User Transition Pattern Analysis for Travel Route Recommendation

Junjie SUN†, Chenyi ZHUANG††, Nonmembers, and Qiang MA†, Member

SUMMARY A travel route recommendation service that recommends a sequence of points of interest for tourists traveling in an unfamiliar city is a very useful tool in the field of location-based social networks. Although there are many web services and mobile applications that can help tourists to plan their trips by providing information about sightseeing attractions, travel route recommendation services are still not widely applied. One reason could be that most of the previous studies that addressed this task were based on the orienteering problem model, which mainly focuses on the estimation of a user–location relation (for example, a user preference). This assumes that a user receives a reward by visiting a point of interest and the travel route is recommended by maximizing the total reward from visiting those locations. However, a location–location relation, which we introduce as a transition pattern in this paper, implies useful information such as visiting order and can help to improve the quality of travel route recommendations. To this end, we propose a travel route recommendation method by combining location and transition knowledge, which assigns rewards for both locations and transitions.

key words: travel route recommendation, sightseeing, location-based social network, matrix factorization

1. Introduction

In recent years, tourism has become one of the most important industries in the world. At the same time, benefiting from the rapid development of location-based social networks (LBSN), user-generated content (for example, photos, check-ins, and ratings) is increasingly available on the Internet. Various services exist to help tourists with a better tourism experience when they travel in an unfamiliar urban area (for example, point of interest (POI) recommendation). One of the important tasks is travel route recommendation, or tour planning, which aims to automatically recommend travel routes that meet users’ requirements; this is addressed in this paper.

Currently, most existing research on, and services for, travel route recommendation are based on the model of the orienteering problem (OP) and its variants [1]. OP is a classical route planning problem whose objective is to maximize the total score on a path passing through a subset of nodes without exceeding the time budget, on an undirected graph with weighted nodes and known travel time between these nodes [2]. For the travel route recommendation task, an urban area is a graph, a POI is a node in the graph, the time cost to travel from one POI to another is the weight of an edge, and users’ satisfaction is represented as the node weight (that is, tourists visit a POI to collect its score). The output is an optimal travel route that maximizes the total user satisfaction score while keeping the total travel time within the budget.

[3] described three key relations in LBSN studies: (1) user–user relations, (2) user–location relations, and (3) location–location relations. Conventional approaches mainly include node rewards (that is, user–location relations) in their objective functions, while the relations between locations are considered as the travel distance in constraints. In other words, their objective function assumes that tourists collect rewards after visiting locations and find the route with the maximal total reward. However, these methods suffer from several drawbacks, listed below:

- **Budget–reward trade-off.** Because the objective is to maximize the route’s reward, the recommended route will be a trade-off between the travel budget and location rewards, subject to several constraints. Taking the travel time budget constraint as an example, to maximize the route’s reward, conventional approaches will avoid locations requiring long travel duration and add locations with a high reward-time ratio. However, such locations may not be of high sightseeing quality (because locations with shorter travel duration have a higher reward–time ratio), so some landmarks with a long travel duration might be ignored during the tour planning.
- **Insufficient consideration of location–location relations.** Failing to consider location–location relations might reduce the recommendation quality. For instance, various influences (for example, public transportation, geographical constraints, and travel guides) may cause tourists to always visit some POIs in a certain order. Disregarding the visiting order and always recommending the shortest route may generate unreasonable routes. Figure 1 shows an example of different visiting orders. Each dimension of the figure represents locations in the Toronto dataset (introduced in Sect. 5.1) and each entry represents the normalized observed transition weight from one POI to another. We observe that transitions in the green rectangle are asymmetric, whereas they are symmetric in the red rectangle; the green rectangle demonstrates the unbalanced visiting orders between locations.
Recently, several studies have attempted to improve the quality of the travel route recommendation system by adjusting the OP-based model. For example, based on the OP, [4] studies the unsatisfactory recommendations caused by the scope of the POI. The authors illustrate that some POIs are large areas and not suitable for representation as a single POI, and propose a method to jointly consider outer and inner travel routes across different ranges of POIs.

To address the above drawbacks and improve the recommendation quality, in this paper, we propose a travel route recommendation method by introducing user transition patterns into the OP model. In general, a travel route consists of nodes (locations) and edges (transitions between locations). From the travel routes of many tourists, we can not only discover popular visiting locations but also transition patterns between locations. As tourists travel from one location to another in the city, they make different transitions with different frequencies (weights); we introduce these as transition patterns in this paper. An idea similar to this paper is considered in [5]. Both location and transition knowledge are jointly modeled to recommend a travel route. They take the travel route recommendation task as a likelihood maximization problem and propose a method to compute probabilities of points and transitions. However, the joint model has been evaluated only under the route length budget and does not show the advantages of introducing transition knowledge.

Transition patterns can be regarded as connections between locations, whereby the higher the weight of a transition, the tighter the relationship between the two locations. A transition pattern implies not only information such as visiting order but also possibly a scenic path between locations (for example, featured streets between POIs or beautiful paths in a large area). In this way, the location-location relations are emphasized.

A large amount of users’ transition data helps us to generate a knowledge graph. This represents the target travel city’s implicit structure information, such as visiting order and tourist flow trends, and helps to restrict the route to be matched with the real situation. For instance, Fig. 2 shows part of the visualized observed transition weight graph of Edinburgh in the dataset (introduced in Sect. 5.1). The thickness of an edge represents the weight of the transition (that is, how many tourists traveled through the edge in the dataset). Basically, POIs in the center of the city are tightly connected to each other and loosely connected to the outer city. In the yellow rectangle, there is a train station. This can be regarded as a type of geographical constraint because most transitions pass on both sides of the station but fewer pass by the station. This type of restriction should be considered in the travel route recommendation.

Therefore, we extract transition patterns from users’ travel route data and use them to compute rewards on edges. This changes the OP to a mixed orienteering problem (MOP) [6]: both nodes and edges are assigned rewards. We recommend travel routes by solving this MOP, aiming to maximize the total reward from both nodes and edges.

Our main contributions are summarized as follows:

- We propose a general framework for recommending travel routes under different travel budget constraints. The experimental results verified that our method, which considers both locations and transitions, can achieve better performance than recently proposed methods.
- We propose a novel method to learn transition patterns by incorporating them with spatiotemporal features. Experimental results reveal the effectiveness of the transitions learned by introducing both spatial and temporal influences.
- A parameter study is included, to understand the balance between location and transition in the tour planning procedure.
- A real tourism dataset of Kyoto is applied for real user evaluation. We compare the recommendation results with and without transition knowledge, which confirms our hypothesis that transition patterns are helpful for improving the quality of travel route recommendations.

