One-day Trip Recommendation for Nearby Spots Based on Users' Locations and Preferences

Siya Bao[1] Masao Yanagisawa[1] Nozomu Togawa[1]

Abstract: Over the past decades, advanced digital applications have widely available to users. Some of the applications enhance the possibility of generating personalized trip routes. In this paper, we propose an algorithm of which aim is to plan various trip routes to help users visit different landmarks that could be really interesting and achievable. The algorithm is based on capturing the information of users and landmarks, which determines lists of landmarks worthy of visiting. Experimental results demonstrate the effectiveness of the proposed algorithm.

1. Introduction

Trip route recommendations can strongly influence users' satisfaction and the success of touristic businesses. It is important to consider the users' perspective and, for example, evaluate which landmarks are worth a detour. Thereby this arises the need and creation of a variety of personalized trip recommendation systems.

The personalized electronic trip recommendation is a device that addresses this problem by suggesting a (near) optimal selection of several landmarks and a sequential and consecutive route with recommended landmarks, taking into account the transportation, the distances and personal preferences etc. The underlying selection and routing problem is called the tourist trip design problem (TTDP) [1].

Park et al. [2] propose a system that reflects users' preferences on restaurant type, price, mood and current distance with mobile context information records through Bayesian Networks. Ye et al. [3] suggest that instead of solely using the ratings from a user, recommendations are more efficient and reliable with taking ratings from users' friends. Wei et al. [4] propose the Route Inference framework based on Collective Knowledge (RICK) to construct popular routes from uncertain trajectories. Ge et al. [5] integrate energy consumption into a mobile recommender system by learning energy-efficient transportation patterns from trajectories.

In this paper, based on the TTDP, we propose an algorithm that is tailored to users' current location information and personal interest ratings as key criteria for nearby spot recommendations and maximize the user's satisfaction within the recommended trip route.

Based on the current studies on TTDP, the abstract model of algorithm can be summarized in the two steps below:

[1] Dept. of Computer Science and Communications Engineering, Waseda University

(1) Retrieving and evaluating of landmarks based on user' preferences [2], [3], [6], [7]
(2) Clustering the landmarks to form a sequential trip route [4], [5], [8]

We first capture the users' preferences on eight types of landmarks through questionnaires. These preferences are used to predict a personal interest weight (score) for each surrounding landmark and select the *top-k* candidate regions with the most interesting landmarks. Next, we explore the spatial characteristics of the candidate regions and merge these regions into several clusters. Finally, in the light of the clusters, we propose a routing algorithm to construct a routable graph, and a list of visiting landmarks with visiting orders is provided for the user.

2. One-day Trip Recommendation

In this section, we propose a one-day trip recommendation algorithm. Fig. 1 overviews the components of the proposed algorithm.

Firstly Section 2.1 describes the structure of landmarks and weight landmarks with the specific user profile. Next, in Section 2.2, we partition the user's current space into disjoint regions and then score these regions by pre-evaluated landmarks obtained from Section 2.1. Section 2.3 describes how we inference the spatial connection between regions as preparations for the further route graph. The algorithm presented in Section 2.4 constructs a routable graph and proposes a trip route to the user.

2.1 Landmark categorization and user file

We set the user's current position S as the starting point and the visiting area is set to be a square around S with the width of 10km and height of 10km. Then we establish a landmark database with all the landmarks in the area. Let L be a set of all the landmarks in this area. Let LT be

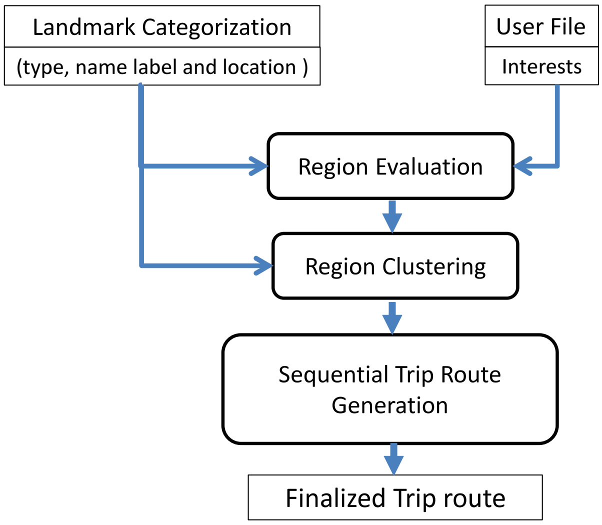


Fig. 1: Algorithm overview.

