

Maximal Likelihood Itinerary Planning with User Interaction Data

Keisuke Otaki¹, Yukino Baba²

¹Toyota Central R&D Labs., Inc., Koraku Mori Building 10F, 1-4-14 Koraku, Bunkyo-ku, Tokyo, 1120004, Japan

²The University of Tokyo, 3-8-1 Komaba, Meguro-ku, Tokyo, 1538902, Japan

Abstract

Planning itineraries is a complex task for tourists. While some tourists have their favorite events and plans (e.g., places to visit, ways to travel) precisely in mind, others would like to explore multiple choices of possible events. To improve the user experiences of tourism, we develop a novel itinerary planning framework in which users directly interact with our system by editing displayed itineraries. Our idea is to collect edition-based feedback via editions by users, to estimate user preferences from the editions, and to utilize this data when generating personalized itineraries. To implement this framework, we generalize the maximum likelihood planning framework by introducing a new optimization problem to estimate transition probabilities between POIs with both historical and interaction data. To explain how the maximum likelihood itinerary planning-based method works, we report our proof-of-concept experiments aiming to provide a new perspective for interactive itinerary planning with user interaction.

Keywords

itinerary recommendation, orienteering problem, maximal likelihood planning, optimization

1. Introduction

Background Planning an itinerary (also called a travel plan or trajectory) is a complex task when a tourist plans a trip. Planning often involves places to visit (e.g., points-of-interests, POIs), places to stay (i.e., accommodations), how to travel between places (e.g., transportation and its mode), booking, and payments (if needed). While some tourists have their favorite places and/or plans exactly in mind, others would like to explore several choices to visit. In the literature, optimization-based methods have been studied as an important component for generating itineraries [1, 2, 3, 4]. A well-known optimization problem called the orienteering problem or its variants are often employed [5, 6]. The orienteering problem is the problem of constructing a trajectory (i.e., sequences of POIs) to maximize the benefits from the visited places under travel distances/time constraints. An important process behind the orienteering problem is how to evaluate the benefits of POIs for users. Using some objective values (e.g., an average rate or staying time of the POI) we can build traditional and common itineraries, while we can build personalized itineraries with some subjective values (e.g, a rate or staying time *by a specific user*).

RecSys Workshop on Recommenders in Tourism (RecTour 2022), September 22th, 2022, co-located with the 16th ACM Conference on Recommender Systems, Seattle, WA, USA

✉ otaki@mosk.tytlabs.co.jp (K. Otaki); yukino-baba@g.ecc.u-tokyo.ac.jp (Y. Baba)

🆔 0000-0001-9431-0867 (K. Otaki)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

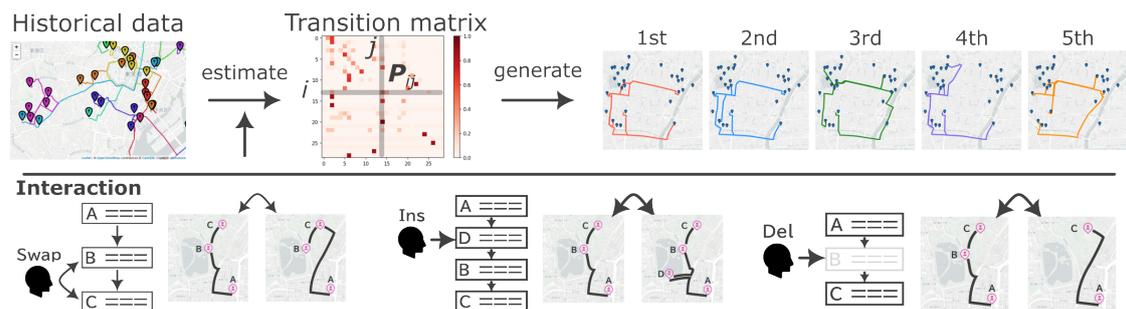


Figure 1: System overview. Our method generates a ranked list of itineraries. Our key component is a transition probability matrix among POIs. We try to update such a matrix using interaction data; we assume that users interact with our interface (e.g., Web/App), and operations like Swap, Ins, and Del are allowed to edit itineraries (see also Sec. 3.1 for editions).