2. Related Work

2.1 Tourist Trip Design Problem

The tourist trip design problem (TTDP) is widely studied in operational research, and OP is suitable as a theoretical model of it. OP-variant models that satisfy more realistic constraints have been proposed. These studies have focused on proposing suitable heuristic planning algorithms,
because it is an NP-hard task. For example, team OP (TOP) extends OP by allowing multiple tours, and is applied for multi-day tour planning (e.g., [7]). Unlike the TOP, a recent study [8] proposes a multi-day tour planning algorithm by maximizing the utility of the worst day.

The most studied extension of OP-based tour planning is TOP with time windows (TOPTW). This adds opening and closing times for each node (that is, POIs’ opening hours) to TOP, and was solved with an iterated local search (ILS) algorithm in [9]. Based on this extension, public transportation services were integrated into [10]. [11] is probably the most complete study of tour planning based on TOPTW; the authors jointly modeled various settings mentioned in previous studies (for example, public transportation service and opening hours). In [4], the authors defined super-POI for the large sightseeing areas, which contain smaller sightseeing spots, and separately plan the outer and inner routes.

However, none of these works has considered modeling users’ travel behavior or directly planning the travel route with predefined values (for example, POI reward).

2.2 Tour Recommendation

There are many tour recommendation methods that have been proposed, based on the OP model, to provide a better recommendation quality. These are more concerned with how best to discover and assign reward values for POIs in the OP model. For instance, personalization has been addressed by considering users’ travel behavior and assigning more weight to users’ preferred POIs.

One of the earliest OP-based tour recommendation methods, proposed in [12], discovered POIs by mining from a large-scale dataset of geotagged social images. It has been extended by Lim et al. [13], to mine social geotagged images’ metadata and estimate users’ preferences with time-based interests. A similar idea was considered in [14], which estimates users’ preferences by the categories of the POIs. With the development of computer vision technology, an image feature-enhanced tour recommendation method was proposed in [15], to find POIs that are visually similar to users’ uploaded images. Preferences can be manually selected in [16], [17]; the authors developed an interactive web system in which users can manually decide travel preferences. A feature-centric matrix factorization model was applied in [18]. With a user–feature matrix, preferences that out of the residential areas also can be evaluated through the collaborative filtering model. [5] is the study most related to our work: it recommends travel routes based on both POIs and transition probabilities, within a travel length budget.

There have also been several studies not based on the OP model. For instance, [19] models the tour planning task as a max-cover problem that finds the most suitable travel routes from other tourists’ travel history, according to keyword matching.

However, the literature reviewed above rarely considered location–location relations, and recommended travel routes based only on user–location matching. In our work, we study location–location relations by considering transition patterns to improve the existing tour recommendation method. Different from our previous work [20], we modified the transition pattern inference method and added more experiments and evaluations.

3. Preliminaries

3.1 Problem Definition

We first define two concepts: travel route and user query, and then define our travel route recommendation task.

Definition 1 (travel route): A travel route is a sequence of POIs $(p_1, p_2, \ldots, p_N)$. Each point $p_i$ is a location that the tourist has visited, and consists of [route identifier, POI identifier, category, date and time, longitude, and latitude].

Definition 2 (user query): A user query is a query $q = (p_s, p_e, B)$ in which $p_s$ and $p_e$ represent the start and end point, respectively, and $B$ represents the travel budget. We use $Q$ to denote all of the queries corresponding to the travel route data. In our work, we consider two types of travel budgets: the route length budget $B_l$ and the time budget $B_t$.

With these two concepts, our problem is defined as follows:

Problem 1: Assuming there are $N$ POIs, $P = \{p_1, p_2, \ldots, p_N\}$, and tourist travel route data for the target city, given a user query, the framework will output a list of POIs of $P$ as a recommended travel route.

3.2 Objective Function

According to the problem definition, we assume that a user can receive a reward by visiting a POI or traveling between POIs. The reward here represents a user’s satisfaction with this tour. Therefore, we recommend a travel route by maximizing the total reward of the tour, which includes the rewards on both locations and transitions. Specifically, given $N$ POIs, $P$, and tourists’ travel route data in the travel city, a travel route is recommended according to the user query by solving the following objective function [13]:

$$
\max \sum_{i=1}^{N-1} \sum_{j=2}^{N} x_{ij} R(p_j|p_i)
$$

(1)

Subject to:

$$
\sum_{j=2}^{N} x_{1j} = \sum_{i=1}^{N-1} x_{Ni} = 1, \sum_{i=2}^{N-1} x_{i1} = \sum_{j=1}^{N-1} x_{Nj} = 0
$$

(2)

$$
\sum_{i=1}^{N-1} x_{ik} = \sum_{j=2}^{N} x_{kj} \leq 1, \forall k = 2, \cdots, N - 1
$$

(3)

$$
\sum_{i=1}^{N-1} \sum_{j=2}^{N} x_{ij} = B_l - 1
$$

(4)

$$
p_i - p_j + 1 \leq (N - 1)(1 - x_{ij}), \forall i, j = 2, \cdots, N
$$

(5)
where $R(p_j|p_i)$ is the reward function for visiting $p_j$ from $p_i$, which will be defined in the following sections; $N$ is the available number of POIs in the travel city; $x_{ij}$ is a binary indicator that equals 1 when a user travels from POI $p_i$ to $p_j$ and equals 0 otherwise. Here we mark $p_1$ as the start location and $p_N$ as the end location. Constraint 2 specifies that a route must begin at the start location and finish at the end location. It also prevents a travel route from revisiting the start location or traveling from the end location. Constraint 3 specifies that each POI can only be visited once. Constraint 4 is the travel budget constraint, which restricts the recommended travel route to be no longer than $B_t$ (i.e., the number of POIs that the tourist wants to visit). Constraint 5 specifies that subtours are to be avoided; this constraint was proposed in the classical traveling salesmen problem [21].

In addition, we consider another budget constraint—the travel time budget—which limits the recommended route’s total travel time to be no greater than $B_t$. The following function defines the time budget constraint and replaces constraint 4:

$$\sum_{i=1}^{N-1} \sum_{j=2}^{N} \text{Cost}(p_i, p_j)x_{ij} \leq B_t,$$

where $\text{Cost}(p_i, p_j)$ includes the travel time from $p_i$ to $p_j$ and the visiting duration time at $p_j$; $B_t$ represents the total travel time budget.

The objective function is similar to other OP-based methods that aim to maximize tourists’ visiting rewards. The key point is how to correctly define the reward function $R(p_j|p_i)$. We introduce different methods to assign rewards to locations and transitions; the reward function $R(p_j|p_i)$ is then defined as a combination of these two types of rewards.

