Knowl Inf Syst (2018) 54:375–406
https://doi.org/10.1007/s10115-017-1056-y

REGULAR PAPER

Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency

Kwan Hui Lim[1,2] · **Jeffrey Chan**[1,3] · **Christopher Leckie**[1] · **Shanika Karunasekera**[1]

Received: 15 December 2015 / Revised: 24 March 2017 / Accepted: 19 April 2017 /
Published online: 5 May 2017

Abstract Tour recommendation and itinerary planning are challenging tasks for tourists, due to their need to select points of interest (POI) to visit in unfamiliar cities and to select POIs that align with their interest preferences and trip constraints. We propose an algorithm called PersTour for recommending personalized tours using POI popularity and user interest preferences, which are automatically derived from real-life travel sequences based on geotagged photographs. Our tour recommendation problem is modeled using a formulation of the Orienteering problem and considers user trip constraints such as time limits and the need to start and end at specific POIs. In our work, we also reflect levels of user interest based on visit durations and demonstrate how POI visit duration can be personalized using this time-based user interest. Furthermore, we demonstrate how PersTour can be further enhanced by: (i) a weighted updating of user interests based on the recency of their POI visits and (ii) an automatic weighting between POI popularity and user interests based on the tourist's activity level. Using a Flickr dataset of ten cities, our experiments show the effectiveness of PersTour against various collaborative filtering and greedy-based baselines, in terms of tour popularity, interest, recall, precision and F_1-score. In particular, our results show the merits of using time-based user interest and personalized POI visit durations, compared to the current practice of using frequency-based user interest and average visit durations.

✉ Kwan Hui Lim
limk2@student.unimelb.edu.au; kwanhui@graduate.uwa.edu.au

Jeffrey Chan
jeffrey.chan@rmit.edu.au

Christopher Leckie
caleckie@unimelb.edu.au

Shanika Karunasekera
karus@unimelb.edu.au

1 Department of Computing and Information Systems, University of Melbourne, Parkville, Australia

2 Data61, CSIRO, Canberra, Australia

3 School of Science, RMIT University, Melbourne, Australia

Keywords Tour recommendation · Itinerary planning · User interests · Personalization · Orienteering problem · Flickr · Wikipedia · Social networks

1 Introduction

Tour recommendation and itinerary planning are challenging tasks due to the different interest preferences and trip constraints (e.g., time limits, start and end points) of each unique tourist.[1] While there is an abundance of information from the Internet and travel guides, many of these resources simply recommend individual points of interest (POI) that are deemed to be popular, but otherwise do not appeal to the interest preferences of users or adhere to their trip constraints. Furthermore, the massive volume of information makes it a challenge for tourists to narrow down to a potential set of POIs to visit in an unfamiliar city. Even after the tourist finds a suitable set of POIs to visit, it will take considerable time and effort for the tourist to plan the appropriate duration of visit at each POI and the order in which to visit the POIs.

To address these issues, we propose the PersTour algorithm for recommending personalized tours where the suggested POIs are optimized to the users' interest preferences and POI popularity. We formulate our tour recommendation problem based on the Orienteering problem [42], which considers a user's trip constraints such as time limitations and the need for the tour to start and end at specific POIs (e.g., POIs near the tourist's hotel). Using geo-tagged photographs as a proxy for tourist visits, we are able to extract real-life user travel histories, which can then be used to automatically determine a user's interest level in various POI categories (e.g., parks, beaches, shopping) as well as the popularity of individual POIs. As tourists have different preference levels between POI popularity and POI relevance to their interests, our PersTour algorithm also allows tourists to indicate their preferred level of trade-off between POI popularity and his/her interest preferences. In cases where the tourist prefers to automate the indication of this trade-off between POI popularity and interest preference, PersTour is also able to determine the appropriate trade-off based on the activity level of the tourist relative to the POI visits of the general population.

Our main contributions[2] are as follows:

1. We propose the PersTour algorithm for recommending personalized tour/trip itineraries with POIs and visit duration based on POI popularity, users' interest preferences and trip constraints. Our tour recommendation problem is modeled in the context of the Orienteering problem (Sect. 3).
2. We introduce the concept of *time-based user interest* for tour recommendation, where a user's level of interest in a POI category is based on his/her time spent at such POIs, relative to the average user. We also compare our time-based user interest to the current practice of using frequency-based user interest and show how time-based user interest results in recommended tours that more accurately reflect real-life travel sequences (Sect. 3.1).
3. We also further enhance *time-based user interest* by implementing an update rule such that user interests are refined based on the recency of their past POI visits. This updating works by giving more emphasis to recent POI visits than those in the more distant past (Sect. 3.1).

[1] We use the terms "tourist" and "user" interchangeably, and similarly for the terms "tour" and "trip."

[2] This publication is an extended version of Lim et al. [27] that appeared in IJCAI'15, with the additional contributions of Points 3, 5 and 7.