Related Work Integrating user preferences for places and/or itineraries with optimization when generating *personalized* itineraries is promising to improving the user experiences. In previous work, Choudhury et al. [2] constructed a sequence of photographic spots using mean staying times of places to reflect user preferences, and they showed that generated trajectories are comparable with those generated by professionals. Lim et al. [3] utilized estimated staying times per user as user preferences in defining an objective function of the orienteering problem, but feedback is not considered. Roy et al. studied the task of interactively planning itineraries with feedback on POIs [7]. They proposed how to model such feedback and utilize them, but a type of supported feedback is limited. Chen et al. [4] adopted a similar strategy by [7], but formally discussed how historical data of itineraries are taken into account in optimization (this framework is referred as *maximum likelihood planning* in the literature [8, 9]).

Contributions We develop a new framework using both historical data and data collected by interaction with users. A systematic overview of our method is illustrated in Figure 1. We expect that using interaction data is promising to improve the user experience, and then generalize a type of feedback to collect richer data from users to estimate their preferences. Our method generates itineraries based on collected itineraries (**Historical data** in Fig. 1) and learned transition probabilities (**Transition matrix** in Fig.1) among point-of-interests (POIs). Further, in order to incorporate with these richer data, we try to update the learned probabilities. Note that we assume that interaction is implemented by some Web interface, and in this paper three types of operations for itineraries, *Swap*, *Ins*, and *Del* of POIs (illustrated in **Interaction** of Fig. 1), are considered. Our proposed method is a follower of both [7] and [4], but our estimation strategy using user interactions is quite new in the sense that such probabilities can be learned with collected interaction data. In our proof-of-concept experiments, we evaluate how collected interaction data affect the resulted ranked lists of itineraries under our assumption that the diversity on the resulted list is important to design itinerary planning service. We confirm that our framework generates diverse itineraries based on collected data.

2. Preliminary

Throughout this paper, $[N] = \{1, 2, \dots, N\}$ for some natural number $N \in \mathbb{Z}$. Any sequence is 1-origin. On a finite set I of symbols representing POIs, a length L -sequence consisting of elements from I is denoted by $\mathbf{X} = \langle X_1, X_2, \dots, X_L \rangle$, where $X_j \in I$ for any $j \in [L]$, and $L = |\mathbf{X}|$. A sequence represents an itinerary, where a user visits X_1 first, X_2 second, and X_L in last. A direct travel from i to j in \mathbf{X} is written as $i \rightarrow j \in \mathbf{X}$. In other words, for some $t \in [|\mathbf{X}| - 1]$, it holds that $X_t = i$ and $X_{t+1} = j$. Further, we write $i \in \mathbf{X}$ if and only if there exists $t \in [|\mathbf{X}|]$ such that $X_t = i$. In this paper, we naturally generalize this relation \in for sets $\mathcal{D} = \{\mathbf{X}_1, \dots, \mathbf{X}_{|\mathcal{D}|}\}$ of sequences. Our framework generates a ranked list of itineraries, and a length- k list $\mathcal{L} = \langle \mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \dots, \mathbf{X}^{(k)} \rangle$ represents a ranked list of itineraries.

2.1. Orienteering problem for itinerary planning

The orienteering problem is a well-studied combinatorial optimization problem [5], and it is applied to generate itineraries in the literature [2, 3, 6]. Without loss of generality, we assume that $1 \in V$ is the start POI and $N \in V$ is the goal POI when planning itineraries. The naive orienteering problem is defined on a complete graph $G = (V, E)$; the vertex set V represents a set of POIs, and the edge set E represents travels among POIs in V , and the problem involves finding a tour on G with some objectives and constraints. We assume that $t_{i,j}$ and $c_{i,j}$ represent the travel time and distance from i to j , respectively. T_{\max} is the total travel time, and the score $\text{Score}(i)$ for each POI $i \in V$ is given. We prepare decision variables $\{x_{i,j} \in \{0, 1\} \mid (i, j) \in E\}$ and $\{u_i \in \mathbb{Z} \mid i \in V\}$ as $x_{i,j} = 1$ if and only if j is visited after i , and u_i represents the order when i is visited. Then, the orienteering problem is formally described below.