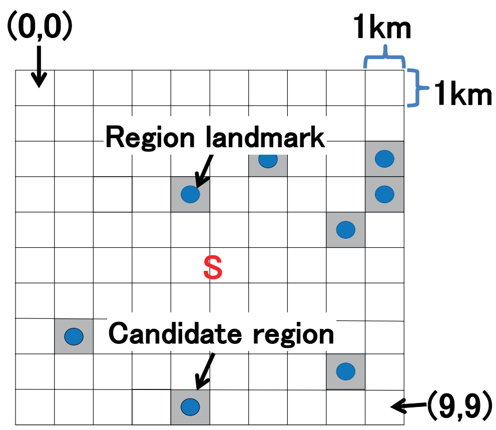


Fig. 2: Region division where the number of candidate regions is 8 ($k = 8$).

a set of eight landmark types, which are *History, Nature, Entertainment, Art, Sport, Food and Drink, Shopping*, and *Night Life*. Every landmark $\ell \in L$ is characterized by its type(s) t, its name label and its coordinates. Note that, a landmark can have more than one types, for example, Meiji Jingu shrine has the two landmark types of *History* and *Nature*.

After collecting nearby landmark information, then it comes to assigning a rating (weight) to a landmark type. The user creates a user profile with his/her interest weight $w(t)$ for every landmark type t by five-point scale, with an explicit definition of each of the possible weights: very dislike (1), dislike (2), fair (3), like (4), and very like (5).

2.2 Region evaluation

As in [9], we divide the given 10km×10km area into 100 small regions with the width of 1km and height of 1km (see Fig. 2). Then each region includes at least one landmark under this scale in most of the cases.

With user's preferences, the system matches the personal preferences in the each landmark. Then we can evaluate each region with a possible visit score as follows:

Let R be a set of such small regions. For a region $r \in R$, we assign it with a visit score $visit(r)$:

$$visit(r) = \sum_{t \in LT} |r(t)| \times w(t) \quad (1)$$

where $r(t)$ is a set of landmarks of which landmark type is t inside the small region r. The landmark of each small region r with the highest $w(t)$ value is called the *region landmark* $p(r)$ in this region.

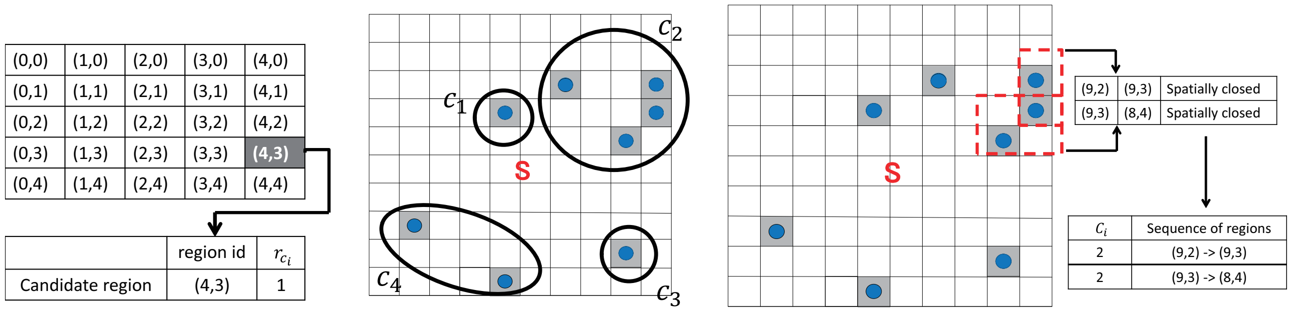
For a one-day trip, the recommended number of landmarks for visiting is usually around 5–10 in the most current trip recommendation systems or websites. Therefore we select k regions with the 1st to k-th highest visit scores given by Eq. (1) as *candidate regions*. Note that the user may refine the selection of regions and change the number k of visiting regions by himself/herself.

2.3 Region clustering

After retrieving the k candidate regions, at this stage, we merge these individual regions to form a geographical cluster c_i where $1 \leq i \leq 4$.