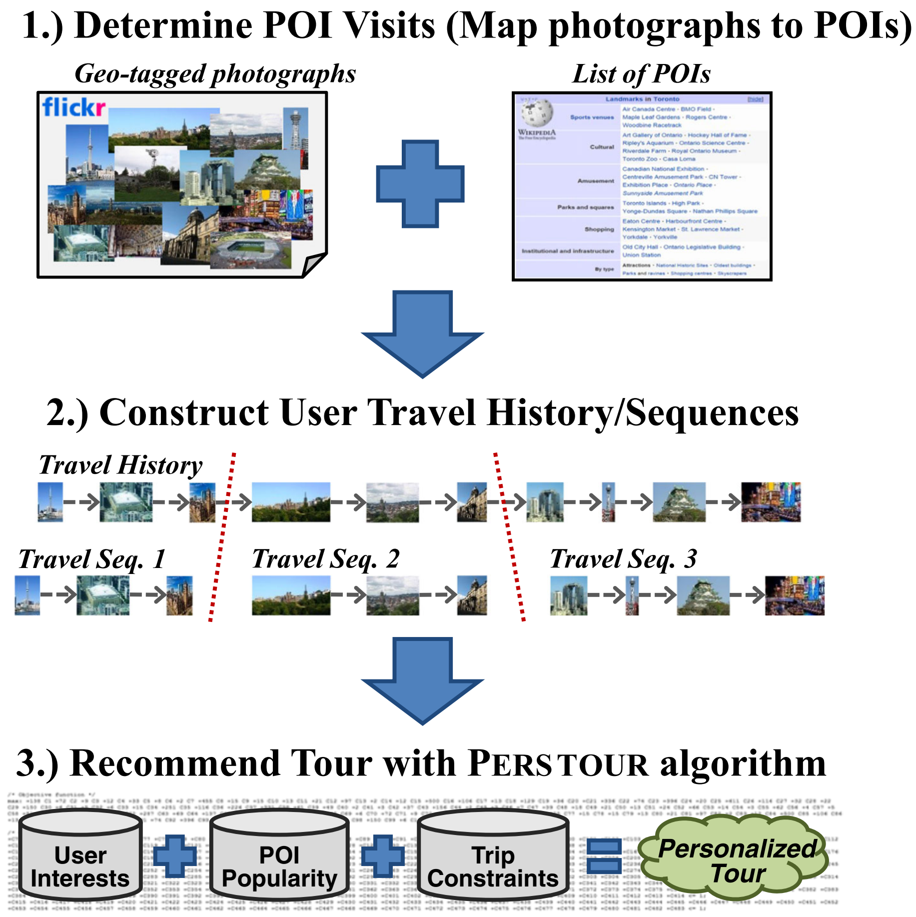

Fig. 1 Tour recommendation framework

4. We demonstrate the *personalization of POI visit duration* using time-based user interest, for the purpose of tour/trip itinerary recommendation. Our results show that personalized visit durations more accurately reflect the real-life POI visit durations of users, compared to the current practice of using average visit duration (Sect. 3.1).
5. While the PERSTOUR algorithm gives tourists the flexibility to indicate their preferred weightage between POI popularity and his/her interests, we also propose two schemes to automatically determine an appropriate weightage based on the tourist's activity level, relative to the general tourist population (Sect. 3.2.1).
6. We implement a framework (Fig. 1) for extracting real-life user travel histories based on their geo-tagged photographs, which are then used for training our PERSTOUR algorithm and serve as ground truth for our subsequent evaluation (Sect. 4).

7. We evaluate different variants of PersTour against various baselines using a Flickr dataset spanning ten cities. Our results show that PersTour out-performs these baselines based on tour popularity, user interest, recall, precision and F_1-score (Sects. 5 and 6).

The rest of the paper is structured as follows: Sect. 2 discusses some related work in tour recommendation; Sect. 3 introduces some preliminaries and defines our research problem; Sect. 4 describes our overall framework for tour recommendation; Sect. 5 outlines our experimental methodology; Sect. 6 discusses our main results and key findings; and Sect. 7 summarizes and concludes our paper.

2 Related work

Tour recommendation has been a well-studied field, with many developed applications [5,7,29,43,47] and research ranging from recommending beautiful, quiet and happy tours [33] to tour recommendation using random walks with restart [30]. In our review of related work, we focus on research related to our work and refer readers to [12,38] for an overview on the general field of tour recommendation. In the following sections, we provide an overview of the Orienteering problem before highlighting some key works in tour itinerary recommendations.

2.1 Background on the Orienteering problem

The Orienteering problem [42] originated from a competition of the same name. In this Orienteering competition, there are multiple navigational checkpoints distributed throughout an area, where each checkpoint is associated with a certain score. The main objective of participants in this competition is to maximize their total score, which is accumulated from visiting the various checkpoints. Participants are only given a limited amount of time to maximize their scores, and the winner is the participant who has accumulated the highest score. Due to this limitation of time, participants have to strategize and select a smaller subset of checkpoints to visit and decide on the sequence to visit these checkpoints. For a more in-depth review of the Orienteering problem, we refer readers to [14,45]. In recent years, various works have used the Orienteering problem to model different variations of the tour recommendation problem, and we discuss some of these works.