$$\max_{x,u} \sum_{i=2}^{N-1} \sum_{j=2}^N \text{Score}(i) \cdot x_{i,j} \quad (1a)$$

$$\sum_{j=2}^N x_{1,j} = \sum_{i=1}^{N-1} x_{i,N} = 1, \quad \sum_{i=1}^{N-1} x_{i,k} = \sum_{j=2}^N x_{k,j} \quad (\forall k = 2, \dots, N-1) \quad (1b)$$

$$\sum_{i=1}^{N-1} \sum_{j=2}^N t_{i,j} x_{i,j} \leq T_{\max} \quad (1c)$$

$$u_1 = 1, u_N = L, 2 \leq u_i \leq N \quad (\forall i = [N] \setminus \{1\}) \quad (1d)$$

$$u_i - u_j + 1 \leq (N-1)(1 - x_{i,j}) \quad (\forall i, j \in [N] \setminus \{1\}) \quad (1e)$$

$$x_{i,j} \in \{0, 1\}, u_i \in \mathbb{Z} \quad (\forall i, j \in [N]) \quad (1f)$$

Note that Eq. (1a) requires us to travel popular POIs in V . Constraints Eq. (1b) ensure the resulted tour is valid. Constraints Eq. (1c) are for bounding the total travel time with respect to T_{\max} . Constraints Eq. (1d) and Eq. (1e) are from the well-known MTZ-constraint to avoid sub-tours [10]. Constraints Eq. (1f) define variables.

To consider the distance among POIs, a multi-objective function based on Eq. (1a) can be adopted with $\alpha, \beta \in \mathbb{R}$:

$$\max_{x,u} -\alpha \times \left(\sum_{i=1}^N \sum_{j=1}^N c_{i,j} \cdot x_{i,j} \right) + \beta \times \left(\sum_{i=2}^{N-1} \sum_{j=2}^N \text{Score}(i) \cdot x_{i,j} \right) \quad (2)$$

In our implementation, to generate length L -sequences from 1 to N , we replace Eq. (1c) with the following constraint $\sum_j y_j \leq L$ with $y_j \in \{0, 1\}$ for all $j \in [N]$, where $y_j = 1$ means that POI j is visited. In addition, by replacing Eq. (1a) with Eq. (2), we can build our baseline itinerary generation method using the orienteering problem.

2.2. Maximum likelihood planning

Let \mathbf{X} be a (feasible) itinerary and \mathcal{X} be the set of all feasible itineraries. In the following, we focus on the case of $|\mathbf{X}| = L$ for any $\mathbf{X} \in \mathcal{X}$. The goal of *maximum likelihood planning* is to solve $\max_{\mathbf{X} \in \mathcal{X}} \Pr(\mathbf{X})$. This problem setting has been attracted much attention in the literature [4, 8, 9]. Under a first order Markov chain approximation, $\Pr(\mathbf{X})$ for $\mathbf{X} = \langle X_1, X_2, \dots, X_L \rangle$ can be approximated as $\Pr(\mathbf{X}) \approx \Pr(X_1)\Pr(X_2 | X_1) \dots \Pr(X_L | X_{L-1})$. An implicit constraint $\Pr(X_1 = 1) = 1$ on our itinerary planning indicates the following:

$$\arg \max_{x,u} \Pr(\mathbf{X}) \approx \arg \min_{x,u} \sum_{(i,j) \in E} -\log \Pr(X_{t+1} = j | X_t = i) \cdot x_{ij} \quad (3)$$

Equation (3) indicates that existing solvers for the orienteering problem are directly applicable to the maximum likelihood planning of Eq. (3) [8, 9]. That is, using a solver with the cost value $\hat{c}_{ij} := -\log \Pr(X_{t+1} = j | X_t = i)$, we can obtain a maximal likelihood route \mathbf{X}^* . In the following, we write $P_{i,j} := \Pr(X_{t+1} = j | X_t = i)$ for the sake of simplicity.

2.3. Generating lists of solutions

Typical optimization problems and their solvers just output an optimal solution. However, displaying multiple solutions is preferred (e.g., on a Web) in applications. We now need to compute (possibly) top- k solutions with existing solvers, and some known methods are already proposed [11], where we need to fix an order among decision variables and to iteratively solve sub-problems. Instead of this procedure that requires some algorithmic design, we use the following implementation to obtain K solutions. Let $\mathcal{L}^{(m)}$ be the ordered set of m solutions and $\mathcal{L}^{(0)} = \emptyset$. To generate a next solution (i.e., m -th solution), we solve the optimization problem under additional constraints of $\mathbf{X} \neq \mathbf{X}^{(j)}$ for $\mathbf{X}^{(j)} \in \mathcal{L}^{(m-1)}$. Finally, we have an ordered set $\mathcal{L}^{(k)} = \{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \dots, \mathbf{X}^{(k)}\}$ consisting of k itineraries.

