,287 visiting locations over 17,863 users. However, most of users are not active enough to use their recoeds as a dataset. Therefore, we select 591 users based on their activity. The selected users visited 251,053 locations and they are distributed around the world.

5.2 Finding a Similar User

Fig. 7 shows an example of two similar users which are selected by our method. User A and user B live in very different locations, but they are similar because they are both students and they like to go shopping. By only considering the physical location of two users, the similarity score between user A and user B is close to zero. However, our proposed method finds the similarity between them since our method utilizes the semantics of the location.

5.3 Performance of Proposed Method

We experimentally evaluate the proposed method with a real-world use case; finding the most similar user of a user. We compare our method to Jaccard

[1] Records a user's place and able to leave a short commentary about the place.
[2] `http://groups.google.com/group/foursquare-api/web/api-documentation`

# of visits	Location name	Category	Region
34	Universitas Kristen Petra – Gedung T	Academic Building, University, Math	East Java
29	Petra Christian University	University	East Java
18	CITO (City of Tomorrow)	Mall	East Java
13	Alfamidi	Other – Shopping	East Java
13	Pakuwon	Mall	Indonesia

(a) User A

# of visits	Location name	Category	Region
49	Pearl City High School	High School	HI
37	Waimalu Plaza	Mall	HI
35	Pearl City Cultural Center	Concert Hall, Event Space	Hawaii
32	Foodland	Other – Shopping, Bakery, Seafood	Hi
31	Westridge Shipping Center	Mall	Hi

(b) User B

Fig. 7. Example of two similar users selected by our method

index which is a popular method to calculate the similarity. As we utilize the semantics of the location to calculate the user similarity for the first time, there is no existing method to be compared with.

To find the most similar user of a user, firstly, we calculate pairwise the user similarity between 591 users. Then, we select the most similar user to each of 591 users. After that, we compare a user's visited locations with the most similar user's visited locations for every user.

In order to measure the accuracy of two methods, we compute the precision, recall and F-measure by comparing his/her visited locations with the visited locations of the most similar user recommended by each method.

Let $f(u)$ returns the set of categories of the top-k locations of user u. The precision is calculated as follows:

$$Precision = \frac{|f(u) \bigcap f(u_r)|}{|f(u_r)|} \tag{5}$$

where u_r is the most similar user selected by a method, which can be our method or Jaccard index.

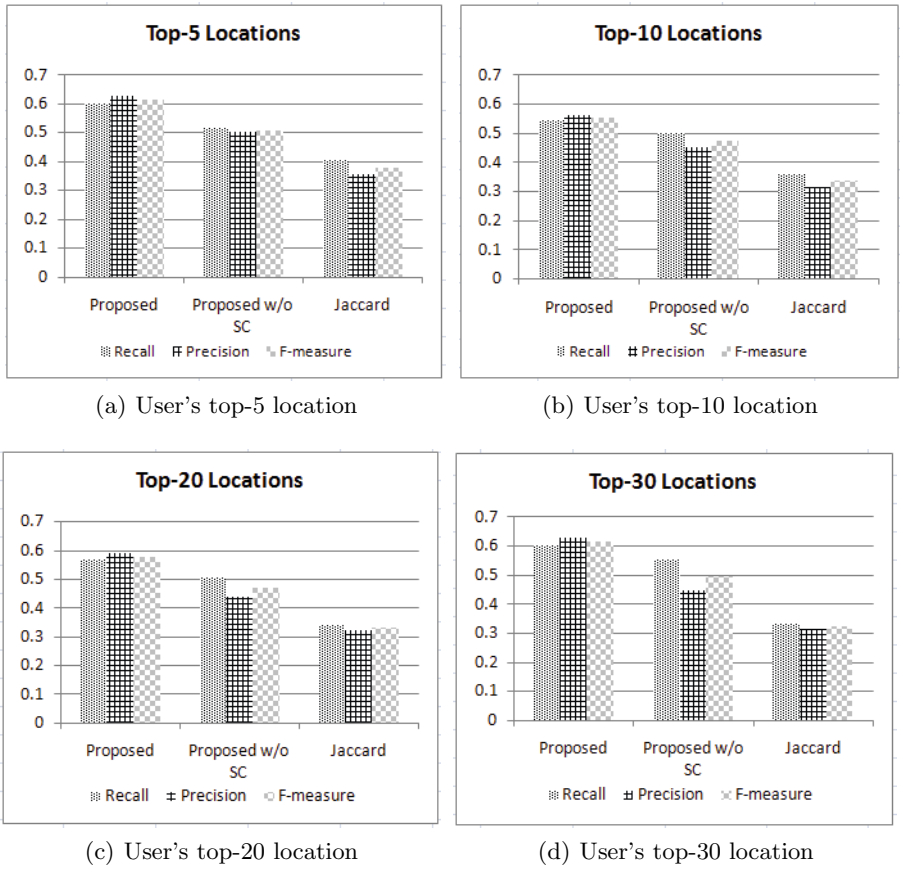
The recall is calculated as follows:

$$Recall = \frac{|f(u) \bigcap f(u_r)|}{|f(u)|} \tag{6}$$

The F- measure is calculated as follows:

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{7}$$

Then, we average the precisions, recalls and F-measures for all users.