2.2 Tour recommendation based on Orienteering problem and its variants

Many tour itinerary recommendation works are based on the Orienteering problem and its variants. For example, Choudhury et al. [9] was one of the earlier tour recommendation studies based on the Orienteering problem, where recommended tours start and end at specific POIs while trying to maximize an objective score. Using a modified Orienteering problem, Gionis et al. [13] utilized POI categories such that recommended tours are constrained by a POI category visit order (e.g., museum → park → beach). Similarly, Lim [24] used a modified Orienteering problem constrained by a mandatory POI category, which corresponds to the POI category a user is most interested in. Based on user-indicated interests and trip constraints (e.g., time budget, start and end locations), Vansteenwegen et al. [44] recommended tours comprising POI categories that best match user interests while adhering to these trip constraints. In the context of theme parks, Lim et al. [25] recommended personalized itineraries with minimal queuing times at attractions, while maximizing user interests and attraction popularity. Others like [1,28] have extended the Orienteering problem for the purpose of recommending tour itineraries for groups of tourists, with the aim of satisfying the diverse interest preferences of multiple tourists in a group.

2.3 Tour recommendation based on other combinatorial optimization problems

In contrast to works based on the Orienteering problem, there are also various tour itinerary recommendation works based on other combinatorial optimization and similar problems. For example, Brilhante et al. [4] formulated tour recommendation as a Generalized Maximum Coverage problem [10], with the objective of finding an optimal set of POIs based on both POI popularity and user interest. Thereafter, Brilhante et al. [6] extended upon the former by using a variation of the Traveling Salesman Problem, with the main aim of finding the shortest route among the set of optimal POIs recommended in [4]. In addition to user interests in tour recommendation, Chen et al. [8] also considered traveling times based on different traffic conditions, using trajectory patterns derived from taxi GPS traces. Focusing on traveling paths based on road segments between POIs, Sun et al. [40] recommended tour itineraries comprising popular POIs and interesting routes between these POIs, with POI and route popularity based on geo-tagged photographs. With further considerations for different transport modes, Kurashima et al. [19,20] used a combined topic and Markov model to recommend tours based on both user interests and frequently traveled routes.

2.4 Top-k POI recommendation and next-location prediction

The recommendation of top-k POIs and next-location predictions is also closely related to our problem of tour itinerary recommendation. For example, LearNext [2] used Gradient Boosted Regression Trees and Ranking SVMs to predict the (single) next POI that a tourist will visit, while [49] performed a similar next-location prediction task using Markov models, along with seasonal and temporal information. Others like [36] used a category-regularized matrix factorization approach for recommending individual POIs, and Kofler et al. [17] proposed a system prototype for recommending individual POIs that are niche and specialized in nature. For top-k POI recommendations, many works utilized variants of matrix factorization or collaborative filtering approaches to recommend a ranked list of k POIs, using information such as friendship links [50], types of activities/users [21] and temporal patterns in POI visits [52].

2.5 Other tourism-related work

There are also many interesting tourism-related studies that utilize geo-tagged photographs for purposes ranging from identifying popular POIs to analyzing tourist behavior. For example, Ji et al. [15] implemented a graph modeling framework to identify popular POIs based on photographs posted in blogs, while [32] used geo-tagged photographs to understand tourist behavior based on their POI visit patterns and time spent. More generally, geo-tagged photographs have been used for other purposes such as predicting friendship relationships based on spatiotemporal links [11], identifying local clusters of interesting events and places [16] and estimating the location where a photograph is taken [22]. For a more comprehensive discussion of research that utilizes geo-tagged photographs, we direct readers to [39], who presented a comprehensive review of current applications and identified various interesting future directions.

2.6 Discussion of differences with previous work

While these previous works are the state of the art in tourism-related research, our proposed work differs from these earlier works in various aspects. First, we automatically derive a relative measure of *time-based user interest* using a user's visit durations at POIs of a specific category, relative to the average visit durations of other users, whereas earlier tour recommendation works either use frequency-based user interest (based on POI visit frequency) or require users to explicitly indicate their interest preferences for tour itinerary recommendation. Second, we plan and recommend tour itineraries with *personalized POI visit durations* that cater to individual users based on their time-based user interests, whereas previous works recommend tour itineraries using the same non-personalized POI visit duration for all users (either the average duration or a fixed duration, e.g., 1 h at all POIs) or do not consider POI visit duration at all. Third, although the works on top-k POI recommendation and next-location prediction are related to our tour itinerary recommendation problem, our proposed problem involves the additional considerations of user interest preferences, POI popularity, time constraints, starting/ending locations and more importantly, recommending a connected tour itinerary that satisfies these considerations, instead of individual POIs. While the other tourism-related works illustrate many interesting applications of geo-tagged photographs, these works use such photographs to study tourist behavior and identify popular POIs, which are distinctly different from the task of recommending a personalized tour itinerary.