We first observe the spatial characteristics of the candidate regions. We define 10 terms for constructing connected geographical areas. Table 1 summarizes the notations used in this paper.

First, as mentioned in Section 2.1, we divide the given area into 100 disjoint regions evenly, and then the *id* of a region r is represented by (x, y) from (0,0) to (9,9) (see Fig. 2). Given a region $r = (x, y)$, the region r belongs to the cluster c_1 if $x \in [0, 4], y \in [0, 4]$ (see Fig. 3(a)), r belongs to cluster c_2 if $x \in [5, 9], y \in [0, 4]$, r belongs to cluster c_3 if $x \in [5, 9], y \in [5, 9]$, and r belongs to cluster c_4 if



(a) Region in the cluster 1. (b) Completed region clustering. (c) Spatial clustering in the cluster 2.

Fig. 3: An example of the region clustering ($k = 8$).

Table 1: Notations

Symbol notation	Description
S	User current location
(x, y)	Region id
r_{c_i}	The cluster that the region r locates in
$r_{c_i}^{in}$	The entrance region in the cluster c_i
$r_{c_i}^{out}$	The exit region in the cluster c_i
$r \to r'$	Sub-route from r to r'
$dis(r, r')$	The distance from r to r'
C_i	Sub-route in the cluster c_i
$C_i \to C_j$	Sub-route from the cluster c_i to c_j
$CR(c_i)$	The candidate regions in the cluster c_i

$x \in [0, 4], y \in [5, 9]$. An example of completed clustered regions are shown in Fig. 3(b). Let $CR(c_i)$ be a set of *candidate regions* belonging to a cluster c_i. Then, if $CR(c_i) = \emptyset$, the cluster c_i is marked as a blank cluster which will not be considered in the further route searching.

Next we merge the connected geographical regions in each cluster c_i Given two candidate regions $r = (x, y) \in CR(c_i)$ and $r' = (x', y') \in CR(c_i)$, if $|x - x'| \leq 1$ and $|y - y'| \leq 1$, then r and r' are called spatially closed, and in the cluster c_i, there must be a sub-route traversing from r to r', which is denoted as $r \to r'$ in C_i (see Fig. 3(c)).

2.4 Sequential trip route generation

Once all the candidate regions are clustered and merged, then we explore the sub-routes in/between the clusters. These sub-routes can be categorized into the two types: inside a cluster and between clusters. The information inferred for a sub-route comprises a moving direction and a transition support indicating the transition relationship between two regions/clusters.

For the sub-routes inside the cluster c_i , we first determine the entrance region $r_{c_i}^{in}$ and the exit region $r_{c_i}^{out}$ by setting four threshold regions of r_{th1}=(2,4), $r_{th2} = (5, 2)$, r_{th3}=(7,5) and r_{th4}=(4,7) as follows (see Fig. 4(a)):

$$r_{c_i}^{in} = \min_{r \in CR(c_i)} dis(r, r_{thi}) \quad (2)$$

$$r_{c_i}^{out} = \min_{r \in CR(c_i)} dis(r, r_{th(i+1)}) \quad (3)$$

In addition, in case of c_4, $r_{th(i+1)} = r_{th1}$. An example of determined entrance and exit regions are shown in Fig. 4(b).

Let C_i be a route for every cluster c_i, which is a sequence of regions. C_i is denoted by:

$$C_i = \{r_{c_i}^{in} \to r_{c_i}^{j} \to \cdots \to r_{c_i}^{m} \to r_{c_i}^{out}\}. \quad (4)$$

Once accomplished selections of the entrance and exit regions, then we search from the entrance region and exit region by turns until every candidate region $r \in CR(c_i)$ is included in the route C_i.

For example, in the case of the cluster c_2 in Fig. 4(c), we start from $r_2^{in} = (6, 2)$, the next nearest region excluding $r_{c_2}^{out}$, is $r = (9, 2)$, and then we initially generate C_2 as:

$$C_2 = \{r_{c_2}^{in} = (6, 2) \to r = (9, 2)\}. \quad (5)$$

As $r = (9, 3)$ is spatially closed to $r = (9, 2)$, then we update C_2 as:

$$C_2 = \{r_{c_2}^{in} = (6, 2) \to r = (9, 2) \to r = (9, 3)\}. \quad (6)$$

There is none of unsearched candidate regions left in c_2 except for $r_2^{out} = (8, 4)$, and then the finalized route C_2 is given by:

$$C_2 = \{r_{c_2}^{in} = (6, 2) \to r = (9, 2) \to r = (9, 3) \to r_{c_2}^{out} = (8, 4)\}. \quad (7)$$

In addition, in the real implementation, the sub-route between any two regions r and r' are calculated as the shortest distance between region landmarks $p(r)$ and $p(r')$.