4. **Reward Functions**

4.1 **Location Reward**

Location reward represents how satisfied the tourist is after visiting the location. Most of the related literature focuses on this, by considering the user–location relation. Because it has been widely studied, we directly apply existing work to assign rewards to locations. Specifically, we adopt two different methods, to adapt to different travel budget constraints and to test the generality of our proposed framework.

**POI ranking.** A naive idea to assign a reward to a location is by using POI popularity, which assumes that the more tourists visit a location, the more attractive it is. As an alternative, machine learning methods allow us to take advantage of more features than POI popularity, to train a model with other users’ travel route data. For instance, the rank support vector machine (RankSVM) is applied in [5], by introducing more features (e.g., category, popularity, and average visit duration) to rank POIs according to the user query under the travel length budget constraint. Then the POI ranking scores can be obtained as POI reward by learned the ranking of POIs using rankSVM, with a linear kernel and L2 loss [22]. The objective function is:

$$\min_w \frac{1}{2}w^Tw + C \sum_{p,p'=P, q\in Q} \max(0, 1 - w^T(\phi_{p,q} - \phi_{p',q}))^2,$$

where $w$ is the parameter vector, $C > 0$ is a regularization constant, $P$ is the set of POIs to rank, and $Q$ represents all of the queries with respect to the travel routes in the training set; $\phi$ is the feature vector for a POI $p$ with respect to a query $q$.

Given a query $q$, the ranking score of POI $p_x$ is then computed as:

$$R_{p_x|q} = w^T\phi_{p_x,q}$$

All ranking scores are scaled to the range [0, 1] with the following softmax function:

$$R_p(p_j|p_i) = \frac{\exp(R_{p_j|q})}{\sum_p \exp(R_{p_j|q})},$$

where $R_{p_j|q}$ and $R_{p_i|q}$ are ranking scores computed with Eq. (8).

**Time-based User Interest.** In addition, we consider the factor of personalization: that tourists may be more interested in their preferred locations. Because we cannot manually decide user preference for queries in $Q$, we adopt time-based user interest [13] to automatically estimate personalized preference, in different POI categories. This is based on the heuristic idea that tourists would stay longer when visiting the category of POIs they like. Time-based user interest is denoted as $\text{IntTime}(\text{Cat}_{p_j})$ which can be computed by comparing each user’s duration on certain POI category to the average visit duration of all users.

With the computed user interest, the personalized location reward can be decided as follows:

$$R_p(p_j|p_i) = \eta\text{IntTime}(\text{Cat}_{p_j}) + (1 - \eta)\text{Pop}(p_j),$$

where $\eta$ is the trade-off parameter that decides the weight between personalized interest $\text{IntTime}(\text{Cat}_{p_j})$ and POI’s popularity $\text{Pop}(p_j)$. Specifically, we set $\eta = 0.5$ as described in [13]. We then normalize each POI reward value by the maximum value, to scale all values to [0, 1].

4.2 **Transition Reward**

To address the drawbacks of OP-based methods and improve the quality of the recommended travel route, we would like to take location–location relations into consideration. Specifically, transition patterns can be regarded as the weights of going from one POI to another, which contains certain features such as visiting order. We extract transition patterns from tourist travel route data and assign a reward to each edge between two POIs.

One naive idea to represent transition pattern is to directly normalize the observed transitions. Then the transition patterns between POIs could be regarded as the rewards
of transitions. However, this idea has several drawbacks:

- The transition data that we can observe are not complete; that is, some locations that have transitions may not be observed in our datasets.
- Such a model cannot deal with new POIs; that is, there are no transitions that can be observed when new POIs are discovered and added to the dataset.

Therefore, given the travel route of all tourists, we extract the number of transitions (user transits from one location to another) for each pair of POIs. Then an observed transition matrix $T' \in \mathbb{R}^{N \times N}$ can be constructed, where $N$ is the number of POIs in the target travel city and each entry in $T'$ denotes the transition frequency times between POIs.

Because some of the transitions are not observed, we aim to infer the completed transition matrix $\hat{T}$ by exploiting observed transitions to predict transition values to unobserved entries. Specifically, we divide each entry by the maximum entry value to normalize $\hat{T}$ and assign transition reward as below:

$$R(p_j|p_i) = \frac{\hat{T}_{i,j}}{\text{max}(\hat{T})} \quad (11)$$

In the following subsections, we propose a spatiotemporal feature enhanced matrix factorization model to assign transition rewards.

**Weighted Transition Matrix Factorization.** Similar to the idea of collaborative filtering, POIs may have common attractions or connections to other POIs, owing to common features that match very well. For instance, a popular landmark has a stronger connection to other locations than some unpopular POIs because many tourists are observed that travel to or from it. A common technology for collaborative filtering in the field of recommender system research [23] is matrix factorization, which factorizes the interaction matrix into two lower-dimensional latent feature matrices. An unobserved interaction entry can then be predicted through the inner product of two latent feature matrices. Therefore, we can infer unobserved transition weights by exploiting matrix factorization technology.

Unlike related work that applied matrix factorization for user–item or user–location interaction pairs [23], [24], we want to use it to study location–location relations (that is, both decomposed lower latent features are about locations). We first construct the observed transition matrix $T'$ by extracting transitions from tourist travel route data. We then divide each entry by the maximum entry value to normalize $T'$ and obtain a matrix $\hat{T}$, in which the value of each entry is in $[0, 1]$ and represents the relative weight of each transition.

One naive solution would be to directly factorize the weighted transition matrix. However, because the features of the two dimensions in the matrix are the same (both about locations), we add an interaction latent component $M$ to represent local interaction models of the decomposed latent variables [25]. Hence, one reasonable solution for inferring the weighted transition matrix from observed data is to factorize the observed weighted transition matrix $T$ as follows:

$$T \approx V_sMV_t^T, \quad (12)$$

where $V_s \in \mathbb{R}^{N \times k}$ and $V_t \in \mathbb{R}^{N \times k}$ represent latent features of source and destination points, respectively; $M \in \mathbb{R}^{k \times k}$ is the interaction matrix that represents the relationship between locations; and $k$ is the number of latent features. The latent matrices $V_s$, $V_t$, and $M$ are then computed by solving the following optimization function:

$$\min_{V_s, V_t, M} \left\| I \odot (T - V_sMV_t^T) \right\|_2^2 + \lambda_I \left[ \|V_s\|^2 + \|M\|^2 + \|V_t\|^2 \right], \quad (13)$$

where $\|\cdot\|_2$ denotes the Frobenius norm; $I$ is a binary matrix each entry of which, $i_{i,j}$, indicates whether a transition has been observed; and $\lambda_I$ is the parameter for the regularization term.