3 Background and Problem definition

In this section, we first examine some preliminary definitions, before introducing a formulation of our tour recommendation problem.

3.1 Preliminaries

If there are m POIs for a particular city, let $P = \{p_1, \ldots, p_m\}$ be the set of POIs in that city. Each POI p is also labeled with a category Cat_p (e.g., church, park, beach) and latitude/longitude coordinates. We denote a function $Pop(p)$ that indicates the popularity of a POI p, based on the number of times POI p has been visited. Similarly, the function $T^{Travel}(p_x, p_y)$ measures the time needed to travel from POI p_x to p_y, based on the distance between POIs p_x and p_y and the indicated traveling speed. For simplicity, we use a traveling speed of 4 km/h, i.e., a leisure walking speed.[3]

Definition 1 *Travel history* Given a user u who has visited n POIs, we define his/her travel history as an ordered sequence, $S_u = ((p_1, t^a_{p_1}, t^d_{p_1}), \ldots, (p_n, t^a_{p_n}, t^d_{p_n}))$, with each triplet $(p_x, t^a_{p_x}, t^d_{p_x})$ comprising the visited POI p_x, and the arrival time $t^a_{p_x}$ and departure time $t^d_{p_x}$ at POI p_x. Thus, the visit duration at POI p_x can be determined by the difference between $t^a_{p_x}$ and $t^d_{p_x}$. Similarly, for a travel sequence S_u, $t^a_{p_1}$ and $t^d_{p_n}$ also indicate the start and end time of the itinerary, respectively. For brevity, we simplify $S_u = ((p_1, t^a_{p_1}, t^d_{p_1}), \ldots, (p_n, t^a_{p_n}, t^d_{p_n}))$ as $S_u = (p_1, \ldots, p_n)$.

[3] $T^{Travel}(p_x, p_y)$ can be easily generalized to different transport modes (e.g., taxi, bus, train) and to also consider the traffic condition between POIs (e.g., longer travel times between two POIs in a congested city, compared to two equal-distanced POIs elsewhere).

Definition 2 *Travel sequence* Based on the travel history S_u of a user u, we can further divide this travel history into multiple travel sequences, i.e., sub-sequences of S_u. We divide a travel history S_u into separate travel sequences if $t^d_{p_x} - t^a_{p_x+1} > \tau$. That is, we separate a travel history into distinct travel sequences if the consecutive POI visits occur more than τ time units apart. Similar to other works [9,24], we choose $\tau = 8\ h$ in our experiments. These travel sequences also serve as the ground truth of real-life user trajectories, which are subsequently used for evaluating our PERSTOUR algorithm and baselines. For a user u with n travel sequences, we use $S^1_u, S^2_u, \ldots, S^n_u$ to denote the different travel sequences in temporal order, such that S^1_u took place before S^2_u.

Definition 3 *Average POI visit duration* Given a set of travel histories for all users U, we determine the average visit duration for a POI p as follows:

$$\bar{V}(p) = \frac{1}{n}\sum_{u \in U}\sum_{p_x \in S_u}\left(t^d_{p_x} - t^a_{p_x}\right)\delta(p_x = p), \quad \forall\, p \in P \tag{1}$$

where n is the number of visits to POI p by all users and $\delta(p_x = p) = \{^{1,\ p_x=p}_{0,\ \text{otherwise}}$. $\bar{V}(p)$ is commonly used in tour recommendation as the POI visit duration for all users [4,6,8], while many earlier works do not factor in POI visit durations at all. In our work, we show how recommended POI visit durations can be personalized to individual users based on their interest (Definition 5), and use $\bar{V}(p)$ as a comparison baseline, i.e., the non-personalized POI visit duration.

Definition 4 *Time-based user interest* As described earlier, the category of a POI p is denoted Cat_p. Given that C represents the set of all POI categories, we determine the interest of a user u in POI category c as follows:

$$Int^{Time}_u(c) = \sum_{p_x \in S_u}\frac{\left(t^d_{p_x} - t^a_{p_x}\right)}{\bar{V}(p_x)}\delta(Cat_{p_x} = c), \quad \forall\, c \in C \tag{2}$$

where $\delta(Cat_{p_x} = c) = \{^{1,\ Cat_{p_x}=c}_{0,\ \text{otherwise}}$. In short, Eq. 2 determines the interest of a user u in a particular POI category c, based on his/her time spent at each POI of category c, relative to the average visit duration (of all users) at the same POI. The rationale is that a user is likely to spend more time at a POI that he/she is interested in. Thus, by calculating how much more (or less) time a user is spending at POIs of a certain category compared to the average user, we can determine the interest level of this user in POIs of this category.