After searching a route C_i in every cluster c_i, we construct sub-routes between clusters with a clockwise direction, which is a more favorable traveling option [10], in an order of $\{c_1 \to c_2 \to c_3 \to c_4\}$. We start from c_1. If $CR(c_2) \neq \emptyset$, then we connect C_1 and C_2 by connecting $r_{c_1}^{out}$ with $r_{c_2}^{in}$. Otherwise, we search for the next non-empty cluster. Then we repeat the connection step until all non-empty clusters are connected (see Fig. 4(d)).

Finally, we connect the $r_{c_1}^{in}$ and $r_{c_4}^{out}$ to the current position S. In addition, if the distance between any two region landmarks is longer than 1km, then the sub-route is black which suggests using a bus/train as the transportation option, otherwise, it is yellow which suggests using walking as the transportation option (see Fig. 4(e)).

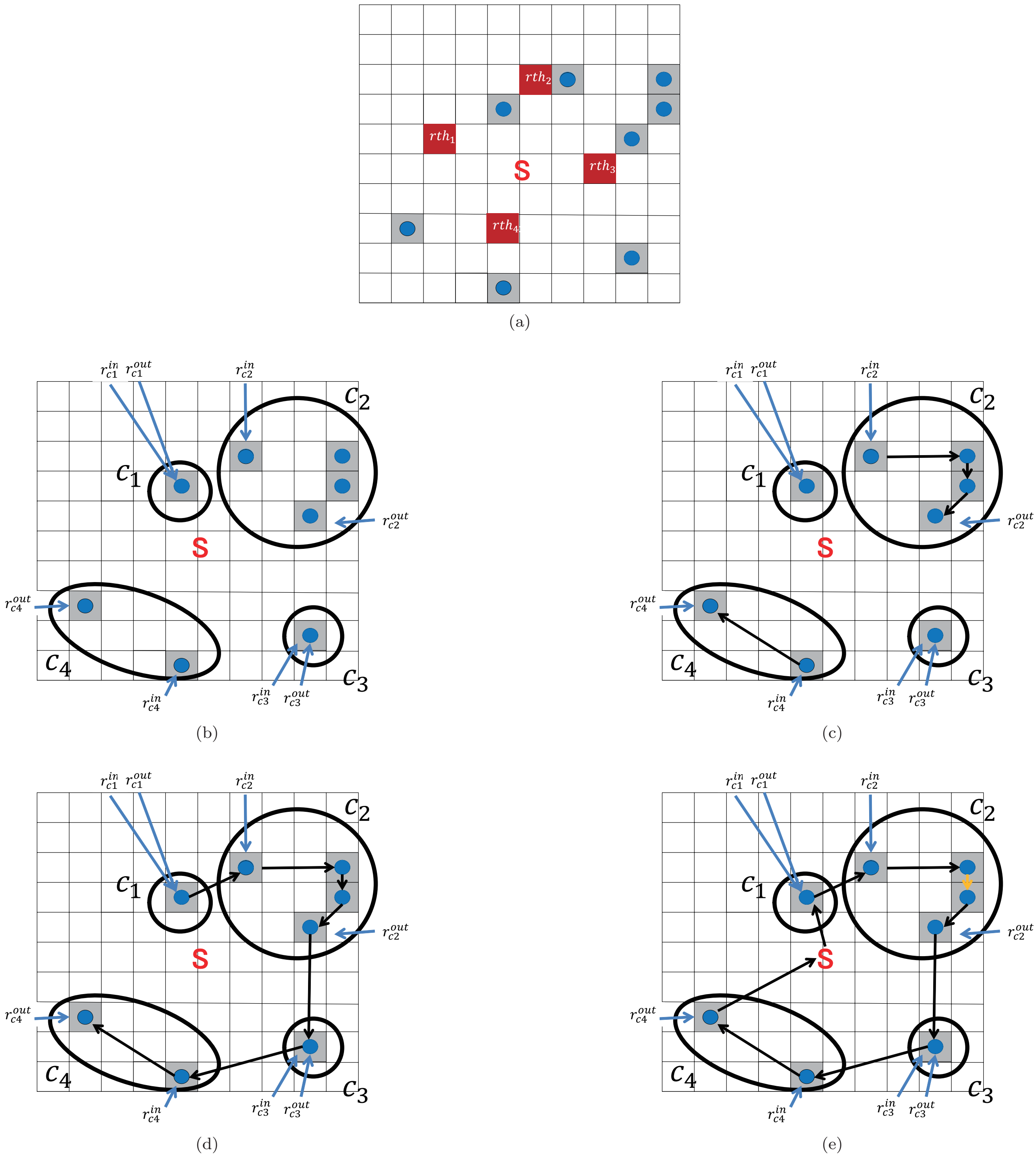


Fig. 4: Generating a hierarchical trip route ($k = 8$).

2.5 Algorithm summary

Our algorithm can be summarized as follows:

(1) Categorize the given landmarks into eight types and rank them with user preferences (Section 2.1).

(2) Divide the given area into small regions and select the most interesting regions as candidate regions (Section 2.2).

(3) Using location-based clustering, we cluster the candidate regions obtained in Section 2.2 into four geo-spatial clusters c_1, c_2, c_3, c_4 (Section 2.3).

(4) Let $CL = \{c_1, c_2, c_3, c_4\}$. Every $c_i \in CL$ contains several candidate regions and then we generate a sub-route C_i within the cluster c_i as a sequence of region landmarks in c_i (Sections 2.3–2.4).

(5) We generate a finalized route T as a loop sequence of the starting point and the clusters, i.e., $T = \{S \to C_1 \to C_2 \to C_3 \to C_4 \to S\}$ (Section 2.4).

3. Implementation and Evaluation

In order to demonstrate the usefulness of the proposed algorithm, one-day-trip cases with 8 different user profiles are discussed. We set the experiment area in Shinjuku area with the height of 10km and the width of 10km, where starting point S is at the Shinjuku railway station. Our goal is to offer a user a trip route with visiting the *top-k* candidate regions. In addition, every candidate region includes just one interesting landmark in this experiment. The algorithm is coded in Java.