With learned latent factors $V_s$, $V_t$, and $M$, the unobserved transition entries are filled with predicted values and an inferred weighted transition matrix $\hat{T}$ can be computed through Eq. (12). Finally, we assign the transition reward $R_T(p_j|p_i)$ through Eq. (11).

**Spatial Influence.** Although we can infer a complete weighted transition matrix by directly factorizing Eq. (13), there is still scope for improving the performance. For instance, spatial influence is an important factor in the relation of a pair of POIs. [24], [26] introduced spatial influence as the physical distance between user and location or between user and user. In contrast, we consider spatial influence as a feature of the relation between locations, which can be applied to enhance our weighted transition matrix factorization model.

Figure 3 shows the cumulative distribution function of transition distance on two different datasets: Glasgow and Toronto. We can see that both curves rise dramatically when the distance is small: most tourists travel at most 2 km from one POI to the next. Tourists tend to visit nearby POIs; in other words, closer POIs have a stronger influence. Therefore, we want to extract location–location spatial features by considering the distance between POIs.

First, we calculate the distance between POIs using the haversine formula [27] to compute the distance between two geographic coordinates. Second, we take the reciprocal of the value in each entry to capture the POI distance reciprocal matrix, in which the longer the distance, the smaller the value. Finally, we normalize each entry by dividing by the
maximum value in the matrix, to construct the POI spatial influence matrix \( G \in \mathbb{R}^{N \times N} \). The spatial influence matrix can be regarded as additional global knowledge under the assumption that the closer pairs of POIs are more likely to be visited together. In other words, each entry represents the confidence of location–location spatial influence, which is larger when the distance is smaller.

We decompose \( G \) to get the lower latent spatial features of location \( V_g \). Unlike Eq. (12), in which source and destination represent different latent variables, the distance between two locations is the same in both directions, so we use the same latent variable \( V_g \in \mathbb{R}^{N \times k} \) to represent the spatial feature of location. Therefore, the spatial influence matrix can be factorized as follows:

\[
G \approx V_g V_g^T,
\]

We can obtain the latent spatial feature \( V_g \) by solving the objective function as follows (also known as nonnegative matrix factorization):

\[
\min_{V_g} \|G - V_g V_g^T\|^2 + \lambda_r \|V_g\|^2,
\]

where \( \lambda_r \) is the parameter for the regularization term.

Temporal Influence. Temporal influence is another important factor for LBSN studies. For example, users’ check-in time flow was considered in [28], which models temporal influence as category preference over time (for example, visiting the office in the morning or the gym in the evening). However, because our target is to enhance the representative ability of location–location relations, we focus more on the temporal influence of POIs. Similar to spatial influence, we aim to obtain location–location temporal features to enhance the latent features \( V_g \) and \( V_g \).

There are many temporal factors related to POIs, users, or travel routes. We consider several of them that are helpful to obtain location–location temporal features. In terms of spatial influence, we assume that a pair of POIs with smaller distance has a larger transition weight. However, this might not work well for all POIs. For instance, users may prefer to go to a restaurant at noon and evening. Suppose a user visits a POI near a restaurant in the morning. From the viewpoint of spatial influence, the restaurant could be visited next because of its higher transition weight. However, from the viewpoint of temporal influence, the restaurant would not be selected because the time is inappropriate to visit. Therefore, by investigating tourist travel route data, we have two observations:

- Each POI has an available open time window and visits vary over 24 hours. As shown in Fig. 4(a), the peak times for visiting are in the morning and afternoon, whereas the peak in Fig. 4(b) is around noon.

- For every travel route of users, Fig. 5 shows the distribution of intervals between users’ visiting times for two POIs. These indicate how long tourists prefer to travel from one POI to the next: most of the transits are under two hours.

The different visiting times in Fig. 4 may be caused by different POI categories and users’ transitions between POIs. We can make the simple assumption that the transition weight between two POIs is greater when the difference between their visiting time peaks is more suited to user transit behavior (that is, smaller than a certain value of visiting time interval). We aim to extract latent spatial features of POIs, which helps to restrict the influence weight upon the spatial influence features.

With the above observations and assumptions, we use a Poisson distribution to fit users’ visiting time intervals and a Gaussian mixture model (GMM) to fit the check-in time data of each POI. We automatically fit the data and set the GMM component range from 1 to 3, which corresponds to morning, afternoon, and night. We then construct the temporal influence matrix \( E \in \mathbb{R}^{N \times N} \) by checking the difference in the means of visiting time distribution between two POIs and taking this value into the fitted visiting time interval distribution to fill the entry value of \( E \). \( E \) is equal to the normalized \( E \), which represents the temporal influence between POIs. We represent this temporal influence matrix \( E \) as below:

\[
E \approx V_e V_e^T,
\]

where \( V_e \) represents the latent temporal features of POIs. To reduce the computational complexity, we assume that the latent temporal features for source and destination are the same. Similar to the spatial influence matrix \( G \), we obtain the latent spatial feature \( V_g \) by solving the following objective function:

\[
\min_{V_e} \|E - V_e V_e^T\|^2 + \lambda_r \|V_e\|^2,
\]
where $\lambda_\ell$ is the parameter for the regularization term.

Having obtained the latent spatial and temporal features, we face the question of how to introduce them into the weighted transition matrix factorization model to enhance the inference ability. By considering the above observations of user transition behaviors, we want the latent location features to be more similar to the latent spatial and temporal features. Figure 6, which visualizes the correlation between the latent location and spatial features, supports this viewpoint. For both Figs. 6(a) and 6(b), each dimension represents locations in the Edinburgh dataset.

Figure 6(a) shows the observed weighted transition matrix while Fig. 6(b) shows the spatial influence matrix (the distance reciprocal matrix). The larger the value, the darker the color. The visualization shows that they are quite similar to each other, which verifies the close relationship between the latent location features and latent spatial features.

Therefore, we introduce the spatial and temporal influences to enhance Eq.(12) by minimizing the difference between the latent location and spatiotemporal features. In other words, we not only require the latent location features $V_s$ and $V_t$ to produce less error, but also want them to be as similar as possible to $V_g$ and $V_e$. Specifically, we use the following objective function to factorize the weighted transition matrix:

$$
\min_{V_s, V_t, M} \left[ \| I \odot (T - V_s M V_t^T) \|^2 + \lambda_g \| V_s - V_g \|^2 + \| V_t - V_e \|^2 + \lambda_e \| M \|^2 \right]^2 \\
+ \lambda_\ell \left[ \| V_s - V_g \|^2 + \| V_t - V_e \|^2 \right] + \lambda_\ell \| M \|^2,
$$

(18)

where $\lambda_\ell$ is the parameter for the regularization term, and $\lambda_g$ and $\lambda_e$ control the weights of the spatial and temporal influence, respectively. For instance, with a large value on $\lambda_g$, the latent location features are more similar to the latent spatial features $V_g$.