3. Proposed Framework

We propose a new framework in which users directly interact with our system by editing displayed itineraries. Our motivation to design this framework is collecting edition-based feedback via editions by users, estimating user preferences from the editions, and using such data when generating personalized itineraries. To implement this framework, we generalize

maximum likelihood planning by defining a new optimization problem to estimate transition probabilities among POIs with interaction data.

3.1. User Editions

A user interacts with service interfaces (e.g., Web/mobile app). We collect edition-based feedback from the user representing his/her preference among *itineraries*. In our system, the following three types of editions are considered.

Swap For $\mathbf{X} = \langle X_1, X_2, \dots, X_{|\mathbf{X}|} \rangle$, a swap of the two adjacent POIs generates a new itinerary $\mathbf{X}' = \langle X'_1, X'_2, \dots, X'_{|\mathbf{X}'|} \rangle$, where there exists $j \in [|\mathbf{X}|-1]$ such that $X'_j = X_{j+1}$, $X'_{j+1} = X_j$ and $X'_k = X_k$ for all $k \in [|\mathbf{X}'|] \setminus \{j, j+1\}$.

Insertion For $\mathbf{X} = \langle X_1, X_2, \dots, X_{|\mathbf{X}|} \rangle$, an insertion of a new location X' generates a new $\mathbf{X}' = \langle X'_1, X'_2, \dots, X'_{|\mathbf{X}'|} \rangle$ such that $|\mathbf{X}| + 1 = |\mathbf{X}'|$ and for some $j \in |\mathbf{X}| \setminus \{1, N\}$ it holds that $X'_j \notin \mathbf{X}$, $\mathbf{X}_k = \mathbf{X}'_k$ for $k \leq j-1, k > j$.

Deletion For $\mathbf{X} = \langle X_1, X_2, \dots, X_{|\mathbf{X}|} \rangle$, a deletion of some location $X' \in \mathbf{X}$ generates a new $\mathbf{X}' = \langle X'_1, X'_2, \dots, X'_{|\mathbf{X}'|} \rangle$ such that $|\mathbf{X}| - 1 = |\mathbf{X}'|$ and for some $j \in |\mathbf{X}| \setminus \{1, N\}$ it holds that $\mathbf{X}_k = \mathbf{X}'_k$ for $k \leq j-1, k > j$.

Note that these are different from existing work (e.g., [7]) that uses feedback only for POIs.

3.2. Maximum likelihood planning with user editions by optimization

We propose a new itinerary planning method using feedback from user editions defined in Sec. 3.1. Our idea consists of the following three steps.

1. converting the estimation task of $\{P_{i,j}\}_{i,j \in [N]}$ as an optimization problem,
2. optimizing our generalized optimization problem from (1) by penalty functions and collected feedback data, and computes a modified $\{\tilde{P}_{i,j}\}_{i,j \in [N]}$, and
3. adopting modified probabilities by (2) when generating maximum likelihood itineraries.

3.2.1. Step 1: Learning-based interpretation

Existing methods estimated $P_{i,j}$ by counting historical data \mathcal{D} (e.g., historical trajectories or routes) [4, 8]. A simple method to estimate $P_{i,j}$ is counting data in \mathcal{D} as $P_{i,j} = \frac{|\{i \rightarrow j \in \mathcal{D}\}|}{|\{i \in \mathcal{D}\}|}$. We cast this estimation problem as the following optimization problem:

$$\hat{P} := \arg \min_P \sum_{i,j} \left| P_{i,j} - \frac{|\{i \rightarrow j \in \mathcal{D}\}|}{|\{i \in \mathcal{D}\}|} \right|^2 \quad \text{subject to} \quad \sum_j P_{i,j} = 1 (\forall i \in [N]) \quad (4)$$

Here Eq. (4) can be solved by closed formula and a solution of Eq. (4) is denoted by \hat{P} below. Note that other variants have been discussed [8, 9]; For example, the Laplace smoothing with $\alpha > 0$ is possible to estimate $\hat{P}_{i,j}$, and this can also included in Eq. (4). In the following, we write the term $\sum_{i,j} \left| P_{i,j} - \frac{|\{i \rightarrow j \in \mathcal{D}\}|}{|\{i \in \mathcal{D}\}|} \right|^2$ with a loss function $L_{\text{data}}(P, \mathcal{D})$.