Table 2 shows the landmark types and landmarks in this experiment and Table 3 lists 8 user profiles. The result routes are presented in Table 4 with visiting orders and an example of the generated route for the user 01 is portrayed in Fig. 5, red dots from 1 to 8 repesent the visiting landmarks with visiting orders for the user 01 in Table 4.

For the user profiles 01–04 in Table 3, the users show strong positive preferences on landmarks of *Food and Drink* and *Shopping* with scores of 5 points and fair positive preferences on landmarks of *Nature* and *]Entertainment* with scores of 3 points. For the recommendation routes 01–04 in Table 4, only landmarks of *Shopping*, *Nature* and *Entertainment* are recommended and sequential trip routes are generated accordingly. It indicates that our algorithm infers and matches users' preferences very well. Likewise, for the user profiles 05–08 in Table 3, the users show strong positive preferences on landmarks of *History*, *Nature* and *Art* with scores of 4 points and fair positive preference on landmarks of *Entertainment* and *Food and Drink* with scores of 3 points. For the recommendation routes 05–08 in Table 4, only landmarks of *History*, *Nature*, *Art* and *Entertainment*are recommended and sequential trip routes are generated respectively. Also, in these cases, our proposed algorithm is superior in recommending visiting landmarks based on individual preferences. Generally speaking, experimental results demonstrate that the proposed algorithm successfully generates the trip routes with better favorable landmark recommendations for different users.

4. Conclusion

In this paper, we have proposed an algorithm that uses users' current location information and personal interest preferences as key criteria for nearby spot recommendations. Experimental results confirm that our algorithm can deal with different users' preferences.

As future works, the algorithm can be extended to better sub-route organizations and distances between landmarks.

References

[1] D. Gavalas, C. Konstantopoulos, K. Mastakas and G. Pantziou, "A survey on algorithmic approaches for solving tourist trip design problems," *Journal of Heuristics*, vol. 20, iss. 3, pp. 291–328, 2014.