Finally, the inferred weighted transition matrix can be computed with Eq.(12) and the transition reward $R_T(p_j|p_i)$ assigned through Eq.(11).

4.3 Combining Location and Transition Reward

After assigning rewards to locations and transitions, we now aim to combine knowledge of locations and transitions to recommend travel routes. Both location and transition rewards are already normalized and each value is in $[0, 1]$. Therefore, we can combine the location and transition rewards with the following equation:

$$
R(p_j|p_i) = \alpha R_p(p_j) + (1 - \alpha) R_T(p_j|p_i),
$$

(19)

where $R(p_j|p_i)$ is the reward function defined in the objective function Eq.(1); $R_p(p_j)$ and $R_T(p_j|p_i)$ are the location and transition reward functions defined in Sects. 4.1 and 4.2, respectively; $\alpha \in (0, 1)$ is a trade-off parameter that indicates the relative importance of location and transition rewards.

We briefly review our definition of reward function, which considers location, transition, and their combination. If we use location reward, the objective function Eq.(1) is similar to the original OP [2], which only includes reward on nodes. If we use transition reward, it is similar to the arc orienteering problem (AOP) [29], which only includes reward on edges. In terms of the combination equation Eq.(19), the reward function $R(p_j|p_i)$ now includes both nodes and edges of the POI graph, which leads the objective function Eq.(1) to be a mixed orienteering problem (MOP) [6]. Although this is quite different from the original OP, it can still be treated as a mixed integer linear problem, which can be solved by optimization tools such as Gurobi† or lp_solve††.

4.4 Optimization and Latent Variable Learning

We solve our travel route recommendation objective function Eq.(1)–(5) in Sect.3.2 with the Gurobi optimization package. For the training strategy, we minimize the objective functions Eq.(13), (15), (17), and (18) with a gradient descent approach by iteratively optimizing the latent variables $V_s$, $V_t$, $M$, $V_g$, and $V_e$; this is supported by the Theano framework.

5. Experiments

5.1 Experimental Setup

To evaluate our proposed method, both quantitative and qualitative experiments were conducted.

Datasets. To evaluate the recommendation performance of our proposed framework, a public LBSN dataset was used [5], [13]. This dataset contains travel route data extracted from Flickr photos in five cities: Budapest, Edinburgh, Glasgow, Toronto, and Vienna. The detailed statistics of the dataset are shown in Table 1. In our travel route recommendation experiments, we randomly divided the data of each city into five folds, and performed a five-fold cross-validation [30] to evaluate different approaches. This means that, when testing on one fold of the dataset, we used the other folds of data to train our models.

To evaluate our proposed method with real tourists, we
adopted a real tourism dataset of Kyoto. This dataset originally contained GPS trail data for 450 foreign tourists and 406 students on school excursion one-day tours. It records the GPS signal every two seconds for each tourist, so a travel route can be formed by linking all these GPS points.

According to the GPS trails, we first mapped all GPS points to 123 POIs that represent sightseeing attractions. We used these data to recommend travel route for real tourists.

**Queries.** To evaluate our proposed framework’s generality, we considered two types of common travel budget (route length budget and travel time budget), which were introduced for tour recommendation in [5], [13]. For simplicity, we use a traveling speed of 4 km/h (a leisurely walking speed) as a tourist’s average speed when traveling between POIs under the travel time budget; this has also been applied for tour recommendation by [13], [15].

For both travel budget constraints, we use queries extracted from tourists’ real travel route sequences. Travel route sequences shorter than three POIs are ignored in the evaluation because they cannot be presented as valid queries. For instance, suppose a tourist started from $p_3$ and traveled to $p_6$, visiting three POIs and spending four hours on this trip. Then a query can be constructed as $q(p_3, p_6, 5)$ and $q(p_3, p_6, 4)$, for route length budget and travel time budget, respectively.

For real tourist evaluation, two queries are tested:

- $q_1 = (\text{KyotoNationalMuseum}, \text{Sanjo}, 4h)$
- $q_2 = (\text{KyotoUniversity}, \text{Sanjo}, 7h)$

These two queries consider different start locations and travel time budgets, finally returning to the hotel near Sanjo. We take the average visiting duration as the travel time for each POI and the travel speed between POIs is set to an average speed of 12 km/h for simplicity (this can be easily replaced by other transportation data: for example, transport time computed by Google maps).

5.1.1 Comparison Methods

To understand the effectiveness of our proposed framework, we compared it with a list of approaches. Under the route length budget, the following approaches were tested:

- **PoIPop.** A baseline approach that recommends travel route based only on POI popularity.
- **PoIRank.** A machine learning method proposed in [5]. By training a RankSVM model with POI features under the travel length budget, it recommends a travel route based on the POI ranking score.
- **Markov.** An explicit feature factorization method which factorizes a pair of POIs by different types of POI features to compute the transition probabilities between POIs. It is similar to our defined transition reward except that the sum of the outer links is equal to 1.

- **Rank + Markov.** A combination method, proposed in [5], that jointly optimizes point preferences and transition probabilities.
- **Tmf, Gtmf, Tgtmf.** Our proposed methods, introduced in Sect. 4.2, that recommend travel routes based only on transition reward. Tmf directly factorizes the weighted transition matrix. Spatial influence features are introduced in Gtmf. Spatial and temporal influences are jointly modeled in Tgtmf.
- **Rank + Tmf, Rank + Gtmf, Rank + Tgtmf.** Our proposed combination methods in Sect. 4.3 to recommend travel routes based on both location and transition rewards. Location reward is assigned with PoIRank and transition reward is assigned with our proposed Tmf, Gtmf, and Tgtmf, respectively.

Under the travel time budget, we combine PersTour in Sect. 4.1 with our proposed Tmf, Gtmf, and Tgtmf, described above, to evaluate the performance of travel route recommendation under the travel time budget.

- **PersTour.** Proposed in [13], which recommends travel routes by estimating personalized time-based user interest from users’ travel history data.
- **PersTour + Tmf, PersTour + Gtmf, PersTour + Tgtmf.** Our proposed combination methods that jointly consider location and transition rewards. PersTour is applied to assign location reward and transition reward is given by Tmf, Gtmf, and Tgtmf, respectively.

5.2 Evaluation Metrics

Various quantitative evaluation metrics, introduced in related work for travel route recommendation, are applied to our evaluation.