Definition 5 *Personalized POI visit duration* Based on our definition of time-based user interest (Eq. 2), we are able to personalize the recommended visit duration at each POI based on each user's interest level. We determine the personalized visit duration at a POI p for a user u as follows:

$$T^{Visit}_u(p) = Int^{Time}_u(Cat_p) \times \bar{V}(p) \tag{3}$$

That is, we are recommending a personalized POI visit duration based on user u's relative interest level in category Cat_p multiplied by the average time spent at POI p. Thus, if a user is more (less) interested in category Cat_p, he/she will spend more (less) time at POI p than the average user.

Definition 6 *Frequency-based user interest* We also define a simplified version of user interest, denoted $Int^{Freq}_u(c)$, which is based on the number of times a user visits POIs of a certain category c (i.e., the more times a user visits POIs of a specific category, the more interested this user is in that category). As using $Int_u^{Freq}(c)$ is the current practice in tour recommendation research [4,6,24], we include it for a more complete study and as a comparison baseline to our proposed $Int_u^{Time}(c)$.

Definition 7 *Time-based user interest with weighted updates* We improve upon the original *time-based user interest* (Definition 4) by giving more emphasis to recent POI visits and less emphasis to POI visits in the distant past. Algorithm 1 details our proposed algorithm. In Line 9 of Algorithm 1, we continuously update user u's interest by minimizing the error between his/her recommended and actual POI visit duration, while $\frac{i}{n}$ ensures that more emphasis is given to more recent POI visits. Lines 6 to 8 calculate the error between the recommended and actual POI visit duration, while Lines 4 and 5 ensure that we perform this update for all POIs in all travel sequences of user u.

Algorithm 1: Time-based user interest with weighted updates

input : $\{S_u^1, S_u^2, \ldots, S_u^n\}$: The past travel sequences of a user u.
output: $Int_u^{Upd}(c)$: The updated interest levels for user u.
1 **begin**
2   **for** *POI category* $c \in C$ **do**
3     $Int_u^{Upd}(c) \leftarrow Int_u^{Time}(c)$;
4   **for** $i \leftarrow 1$ **to** n **do**
5   **for** *POI* $p \in S_u^i$ **do**
6       $recomTime \leftarrow Int_u^{Upd}(Cat_p) \times \bar{V}(p)$;
7       $actualTime \leftarrow t_p^d - t_p^a$;
8       $error \leftarrow \frac{recomTime - actualTime}{\bar{V}(p)}$;
9       $Int_u^{Upd}(c) \leftarrow Int_u^{Upd}(c) - \alpha \frac{i}{n} error$;

The intuition behind Algorithm 1 is that more recent POI visits are more relevant to a user and thus should contribute more to the modeling of this user's interest. Similarly, other researchers have also observed people's preference for more recent activities/information and utilized this recency preference for next check-in location prediction [26], location-based domain expert identification [23] and personalized music recommendation [35].

Definition 8 *Personalized POI visit duration with weighted updates* Similar to Definition 5, we can then recommend a personalized POI visit duration to POI p for a user u based on his/her *time-based user interest with weighted updates*, as follows:

$$T_u^{VisitUpd}(p) = Int_u^{Upd}(Cat_p) \times \bar{V}(p) \tag{4}$$

Similar to Definition 5, we are personalizing the POI visit duration for user u based on his/her updated interest level in category Cat_p multiplied by the average time that users spend at POI p.

3.2 Problem definition

We now define our tour recommendation problem in the context of the Orienteering problem and its integer problem formulation [24,42,45]. Given the set of POIs P, a budget B, starting

POI p_1 and destination POI p_N, our goal is to recommend an itinerary $I = (p_1, \ldots, p_N)$ that maximizes a certain score S while adhering to the budget B.[4] In this case, the score S is represented by the popularity and user interest of the recommended POIs using the functions $Pop(p)$ and $Int(Cat_p)$, respectively. The budget B is based on time spent and calculated using the function $Cost(p_x, p_y) = T^{Travel}(p_x, p_y) + T_u^{Visit}(p_y)$, i.e., using both traveling time and personalized visit duration at the POI. One main difference between our work and earlier work is that we personalize the visit duration at each recommended POI based on user interest (Definition 5), instead of using the average visit duration for all users or not considering visit duration at all. Formally, we want to find an itinerary $I = (p_1, \ldots, p_N)$ that:

$$Max \sum_{i=2}^{N-1} \sum_{j=2}^{N} x_{i,j} \Big(\eta Int(Cat_i) + (1-\eta) Pop(i) \Big) \tag{5}$$

where $x_{i,j} = 1$ if both POI i and j are visited in sequence (i.e., we travel directly from POI i to j), and $x_{i,j} = 0$ otherwise. We attempt to solve for Eq. 5, such that:

$$\sum_{j=2}^{N} x_{1,j} = \sum_{i=1}^{N-1} x_{i,N} = 1 \tag{6}$$

$$\sum_{i=1}^{N-1} x_{i,k} = \sum_{j=2}^{N} x_{k,j} \le 1, \quad \forall\, k = 2, \ldots, N-1 \tag{7}$$

$$\sum_{i=1}^{N-1} \sum_{j=2}^{N} Cost(i, j) x_{i,j} \le B \tag{8}$$

$$2 \le p_i \le N, \quad \forall\, i = 2, \ldots, N \tag{9}$$

$$p_i - p_j + 1 \le (N-1)(1 - x_{i,j}), \quad \forall\, i, j = 2, \ldots, N \tag{10}$$