3.2.2. Step 2: Generalization

Our key idea in this paper is generalizing Eq. (4) to consider user feedback data using penalty functions. Intuitively, we define a new objective functions like $L_{\text{data}}(P, \mathcal{D}) + L(P, \mathcal{D}, \mathcal{D}_{\text{int}})$, where $L(P, \mathcal{D}, \mathcal{D}_{\text{int}})$ is the penalty term related to all of the estimated probabilities P , historical itinerary \mathcal{D} , and feedback data \mathcal{D}_{int} collected from users. In practice, we propose the following methods for the three types of editions.

Swap Let us explain using examples of length-4 sequences $\mathbf{X} = \langle X_1, X_2, X_3, X_4 \rangle$ and $\mathbf{X}' = \langle X_1, X_3, X_2, X_4 \rangle$. For \mathbf{X} and \mathbf{X}' , we encode the relation $\mathbf{X} \prec \mathbf{X}'$ by $\Pr(\mathbf{X}) < \Pr(\mathbf{X}')$. With our approximation, we have $P_{X_1, X_2} P_{X_2, X_3} P_{X_3, X_4} < P_{X_1, X_3} P_{X_3, X_2} P_{X_2, X_4}$. Then, we adopt a loss term $L_{\text{swap}}(P_{X_1, X_2} P_{X_2, X_3} P_{X_3, X_4} - P_{X_1, X_3} P_{X_3, X_2} P_{X_2, X_4})$ for each 4-tuple (X_1, X_2, X_3, X_4) , and add this term to our learning problem in Eq. (4) (see also **Swap** in Fig. 1).

Insertion For two example length-3 and 4 itineraries $\mathbf{X} = \langle X_1, X_2, X_3 \rangle$ and $\mathbf{X}' = \langle X_1, X_2, X_4, X_3 \rangle$, the insertion is encoded by $\Pr(\mathbf{X}) \leq \Pr(\mathbf{X}')$. Similarly, we should have $P_{X_2, X_3} \leq P_{X_2, X_4} P_{X_4, X_3}$, and a loss function L_{ins} is adopted as a penalty term (see also **Ins** in Fig. 1).

Deletion In contrast, for two example length-4 and 3 itineraries $\mathbf{X} = \langle X_1, X_2, X_3, X_4 \rangle$ and $\mathbf{X}' = \langle X_1, X_3, X_4 \rangle$, we can use a loss function L_{del} as well (see also **Del** in Fig. 1).

In summary, we can collect datasets \mathcal{D}_{int} by designing interfaces, and data like the above example (X_1, X_2, X_3, X_4) for **Swap** are stored to modify the transition probability $P_{i,j}$. Here we define a new objective function to estimate $P_{i,j}$ using both \mathcal{D} and $\mathcal{D}_{\text{int}} := \mathcal{D}_{\text{swap}} \sqcup \mathcal{D}_{\text{ins}} \sqcup \mathcal{D}_{\text{del}}$ as follows.

$$f := \gamma \times L_{\text{data}}(P, \mathcal{D}) + \delta_1 \times \sum_{(X_1, X_2, X_3, X_4) \in \mathcal{D}_{\text{swap}}} L_{\text{swap}}(X_1, X_2, X_3, X_4) + \delta_2 \times \sum_{(X_2, X_3, X_4) \in \mathcal{D}_{\text{ins}}} L_{\text{ins}}(X_2, X_3, X_4) + \delta_3 \times \sum_{(X_1, X_2, X_3) \in \mathcal{D}_{\text{del}}} L_{\text{del}}(X_1, X_2, X_3). \quad (5)$$

We write $\tilde{P} := \arg \min_P f(P; \gamma, \delta_1, \delta_2, \delta_3)$ subject to $\sum_j P_{i,j} = 1$ for all $i \in [N]$.

3.2.3. Step 3: Planning with modified probabilities

After computing Eq. (5), we obtain \tilde{P} instead of \hat{P} from Eq. (4), where we expect that \tilde{P} can reflect all interaction information from \mathcal{D}_{int} by soft constraints. We then could obtain different itineraries (e.g., top- k itineraries) by using \tilde{P} rather than those obtained by \hat{P} .