**Evaluation on Travel Route Recommendation.** One common evaluation in recommender system research is to compare the recommended result with the user’s selection in real life. Therefore, we evaluate all of the methods by comparing the recommended travel route with the user’s actual travel route in query set $Q$. The performance of all approaches is evaluated by using the following metrics.

- **Tour F1 score: $F_1$.** Because the $F_1$ score is a common metric for evaluating POI and travel route recommendations ([5], [13], [31]), we use the tour $F_1$ score as an evaluation metric for the recommended travel route. The tour $F_1$ score is the harmonic mean of the tour recall $R$ (how many of the user’s real visited POIs are recommended) and tour precision $P$ (how many recommended POIs are in the user’s real travel route). It is applied to a recommended travel route excluding the
Table 2 Performance comparison of travel route recommendation under the route length budget in terms of tour $F_1$ score ($T_{F_1}$).

<table>
<thead>
<tr>
<th></th>
<th>Budapest</th>
<th>Edinburgh</th>
<th>Glasgow</th>
<th>Toronto</th>
<th>Vienna</th>
</tr>
</thead>
<tbody>
<tr>
<td>PoiPop</td>
<td>0.216±0.277</td>
<td>0.331±0.378</td>
<td>0.346±0.429</td>
<td>0.200±0.320</td>
<td>0.316±0.342</td>
</tr>
<tr>
<td>PoiRank</td>
<td>0.297±0.331</td>
<td>0.314±0.365</td>
<td>0.241±0.361</td>
<td>0.372±0.428</td>
<td>0.309±0.340</td>
</tr>
<tr>
<td>Markov</td>
<td>0.253±0.323</td>
<td>0.237±0.323</td>
<td>0.335±0.430</td>
<td>0.291±0.405</td>
<td>0.200±0.287</td>
</tr>
<tr>
<td>TMF</td>
<td>0.170±0.283</td>
<td>0.311±0.366</td>
<td>0.223±0.361</td>
<td>0.327±0.411</td>
<td>0.199±0.293</td>
</tr>
<tr>
<td>GTMF</td>
<td>0.262±0.332</td>
<td>0.367±0.391</td>
<td>0.327±0.431</td>
<td>0.346±0.420</td>
<td>0.317±0.345</td>
</tr>
<tr>
<td>TGTMF</td>
<td>0.280±0.341</td>
<td>0.383±0.395</td>
<td>0.416±0.454</td>
<td>0.384±0.430</td>
<td>0.316±0.355</td>
</tr>
<tr>
<td>Rank+Markov</td>
<td>0.299±0.335</td>
<td>0.333±0.373</td>
<td>0.388±0.438</td>
<td>0.365±0.423</td>
<td>0.281±0.332</td>
</tr>
<tr>
<td>Rank+TMF</td>
<td>0.289±0.333</td>
<td>0.323±0.374</td>
<td>0.287±0.397</td>
<td>0.376±0.428</td>
<td>0.270±0.332</td>
</tr>
<tr>
<td>Rank+GTMF</td>
<td>0.326±0.251</td>
<td>0.368±0.391</td>
<td>0.368±0.439</td>
<td>0.385±0.423</td>
<td>0.332±0.350</td>
</tr>
<tr>
<td>Rank+TGTMF</td>
<td>0.337±0.355</td>
<td>0.383±0.395</td>
<td>0.452±0.458</td>
<td>0.402±0.432</td>
<td>0.356±0.364</td>
</tr>
</tbody>
</table>

start and end POIs. It is defined as follows:

$$T_{F_1} = \frac{2 \times T_P \times T_R}{T_P + T_R}$$  \hspace{1cm} (20)

- **Tour pairs-$F_1$ score**: $pairs-F_1$. The tour $F_1$ score only considers individual POIs in the travel route, while the visiting order of a pair of POIs is ignored. We apply the tour pairs-$F_1$ score, which was introduced in [5], to evaluate both POI identity and visiting order at the same time. The start and end POIs are included because they contain pairwise information. It is defined as:

$$pairs-F_1 = \frac{2 \times P_{Pair} \times R_{Pair}}{P_{Pair} + R_{Pair}},$$  \hspace{1cm} (21)

where $P_{Pair}$ and $R_{Pair}$ are the precision and recall, respectively, of ordered POI pairs compared to the ground truth. Tour pairs-$F_1$ takes a value between 0 and 1 and will achieve a score of 1 if and only if the POIs and the visiting order are exactly the same as the user’s real travel route.

**Evaluation of Transition Weight Inference.** To evaluate the performance of our proposed transition weight inference approaches, we normalize and randomly select 30% of the POIs from the observed matrix as the validation set, and then use different approaches to infer the unobserved transition weights (that is, new POIs are added to the dataset and no transition data can be observed). We adopt the following metric to evaluate the performance of the transition weight inference:

- **Root-Mean-Square Error (RMSE) of Transition Weight**: RMSE is a frequently used metric to measure the difference between a predicted value and the value actually observed. Let $T_{inf}$ represent the inferred transition weight and $T_{true}$ be the transition weight of the real data. For each entry value $u$, taken from the set of all inferred entries $U$, we compute the RMSE of transition weight inference as:

$$R M S E = \sqrt{\frac{\sum_{i \in U} (T_{inf} - T_{true})^2}{|U|}}.$$  \hspace{1cm} (22)

**Questionnaire for Real Tourists.** For each user query, we recommend two travel routes, generated by two different approaches. One is to recommend travel routes based on POI popularity, which can be regarded as the cold start scenario of PerksTour. The other is our proposed combination method, in which we infer the transition reward by employing all tourists’ travel route data and apply them on the POI popularity baseline. We set the trade-off parameter $\alpha = 0.7$; the reason for this setting will be explained below.

Besides the recommended travel routes, a questionnaire was provided and we ask for feedback of each recommendation result. It requires users to give a rating score, ranging from 1 to 10, for each recommended route and select one with more reference value for their tour.

5.3 Results

For the quantitative evaluation results, the travel route recommendation performance of various approaches with a route length constraint is summarized in Tables 2 and 3. Each table is divided into three parts: representing location-based, transition-based, and combination methods. Performance with a travel time constraint is summarized in Tables 4 and 5. Performance on the two types of constraints is evaluated using the tour $F_1$-score and the tour pairs-$F_1$-score measurements. The best method for each dataset (city) is shown in bold; the second best is shown in italic. The performance of transition weight inference in terms of RMSE is shown in Table 6; the best result method for each dataset is shown in bold. Finally, for real user evaluation, the rating scores are summarized in Fig. 9.