Equation 5 is a multi-objective function that maximizes the popularity and interest of all visited POIs in the itinerary, where η is the weighting given to the popularity and interest components. Equation 5 is also subject to constraints 6–10. Constraint 6 ensures that the itinerary starts at POI 1 and ends at POI N, while constraint 7 ensures that the itinerary is connected and no POIs are visited more than once. Constraint 8 ensures that the time taken for the itinerary is within the budget B, based on the function $Cost(p_x, p_y)$ that considers both traveling time and personalized POI visit duration. Given that p_x is the position of POI x in itinerary I, constraints 9 and 10 ensure that there are no sub-tours in the proposed solution, adapted from the sub-tour elimination used in the Traveling Salesman Problem [31].

Based on this problem definition, we can then proceed to solve our tour recommendation problem as an integer programming problem. For solving this integer programming problem, we used the lpsolve linear programming package [3]. We denote our proposed algorithm for personalized tour recommendation as PersTour and shall describe our overall framework and the different PersTour variants in the following section.

3.2.1 Adaptive weighting

As introduced in Eq. 5, the η parameter offers tourists the flexibility to indicate their preferences for POI popularity and interest alignment. In this section, we propose two methods

[4] Although we examine POIs in this work, our tour recommendation problem definition can be easily modified such that a recommended tour itinerary starts and ends at a specific hotel where the tourist is staying at.

that automatically determine an appropriate value for the η parameter based on the POI visits by the general user population.

Given all users U and their set of travel histories $S_{u \in U}$, we define the number of POI visit count for a user u as: $C_u = |S_u|$. Similarly, C_{max} denotes the maximum POI visit count out of all users $u \in U$. We determine the η value (i.e., adaptive weighting) for a user u using the following two methods.

– *Adaptive weights based on scaling (PT-AS)* This method determines the η value for a user u as follows: $\eta = \frac{C_u}{C_{max}}$. In short, we are scaling the POI visit count of a user u by the maximum POI visit count of all users.
– *Adaptive weights based on cumulative distribution (PT-AC)* This method determines the η value for a user u as follows: $\eta = P(C \leq C_u)$. That is, we are building a probability distribution function based on all users' POI visit counts, and then calculating the probability that a random variable C (i.e., the POI visit count) is less than or equal to the POI visit count of user u.

4 Tour recommendation framework

Figure 1 outlines our overall tour recommendation framework. This framework requires a list of POIs (with lat/long coordinates and POI categories) and a set of geo-tagged photographs (with lat/long coordinates and time taken), which can be easily obtained from Wikipedia and Flickr, respectively. Thereafter, the main steps in our framework are:

Step 1: Determine POI visits (map photographs to POIs) We first determine the POI visits in each city by mapping the set of geo-tagged photographs to the list of POIs. In particular, we map a photograph to a POI if their coordinates differ by $<$200m based on the Haversine formula [37], which is used for calculating spherical (earth) distances. If a photograph is within 200m of multiple POIs, we only map this photograph to the nearest POI, i.e., no photograph is mapped to multiple POIs.

Step 2: Construct travel history/sequences Based on the POI visits from Step 1, we can construct the travel history of each user by sorting their POI visits in ascending temporal order (Definition 1). Using each user's travel history, we then proceed to group consecutive POI visits as an individual travel sequence, if the consecutive POI visits differ by $<$8 h (Definition 2). Thus, we are also able to determine the POI visit duration based on the time difference of the first and last photograph taken at each POI.

Step 3: Recommend tours using **PersTour**. As described in Sect. 3.2, there can be different variants of PersTour, based on the value of η and the type of interest function chosen. The value of η indicates the weight given to either POI popularity or user interest, while the interest function can be either frequency-based interest (Int_u^{Freq}), time-based interest (Int_u^{Time}) or time-based interest with weighted updates (Int_u^{Upd}). We experiment with the following variants:

– *PersTour using* $\eta = 0$ $(PT-0)$ PersTour with full emphasis on optimizing POI popularity, ignoring user interest (i.e., no need to choose between Int_u^{Time} or Int_u^{Freq}).
– *PersTour using* $Int^F req_u$ *and* $\eta = 0.5$ $(PT-.5F)$ PersTour with balanced emphasis on optimizing both POI popularity and *frequency-based* user interest.
– *PersTour using* $Int^T ime_u$ *and* $\eta = 0.5$ $(PT-.5T)$ PersTour with balanced emphasis on optimizing both POI popularity and *time-based* user interest.
– *PersTour using* $Int^U pd_u$ *and* $\eta = 0.5$ $(PT-.5U)$ PersTour with balanced emphasis on optimizing both POI popularity and *time-based* user interest with *weighted updates*.