(a) User's top-5 location (b) User's top-10 location

(c) User's top-20 location (d) User's top-30 location

Fig. 8. Experimental results with various top-k locations

As shown in Fig. 8, our proposed method outperforms the Jaccard index for every different top-k location setting. Our method is 84% higher in the precision, 61% in the recall and 72% in the f-measure than the Jaccard index on the average. Since a frequency of visits to a location is represented by a binary value in Jaccard index, we make our proposed method not to use significant scores and compare with Jaccard index. This modified version of our method is labeled as *Proposed w/o SC* in Fig. 8. However, the result that our method shows higher performance than the Jaccard index remains unchanged.

5.4 Top-k Location

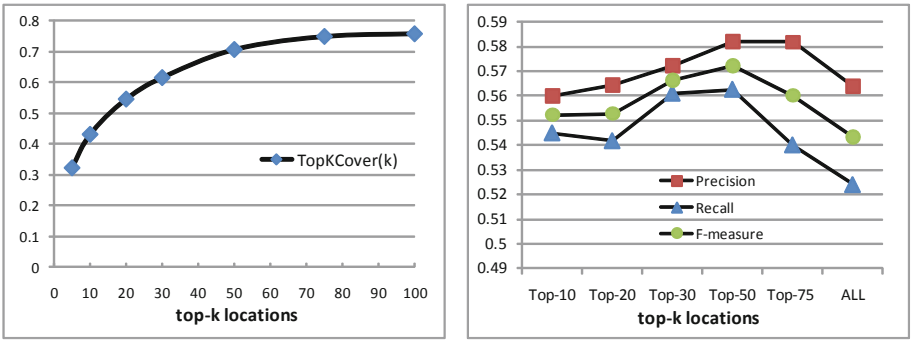
Since we use only the $top-k$ *locations* of a user to calculate the similarity, we experimentally show that considering only the $top-k$ *locations* of a user results is better than considering all the visited location of a user.

To show the visits to $top-k\ locations$ are greater part of total visits, we devise $TopKCover(k)$ which shows the ratio of $top-k\ locations$ visits to total visits.

$$TopKCover(k) = \frac{\sum_{u \in U}(\frac{TopkVisit_k(u)}{TotalVisit(u)})}{|U|} \quad (8)$$

where U is the set of all users, $TopkVisit_k(u)$ is the user u's number of visits to $top-k\ locations$ of user u, $TotalVisit(u)$ is the total number of visits of user u.

From the result of Fig. 9 (a), we can consider that using more than 100 visited locations to characterize a user is meaningless, also the small number of top-k locations covers a large part of visits. The top-5 locations cover 32%, top-10 cover 43%, top-20 cover 55% and top-30 cover 62% of visits.



(a) ratio of $top-k\ locations$ visits to total visits

(b) precision, recall, F-measure on various top-k settings

Fig. 9. Two experimental results to select proper top-k

Fig. 9 (b) shows that considering only the $top-k\ locations$ of a user is better than considering a large number or all of the user's visits. All the three measure start dropping after top-50 locations.

6 Conclusion

In this paper, we proposed an accurate and efficient user similarity calculation method. Our method utilizes the semantics of the location, while the other existing previous researches have been focused on only the physical location. We also utilize the location category hierarchy to semantically match locations, and the experimental results show that the proposed method outperforms the popular Jaccard index. We also experimentally show how many numbers of the locations has the meaning to a user, and it helps us to understand the user's behavior. As a future work, we would aggregate the semantics of the location with some other information such as user generated tags to get more accurate results.

Acknowledgments. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government (MEST) (No. 2010-0000863).

References

1. Chen, Y., Jiang, K., Zheng, Y., Li, C., Yu, N.: Trajectory simplification method for location-based social networking services. In: International Workshop on Location Based Social Networks, pp. 33–40 (2009)
2. Ehrlich, K., Lin, C.Y., Griffiths-Fisher, V.: Searching for experts in the enterprise: combining text and social network analysis. In: International ACM SIGGROUP Conference on Supporting Group Work, pp. 117–126 (2007)
3. Giannotti, F., Nanni, M., Pinelli, F., Pedreschi, D.: Trajectory pattern mining. In: 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 330–339 (2007)
4. Guy, I., Ronen, I., Wilcox, E.: Do you know?: recommending people to invite into your social network. In: International Conference on Intelligent User Interfaces, pp. 77–86 (2009)
5. Krumm, J., Horvitz, E.: Predestination: Inferring destinations from partial trajectories. In: 8th International Conference on Ubiquitous Computing, pp. 243–260 (2006)
6. Li, Q., Zheng, Y., Xie, X., Chen, Y., Liu, W., Ma, W.Y.: Mining user similarity based on location history. In: 16th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, p. 34 (2008)
7. McDonald, D.W.: Recommending collaboration with social networks: a comparative evaluation. In: Conference on Human Factors in Computing Systems, pp. 593–600 (2003)
8. Nisgav, A., Patt-Shamir, B.: Finding similar users in social networks: extended abstract. In: 21st Annual ACM Symposium on Parallel Algorithms and Architectures, pp. 169–177 (2009)
9. Terveen, L.G., McDonald, D.W.: Social matching: A framework and research agenda. ACM Trans. Comput. -Hum. Interact, 401–434 (2005)