3.3. How our new optimization problem modifies transition matrices

We explain our framework using toy examples. Let us prepare $N = 10$ synthetic locations and generate $P_{X,Y}$ for $X, Y \in [N]$ randomly with $P_{X,X} = 0$. For our loss function, we adopt the

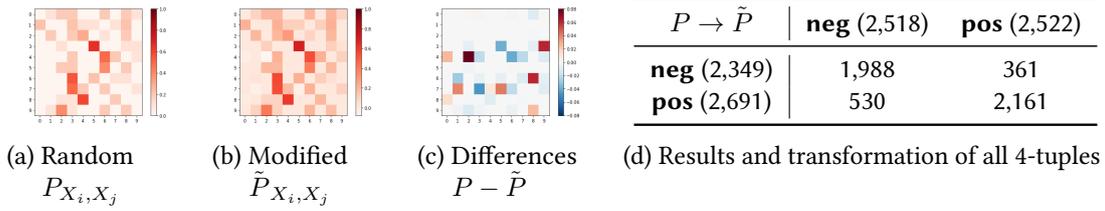


Figure 2: How our problem in Eq. (5) modifies P as \tilde{P} with 10 interaction **Swap** pairs and $\gamma = 0.25$, $\delta_1 = 16$, $\delta_2 = \delta_3 = 0$.

Frobenius norm for L_{data} and the tanh function for L_{swap} in Eq. (5), and explain our proposed method works as we expected for **Swap** operation as an example.

We first prepare a random transition matrix as shown in Fig. 2a. We randomly select 10 tuples (X_1, X_2, X_3, X_4) that violates the **Swap** condition to build $\mathcal{D}_{\text{swap}}$. Here, (X_1, X_2, X_3, X_4) is **neg** if $P_{X_1, X_2} P_{X_2, X_3} P_{X_3, X_4} < P_{X_1, X_3} P_{X_3, X_2} P_{X_2, X_4}$, and **pos** otherwise. We assume that a user says $\langle X_1, X_2, X_3, X_4 \rangle \prec \langle X_1, X_3, X_2, X_4 \rangle$. Using parameters $\gamma = 0.25$, $\delta_1 = 16$, $\delta_2 = \delta_3 = 0.0$, we compute a modified matrix \tilde{P} (as illustrated in Fig. 2b. $P - \tilde{P}$ is also shown in Fig. 2c).

In results, we have 2,349 **neg** and 2,691 **pos** tuples by P (i.e., total ${}_{10}P_4 = 5,040$ tuples), and 2,518 **neg** and 2,522 **pos** tuples in \tilde{P} , respectively. Out of 2,349 **neg** tuples by P , 1,988 tuples remain **neg**, and 361 tuples become **pos**. Similarly, out of 2,691 **pos** tuples by P , 530 tuples become **neg**, while 2,161 tuples are **pos** as well, as summarized in Fig. 2d. For $\mathcal{D}_{\text{swap}}$, \tilde{P} satisfies the condition for 7 tuples out of 10. We then confirm that 10 interaction samples in $\mathcal{D}_{\text{swap}}$ softly affect a subset of 4-tuples by \tilde{P} . Note that other loss functions for both L_{data} and L_{swap} are applicable.

4. Proof-of-concept experiments

We demonstrate how our proposed framework works with crawled real data. In the following experiments, we keep the two functions (the Frobenius norm for L_{data} and tanh for L_{swap}) for our method, and focus on **Swap** only in Eq. (5). In this paper, we only evaluate how collected interaction data $\mathcal{D}_{\text{swap}}$ affect the computed ranked list of itineraries. To evaluate this, we compare resulted lists with multiple settings, and quantitatively compare them.

Setup We extracted user-generated itineraries from TripHobo and rating data from TripAdvisor¹. We found itineraries tagged with Tokyo, and collected individual itineraries. An itinerary consists of several days (i.e., on day 1, on day 2, etc.). We then divided the multi-day itinerary into a set of one-day ones to focus on planning within a day. We collected such one-day itineraries to make a whole set as historical data. From the whole dataset, we only sampled a selected area of Tokyo, named Asakusa², and we finally have 245 itineraries in our \mathcal{D} . With

¹<https://www.triphobo.com/> and <https://www.tripadvisor.jp/> (access confirmed at June, 2022.)

²By selecting POIs whose locations are included in the area of latitude [35.443674, 35.825408] and longitude [139.514896, 139.927981].