– *PersTour using* Int_u^{Freq} and $\eta = 1$ (PT-1F) PERSTOUR with full emphasis on optimizing *frequency-based* user interest, ignoring POI popularity.
– *PersTour using* $IntTime_u$ *and* $\eta = 1$ $(PT-1T)$ PERSTOUR with full emphasis on optimizing *time-based* user interest, ignoring POI popularity.
– *PersTour using* $IntUpd_u$ *and* $\eta = 1$ $(PT-1U)$ PERSTOUR with full emphasis on optimizing *time-based* user interest with *weighted updates*, ignoring POI popularity.
– *PersTour using* $IntUpd_u$ *and adaptive weighting* η by *scaling* $(PT-AS)$ PERSTOUR with emphasis on optimizing both POI popularity and *time-based* user interest with *weighted updates*, where emphasis is based on adaptive weighting by scaling of POI visit counts.
– *PersTour using* $IntUpd_u$ *and adaptive weighting* η *by cumulative distribution* $(PT-AC)$ PERSTOUR with emphasis on optimizing both POI popularity and *time-based* user interest with *weighted updates*, where emphasis is based on adaptive weighting by cumulative distribution of POI visit counts.

These variants allow us to best evaluate the effects of different η values and compare between frequency-based interest and time-based interest (with and without weighted updates). As PT- 0 does not consider user interest, there is no need to choose between time-based or frequency-based user interest. The PT- 0, PT- .5F, PT- .5T, PT- .5U and PT- 1F, PT- 1T, PT- 1U algorithms allow us to investigate the effect of different emphasis on POI popularity and the different types of user interests, i.e., by adjusting the η parameter. These algorithms offer tourists the flexibility to explicitly specify their preference between the two components of POI popularity and user interests. If the tourist prefers to determine this preference automatically, the PT- AS and PT- AC algorithms provide alternatives where this emphasis (i.e., the η parameter) between the two components of POI popularity and user interests can be automatically learned.

5 Experimental methodology

In this section, we elaborate on the experimental dataset, baseline algorithms and evaluation metrics that are used for our experimental evaluation.

5.1 Dataset

For our experiments, we use the Yahoo! Flickr Creative Commons 100M (YFCC100M) dataset [41,48], which consists of 100M Flickr photographs and videos. This dataset also comprises the meta information regarding the photographs, such as the date/time taken, geo-location coordinates and accuracy of these geo-location coordinates. The geo-location accuracy ranges from world level (least accurate) to street level (most accurate).

Using the YFCC100M dataset, we extracted geo-tagged photographs that were taken in ten different cities, namely Toronto, Osaka, Glasgow, Budapest, Perth, Vienna, Delhi, Edinburgh, Tokyo and London. To ensure the best accuracy and generalizability of our results, we only chose photographs with the highest geo-location accuracy and experimented on ten touristic cities around the world. A more detailed description of our dataset is shown in Table 1. This dataset is also publicly available at https://sites.google.com/site/limkwanhui/datacode#ijcai15.

5.2 Baseline algorithms

We compare our PERSTOUR algorithms against five different baseline algorithms, which can be divided into two broad categories. The first category is based on the popular collaborative

Table 1 Dataset description

City	No. of photographs	No. of users	# POI visits	# travel sequences
Toronto	157,505	1395	39,419	6057
Osaka	392,420	450	7747	1115
Glasgow	29,019	601	11,434	2227
Budapest	36,000	935	18,513	2361
Perth	18,462	159	3643	716
Vienna	85,149	1155	34,515	3193
Delhi	13,919	279	3993	489
Edinburgh	82,060	1454	33,944	5028
Tokyo	55,364	979	15,622	3798
London	164,812	2963	38,746	8373

filtering (CF) recommender systems [34,51,52], which utilizes a user's (tourist's) rating on the items (POIs) to recommend a set of item for another user based on their user similarities. Based on two definitions of user ratings, we implemented two variations of CF-based baseline algorithms, namely

- *Collaborative filtering based on photographs uploaded* ($CF - Pho$) The user/tourist's rating on each item/POI is based on the number of uploaded photographs of that particular POI he/she has uploaded, i.e., a higher number of uploaded photographs correspond to a higher rating for that POI.
- *Collaborative filtering based on POIs visited* ($CF - Bin$) The user/tourist's rating on each item/POI is based on whether they have visited that particular POI, i.e., a binary rating of 1 (visited) or 0 (not visited).