[2] M. H. Park, J. H. Hong and S. B. Cho, "Restaurant recommendation for group of people in mobile environments using probabilistic multi-criteria decision making," in *Proc. 8th Asia-Pacific Conference on Computer-Human Interaction (APCHI '08)*, pp. 114–122, 2008.

[3] M. Ye, P. Yin and W. C. Lee, "Location recommendation for location-based social networks," in *Proc. 18th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '10)*, pp. 458–461, 2010.

[4] L. Y. Wei, Y. Zheng and W. C. Peng, "Constructing popular routes from uncertain trajectories," in *Proc. 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*, pp 195–203, 2012.

[5] Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser and M. Pazzani, "An energy-efficient mobile recommender system," in *Proc. 16th ACM. SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '10)*, pp. 899–908, 2010.

[6] L. D. Prete and L. Capra, "diffeRS: A mobile recommender service," in *Proc. 11th IEEE International Conference on Mobile Data Management (MDM '10)*, pp. 21-26, 2010

[7] Y. Shi, P. Serdyukov, A. Hanjalic and M. Larson, "Personalized landmark recommendation based on geotags from photo sharing sites," in *Proc. 5th AAAI Conference on Weblogs and Social Media (ICWSM '11)*, pp. 622–625, 2011.

[8] X. Lu, C. Wang, J. M. Yang, Y. Pang and L. Zhang, "Photo2trip: generating travel routes from geotagged photos for trip planning," in *Proc. 18th ACM international conference on Multimedia, MM '10*, pp. 143-152, 2011.

[9] S. Bao, T. Nitta, D. Shindou, M. Yanagisawa and N. Togawa "A landmark-based route recommendation algorithm for pedestrian walking strategies," in *Proc. 4th IEEE Global Conference on Consumer Electronics (GCCE '15)*, pp. 672–673, 2015.

[10] W. Z. Shi, M. Goodchild, B. Lees and Y. Leung, "Direction used based on gender," *Advances in Geo-Spatial Information Science*, 1st edition, Nottingham, UK, CRC Press, pp. 251–252, 2012.

Table 2: Landmark types and landmarks.

Landmark type	Landmarks
History	明治神宮, 新宿御苑, 皇居, 迎賓館
Nature	明治神宮, 新宿御苑, 皇居
Entertainment	Bunkamura, 東京ターワ, 東京都庁, 代々木競技場
Art	迎賓館, 日本近代文學館
Sport	代々木競技場
Food+Drink	伊勢丹, 小田急百貨店, 西武渋谷店
Shopping	伊勢丹, 小田急百貨店, 西武渋谷店, SHIBUYA109, エルメス
Night Life	Bunkamura

Table 3: User profile.

User ID	History	Nature	Entertainment	Art	Sport	Food+Drink	Shopping	Night Life	*top-k*
01	2	3	3	2	2	4	5	2	8
02	2	3	3	2	2	4	5	2	7
03	2	3	3	2	2	4	5	2	6
04	2	3	3	2	2	4	5	2	5
05	4	4	3	4	1	3	1	1	8
06	4	4	3	4	1	3	1	1	7
07	4	4	3	4	1	3	1	1	6
08	4	4	3	4	1	3	1	1	5

Table 4: Trip route results.

User ID	1st	2nd	3rd	4th	5th	6th	7th	8th
01	伊勢丹	新宿御苑	明治神宮	西武渋谷店	SHIBUYA109	エルメス	Bunkamura	小田急百貨店
02	伊勢丹	新宿御苑	明治神宮	西武渋谷店	SHIBUYA109	Bunkamura	小田急百貨店	
03	伊勢丹	新宿御苑	明治神宮	西武渋谷店	SHIBUYA109	小田急百貨店		
04	伊勢丹	新宿御苑	西武渋谷店	SHIBUYA109	小田急百貨店			
05	皇居	迎賓館	東京タワー	新宿御苑	代々木競技場	明治神宮	日本近代文学館	東京都庁
06	皇居	迎賓館	東京タワー	新宿御苑	代々木競技場	明治神宮	東京都庁	
07	皇居	迎賓館	新宿御苑	代々木競技場	明治神宮	東京都庁		
08	皇居	迎賓館	新宿御苑	代々木競技場	明治神宮			

Fig. 5: An example of the generated route for the user 01.