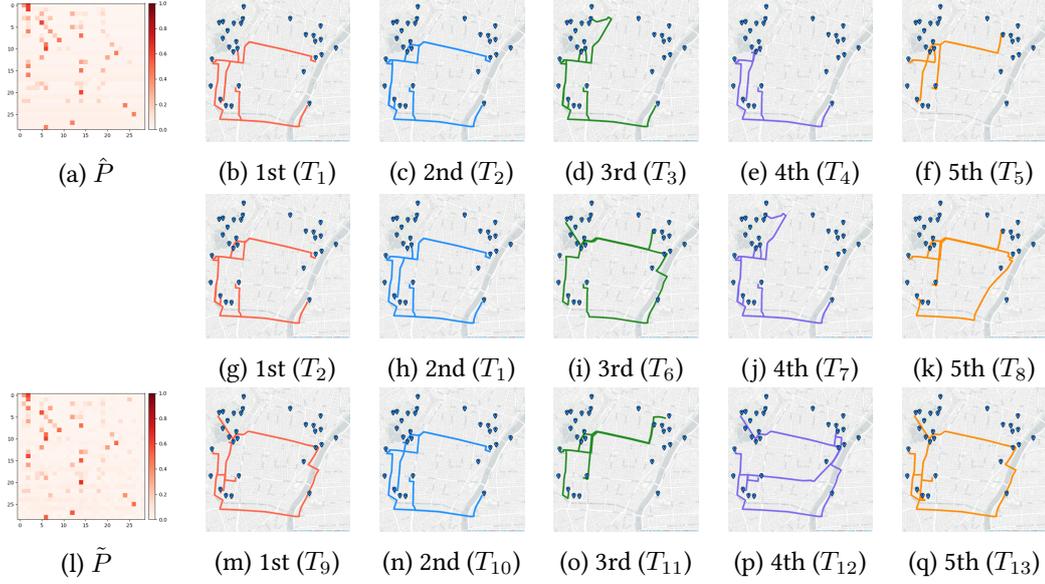


Figure 3: Two matrices \hat{P} in Fig. 3a and \tilde{P} in Fig. 3l. Resulted itineraries with \hat{P} and $\beta = 0$ (above, from Fig. 3b–Fig. 3f), \hat{P} and $\beta = 1$ (middle, from Fig. 3g–Fig. 3k), and with \tilde{P} and $\beta = 0$ (bottom, from Fig. 3m–Fig. 3q).

extracted itineraries, we also collected a set $[N]$ of all POIs in the data ($N = 29$). For each poi $i \in [N]$, we obtained $\text{Score}(i)$ from stars recorded in TripAdvisor.

We implemented our top- k itinerary planning algorithm (as in Sec. 2.3), set $k = 5$, and tested $\alpha = 1$ and $\beta \in \{0, 1\}$. To evaluate our method, we compared obtained lists of top-5 itineraries by \hat{P} and \tilde{P} . To learn \tilde{P} , we just randomly sampled 300 pairs as $\mathcal{D}_{\text{swap}}$ from **neg** 4-tuples as an simulation data. Parameter settings were the same to those in Sec. 3.3.

Visualization of generated itineraries For random start and goal POIs out of 29 POIs, with \hat{P} for both $\beta = 0$ and $\beta = 1$ cases, the baseline method generated 10 itineraries in total, as illustrated in Fig. 3, where we had 8 unique itineraries. Using identifiers depicted in Fig. 3, we had $\mathcal{L}_1 = \langle T_1, T_2, T_3, T_4, T_5 \rangle$ when \hat{P} and $\beta = 0$, and $\mathcal{L}_2 = \langle T_2, T_1, T_6, T_7, T_8 \rangle$ when \hat{P} and $\beta = 1$. With \tilde{P} for both $\beta = 0$, Fig. 3 illustrates a new itineraries generated through our framework, where we have a new list $\mathcal{L}_3 = \langle T_9, T_{10}, T_{11}, T_{12}, T_{13} \rangle$ when \tilde{P} and $\beta = 0$. For \tilde{P} and $\beta = 1$, we have another list $\mathcal{L}_4 = \langle T_{10}, T_9, T_{11}, T_{13}, T_2 \rangle$. Note that \mathcal{L}_4 is not illustrated in Fig. 3 as itineraries in \mathcal{L}_4 are already illustrated.