**Comparison under the Route Length Budget.** Under the route length constraint, as shown in Tables 2 and 3, our proposed combination method Rank+TGTMF outperforms all other baseline methods on different datasets in terms of both the tour $F_1$ and tour pairs-$F_1$ scores. This result verifies that our proposed framework is efficient by considering both location and transition rewards, and transition knowledge is helpful to improve the performance of the travel route recommendation.

Although Poirank uses machine learning method RankSVM with more features, it is not always better than baseline Poipop, which indicates that POI popularity is more efficient than other features. For instance, most tourists will not miss visiting famous landmarks of the target travel city.
Among all of the transition-based approaches, Tmf, directly factorizing the weighted transition matrix, had no effect on the explicit feature pair factorization method Markov, which indicates that the latent location features might include those evident feature pairs. TGTmf achieves the best result by using global extra spatial and temporal features. Specifically, we observe a significant improvement in performance over Tmf when incrementally introducing additional spatial and temporal influences.

**Comparison under the Travel Time Budget.** Performance under the travel time constraint is shown in Tables 4 and 5. PersTour+TGTmf, which combines spatial and temporal feature enhanced transition knowledge with personalized interest, achieves the best result. Again, we observe that performance is improved when additional spatial and temporal influences are incrementally introduced. In particular, the performance of Tmf noticeably improved when spatial influence features were introduced. This indicates that the location–location relation relies heavily on the distance between locations in the tour planning task.

Both our proposed PersTour+GTmf and PersTour+TGTmf consistently outperform PersTour, which only uses time-based user interest location rewards, in terms of the F1 and pairs-F1 scores. This verifies our idea of introducing transition knowledge to recommend travel routes.

**Comparison with Transition Weight Inference.** We use the RMSE of transition weights to evaluate different transition weight inference approaches, applied to all five datasets; the results are shown in Table 6. Similar to the results on the performance of the travel route recommendation introduced above, the naive direct factorization method Tmf is not efficient enough when compared with the explicit feature pair factorization method Markov. However, the performance notably improves when the additional spatial and temporal features are introduced. In particular, spatial influence is the most effective factor in the factorization and transition weight inference, which is consistent with the results on the performance of recommendation.

For reference, Fig. 8 shows a visualization of transition weight inference from applying our proposed method to the tourism dataset of Kyoto. Each red point represents a POI and each black edge represents a transition. The thicker a black edge, the more users travel between the two POIs. The visualization of the observed transitions is shown in Fig. 8 (a), from which we can easily find the location of several landmarks because they have more links to other POIs. For instance, Fushimi Inari shrine, located in the lower right corner, is a famous sightseeing attraction in Kyoto. Figure 8 (b) shows the inferred transition weight graph, which

---

**Table 3** Performance comparison of travel route recommendation under the route length budget in terms of tour pairs-F1 score ($pairs$-$F_1$).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Budapest</th>
<th>Edinburgh</th>
<th>Glasgow</th>
<th>Toronto</th>
<th>Vienna</th>
</tr>
</thead>
<tbody>
<tr>
<td>PoPoP</td>
<td>0.315±0.169</td>
<td>0.436±0.257</td>
<td>0.506±0.296</td>
<td>0.386±0.201</td>
<td>0.400±0.222</td>
</tr>
<tr>
<td>PoRANK</td>
<td>0.372±0.226</td>
<td>0.424±0.248</td>
<td>0.432±0.241</td>
<td>0.508±0.295</td>
<td>0.396±0.209</td>
</tr>
<tr>
<td>Markov</td>
<td>0.348±0.235</td>
<td>0.369±0.215</td>
<td>0.501±0.300</td>
<td>0.454±0.277</td>
<td>0.321±0.181</td>
</tr>
<tr>
<td>Tmf</td>
<td>0.292±0.204</td>
<td>0.423±0.246</td>
<td>0.418±0.253</td>
<td>0.480±0.281</td>
<td>0.320±0.205</td>
</tr>
<tr>
<td>Gtmf</td>
<td>0.357±0.247</td>
<td>0.465±0.275</td>
<td>0.494±0.304</td>
<td>0.494±0.282</td>
<td>0.403±0.228</td>
</tr>
<tr>
<td>TGTmf</td>
<td>0.367±0.253</td>
<td>0.475±0.282</td>
<td>0.555±0.322</td>
<td>0.521±0.294</td>
<td>0.407±0.246</td>
</tr>
<tr>
<td>Rank+Markov</td>
<td>0.379±0.233</td>
<td>0.437±0.259</td>
<td>0.534±0.301</td>
<td>0.502±0.289</td>
<td>0.380±0.206</td>
</tr>
<tr>
<td>Rank+Tmf</td>
<td>0.370±0.253</td>
<td>0.429±0.254</td>
<td>0.464±0.280</td>
<td>0.514±0.292</td>
<td>0.370±0.208</td>
</tr>
<tr>
<td>Rank+GTMF</td>
<td>0.401±0.256</td>
<td>0.466±0.276</td>
<td>0.521±0.309</td>
<td>0.520±0.290</td>
<td>0.415±0.228</td>
</tr>
<tr>
<td>Rank+TGTmf</td>
<td>0.411±0.262</td>
<td>0.475±0.280</td>
<td>0.580±0.327</td>
<td>0.533±0.297</td>
<td>0.435±0.247</td>
</tr>
</tbody>
</table>

**Table 4** Performance comparison of travel route recommendation under the travel time budget in terms of tour $F_1$ score ($F_1$).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Budapest</th>
<th>Edinburgh</th>
<th>Glasgow</th>
<th>Toronto</th>
<th>Vienna</th>
</tr>
</thead>
<tbody>
<tr>
<td>PersTour</td>
<td>0.372±0.389</td>
<td>0.358±0.391</td>
<td>0.539±0.458</td>
<td>0.410±0.438</td>
<td>0.320±0.360</td>
</tr>
<tr>
<td>PersTour+Tmf</td>
<td>0.343±0.405</td>
<td>0.378±0.425</td>
<td>0.532±0.459</td>
<td>0.425±0.444</td>
<td>0.365±0.374</td>
</tr>
<tr>
<td>PersTour+GTmf</td>
<td>0.394±0.401</td>
<td>0.430±0.401</td>
<td>0.541±0.458</td>
<td>0.450±0.449</td>
<td>0.379±0.377</td>
</tr>
<tr>
<td>PersTour+TGTmf</td>
<td>0.423±0.402</td>
<td>0.467±0.412</td>
<td>0.561±0.458</td>
<td>0.454±0.449</td>
<td>0.395±0.385</td>
</tr>
</tbody>
</table>