Evaluations To evaluate itineraries in terms of scores ($\sum_{i=2}^{N-1} \sum_{j=2}^N \text{Score}(i)x_{i,j}$) and ranking (i.e., $\mathcal{L}_1, \mathcal{L}_2$, and \mathcal{L}_3), we first measure total scores and travel costs of each itinerary. Figure 4a shows a scatter plot of the two terms of Eq. (2); x -axis shows total travel distances of itineraries and y -axis represent obtained values by itineraries. Next, we evaluate \mathcal{L}_3 with different sizes of $\mathcal{D}_{\text{swap}}$. Figure 4b shows how top-5 lists vary when numbers of **neg** samples increases (corresponding to x -axis, from 0 to 500.), where y -axis represents top- k ranking with black

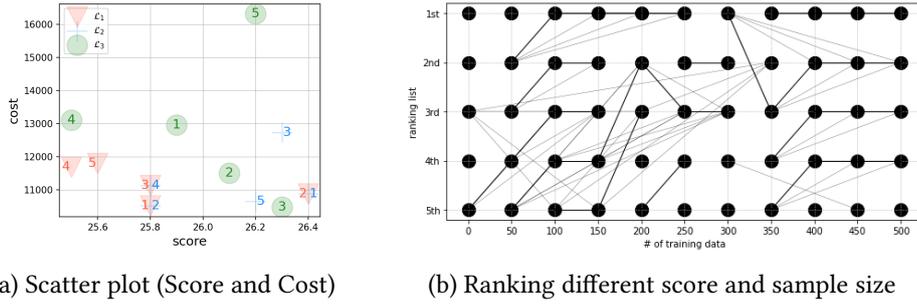


Figure 4: Comparisons of itineraries and ranking. Fig. 4a shows a scatter plot among scores and travel costs. Fig. 4b shows how ranking lists vary when the number of training data in $\mathcal{D}_{\text{swap}}$ increases.

circles, and a line between two circles indicates the two itineraries are the same.

In results, Fig. 3, Fig. 4a, and Fig. 4b indicate that we could generate a variety of itineraries by our approach. In other words, our proposed method diversified the ranking results based on interaction data.

5. Conclusion

We proposed a new framework in which users directly interact with our system by editing displayed itineraries. Our idea is collecting rich feedback via editions by users, and utilizing such data when generating personalized itineraries. Throughout our proof-of-concept experiments, we confirm that our method could diversify ranking generated by top- k itinerary generation via the orienteering problem.

In our future work, we will investigate more deeply learning-based methods via interaction data, and plan a quantitative user study to develop interaction and optimization-based itinerary planning method like [2].

References

- [1] A. A. da Silva, R. Morabito, V. Pureza, Optimization approaches to support the planning and analysis of travel itineraries, *Expert Systems with Applications* 112 (2018) 321–330. URL: <https://www.sciencedirect.com/science/article/pii/S0957417418303920>. doi:<https://doi.org/10.1016/j.eswa.2018.06.045>.
- [2] M. De Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, C. Yu, Automatic construction of travel itineraries using social breadcrumbs, in: *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia, HT '10*, Association for Computing Machinery, New York, NY, USA, 2010, pp. 35–44.
- [3] K. H. Lim, J. Chan, C. Leckie, S. Karunasekera, Personalized tour recommendation based on user interests and points of interest visit durations, in: *Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15*, AAAI Press, Buenos Aires, 2015, pp. 1778–1784.

- [4] D. Chen, C. S. Ong, L. Xie, Learning points and routes to recommend trajectories, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 2227–2232.
- [5] P. Vansteenwegen, W. Souffriau, D. Van Oudheusden, The orienteering problem: A survey, *European Journal of Operational Research* 209 (2011) 1–10.
- [6] Z. Friggstad, S. Gollapudi, K. Kollias, T. Sarlos, C. Swamy, A. Tomkins, Orienteering algorithms for generating travel itineraries, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 180–188.
- [7] S. Basu Roy, G. Das, S. Amer-Yahia, C. Yu, Interactive itinerary planning, in: 2011 IEEE 27th International Conference on Data Engineering, ICDE '11, IEEE Computer Society, USA, 2011, pp. 15–26. doi:10.1109/ICDE.2011.5767920.
- [8] R. Canoy, T. Guns, Vehicle routing by learning from historical solutions, in: Proc. of CP2019, 2019, pp. 54–70.
- [9] J. Mandi, R. Canoy, V. Bucarey, T. Guns, Data driven vrp: A neural network model to learn hidden preferences for vrp, in: Proc. of CP2021, 2021.
- [10] C. E. Miller, A. W. Tucker, R. A. Zemlin, Integer programming formulation of traveling salesman problems, *Journal of the ACM (JACM)* 7 (1960) 326–329.
- [11] E. L. Lawler, A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem, *Management science* 18 (1972) 401–405.