**Table 5** Performance comparison of travel route recommendation under the travel time budget in terms of tour pairs-$F_1$ score ($pairs$-$F_1$).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Budapest</th>
<th>Edinburgh</th>
<th>Glasgow</th>
<th>Toronto</th>
<th>Vienna</th>
</tr>
</thead>
<tbody>
<tr>
<td>PersTour</td>
<td>0.387±0.342</td>
<td>0.412±0.321</td>
<td>0.596±0.383</td>
<td>0.511±0.336</td>
<td>0.325±0.309</td>
</tr>
<tr>
<td>PersTour+Tmf</td>
<td>0.356±0.321</td>
<td>0.442±0.308</td>
<td>0.607±0.366</td>
<td>0.513±0.342</td>
<td>0.382±0.323</td>
</tr>
<tr>
<td>PersTour+GTmf</td>
<td>0.416±0.329</td>
<td>0.471±0.331</td>
<td>0.616±0.365</td>
<td>0.538±0.345</td>
<td>0.386±0.312</td>
</tr>
<tr>
<td>PersTour+TGTmf</td>
<td>0.434±0.352</td>
<td>0.504±0.343</td>
<td>0.627±0.373</td>
<td>0.540±0.348</td>
<td>0.402±0.326</td>
</tr>
</tbody>
</table>

**Table 6** Performance comparison of transition weight inference in terms of RMSE.

<table>
<thead>
<tr>
<th>Method</th>
<th>Budapest</th>
<th>Edinburgh</th>
<th>Glasgow</th>
<th>Toronto</th>
<th>Vienna</th>
</tr>
</thead>
<tbody>
<tr>
<td>Markov</td>
<td>0.10234</td>
<td>0.09195</td>
<td>0.10486</td>
<td>0.08776</td>
<td>0.09423</td>
</tr>
<tr>
<td>Tmf</td>
<td>0.09874</td>
<td>0.08995</td>
<td>0.09064</td>
<td>0.10056</td>
<td>0.10285</td>
</tr>
<tr>
<td>GTmf</td>
<td>0.08123</td>
<td>0.08003</td>
<td>0.07301</td>
<td>0.08270</td>
<td>0.08456</td>
</tr>
<tr>
<td>TGTmf</td>
<td>0.07923</td>
<td>0.07906</td>
<td>0.07208</td>
<td>0.08105</td>
<td>0.08023</td>
</tr>
</tbody>
</table>
fills unobserved transition weights according to the observed transitions.

**Impact of the Trade-off Parameter.** As introduced in Sect. 4.3, we combine location and transition rewards by using a trade-off parameter $\alpha$. This parameter controls the relative weights of location and location reward. The larger the value of $\alpha$, the more the tour planning depends on locations, as opposed to transitions. Because this is a very important parameter that directly affects the final travel route planning procedure, we have to discover the effects of different values. Therefore, in order to understand the importance of the reward for each part, we evaluate the impact of $\alpha$ by plotting recommendation performance under different $\alpha$ values.

Figure 7 plots the impact of the trade-off parameter in terms of the recommendation performance $F_1$ score under the route length budget. We take the performance produced by PoIRank as the baseline that only uses location reward, which will not change with the value of $\alpha$. Then various types of transition knowledge produced by Markov, Tmf, GTmf, and TGTmf are included upon the PoIRank baseline.

According to the plot results, we observe that the importance of location and transition varied in different datasets (cities). This variation might demonstrate the different travel styles of different cities. For instance, transition patterns are more emphasized in Edinburgh and Glasgow, according to Figs. 7 (b) and 7 (c), because the performance decreases as $\alpha$ increases. This can also be verified from Fig. 2, which shows that POIs in central Edinburgh are tightly linked to each other. In contrast, cities such as Budapest and Toronto rely more on location rewards, according to Figs. 7 (a) and 7 (d); the effects of transition reward on these datasets are not obvious.

Another interesting finding from these figures is that the trends of the different curves by introducing transition patterns are consistent with each other. The only difference might be caused by the inference ability of different methods. For instance, all $F_1$ scores in Figs. 7 (a) and 7 (d) gradually increase as $\alpha$ rises, while the method RANK+TGTmf achieves the best result. The reason might be the higher transition weight inference ability of our proposed method TGTmf, which includes additional spatial and temporal features.

Finally, among all different values of $\alpha$ on different datasets, we find that a value between 0.5 and 0.7 can always achieve a higher performance upon the location reward. This could be an important hint for developing real travel route recommendation systems with our proposed framework.

**Evaluation of Real Tourists** We collected feedback from seven foreign tourists according to the experimental setting described in Sect. 5.2 and calculated simple statistics for the results. The average rating scores of two examined
queries are summarized in Fig. 9 (a) and the users’ selection is shown in Fig. 9 (b). As shown in Fig. 9, the average rating scores of the routes recommended by adding transition knowledge are higher than those based only on POI popularity, on both queries. Therefore, almost all users selected the routes recommended by our proposed combination method, which were considered more helpful for their trip.

To better illustrate the effectiveness of our approach, we use a case study with query $q_1$ to see how transition knowledge affects the recommendation result compared with the location-based baseline method, which is presented in Fig. 10. The query $q_1$ implies that a user is departing from Kyoto National Museum to the hotel near Sanjo, hoping for a four-hour tour. Both routes are plotted on the Google map: red points indicate the most popular POIs and purple points represent relatively unpopular POIs. As shown by the plotted results, the main differences are caused by the visiting orders. Without transition knowledge, it maximized POI rewards and suggested users to visit Heian Shrine after visiting Maruyama Park. However, with transition knowledge, it suggested users to travel through Maruyama Park, Yasaka Shrine, and Gion in a straight line, which looks more natural and suitable for tourists. Also, our method changed the visiting order and suggested users to travel through Ninenzaka and Sanneizaka to Kiyomizu-dera, which is correct in reality because tourists have to climb these paths before entering Kiyomizu-dera.

6. Conclusion

This paper has proposed a technique that combines locations and transitions extracted from travel route data to recommend sightseeing routes according to user queries. To infer the city’s transition patterns, which represent location-location relations, we enhance the latent factorization model of a weighted transition matrix with additional spatial and temporal features, to enrich the latent features of location-location relations.

Real tourist travel route datasets were adopted for both qualitative and quantitative evaluation. Two types of travel budget constraints were considered to examine the generality of our proposed method. Comparing our method with recently published work, our method outperforms others in both qualitative and quantitative evaluations. Through the analysis on the recommendation performance and trade-off parameter, we reveal the efficiency by introducing transition patterns into the travel route recommendation task.

Acknowledgments

This work is partly supported by MIC SCOPE (172307001) and JSPS KAKENHI (16K12532).

References