As CF-based algorithms recommend the top-K individual POIs instead of an itinerary of connected POIs, we implemented additional processing steps to ensure a consistent output result for our tour recommendation problem. Based on a starting POI p_1 (like our PersTour algorithm), the CF- Pho and CF- Bin algorithms will iteratively add in the highest ranked POI from the top-K results, until either: (i) the budget B is exhausted or (ii) the destination POI p_N is reached.

The second category of baseline algorithms is variations of greedy-based approaches that have also been used in other tour recommendation research [4,6,46]. Similar to our PersTour approach, these baseline algorithms commence from a starting POI p_1 and iteratively choose a next POI to visit until either: (i) the budget B is exhausted; or (ii) the destination POI p_N is reached. The sequence of selected POIs thus forms the recommended itinerary, and the three greedy-based baselines are:

- *Greedy nearest* ($GNear$) Chooses the next POI to visit by randomly selecting from the three *nearest*, unvisited POIs.
- *Greedy most popular* ($GPop$) Chooses the next POI to visit by randomly selecting from the three *most popular*, unvisited POIs.
- *Random selection* ($Rand$) Chooses the next POI to visit by *randomly selecting* from all unvisited POIs.

GNear and GPop are meant to reflect tourists' behavior by visiting nearby and popular POIs, respectively, while Rand shows the performance of a random-based approach.

5.3 Evaluation

We evaluate PERSTOUR and the baselines using leave-one-out cross-validation [18], i.e., when evaluating a specific travel sequence of a user, we use this user's other travel sequences for training our algorithms. Specifically, we consider all real-life travel sequences with $\geq$3 POI visits and evaluate the algorithms using the starting POIs and destination POIs of these travel sequences. Thereafter, we evaluate the performance of each algorithm based on the recommended tour itinerary I using the following metrics:[5]

1. *Tour recall*: $T_R(I)$ The proportion of POIs in a user's real-life travel sequence that were also recommended in itinerary I. Let P_r be the set of POIs recommended in itinerary I and P_v be the set of POIs visited in the real-life travel sequence, tour recall is defined as: $T_R(I) = \frac{|P_r \cap P_v|}{|P_v|}$.
2. *Tour precision*: $T_P(I)$ The proportion of POIs recommended in itinerary I that were also in a user's real-life travel sequence. Let P_r be the set of POIs recommended in itinerary I and P_v be the set of POIs visited in the real-life travel sequence, tour precision is defined as: $T_P(I) = \frac{|P_r \cap P_v|}{|P_r|}$.
3. *Tour*: F_1-*score* $T_{F_1}(I)$ The harmonic mean of both the recall and precision of a recommended tour itinerary I defined as: $T_{F_1}(I) = \frac{2 \times T_P(I) \times T_R(I)}{T_P(I) + T_R(I)}$.
4. *Root-mean-square error (RMSE) of POI visit duration*: $T_{RMSE}(I)$ The level of error between our recommended POI visit durations in itinerary I compared to the real-life POI visit durations taken by the users. Let $I^s \in I$ be the recommended POIs that were visited in real life,[6] and D_r and D_v be the recommended and real-life POI visit durations, respectively, RMSE is defined as: $T_{RMSE}(I) = \sqrt{\frac{\sum_{p \in I^s} (D_r - D_v)^2}{|I^s|}}$.
5. *Tour popularity:* $T_{Pop}(I)$ The overall popularity of all POIs in the recommended itinerary I defined as: $T_{Pop}(I) = \sum\limits_{p \in I} Pop(p)$.
6. *Tour interest:* $T^u_{Int}(I)$ The overall interest of all POIs in the recommended itinerary I to a user u defined as: $T^u_{Int}(I) = \sum\limits_{p \in I} Int_u(Cat_p)$.
7. *Popularity and interest rank:* T^a_{Rk} The average rank of an algorithm a based on its T_{Pop} and T_{Int} scores ranked against other algorithms (1 = best, 12 = worst).

We selected these metrics to better evaluate the following: (i) time-based versus frequency-based user interest, using Metrics 1–3; (ii) personalized versus non-personalized POI visit durations, using Metric 4; and (iii) PERSTOUR versus baselines, using Metrics 5–7. As personalized POI visit durations only apply to PERSTOUR and not the baselines, we only report T_{RMSE} scores for the PT- 0, PT- .5F, PT- .5T, PT- .5U, PT- 1F, PT- 1T and PT- 1U algorithms. Our baseline for comparing T_{RMSE} is variants of PERSTOUR that use non-personalized POI visit durations, i.e., average POI visit durations.

6 Results and discussion

In this section, we discuss our experimental results and highlight our main findings.

5 Some metrics are rounded off to the same value, but are different values before rounding. The bold-faced values indicate the best performing metrics.

6 We can only compare POI visit durations for POIs in itinerary I that were "correctly" recommended (i.e., visited in real life).

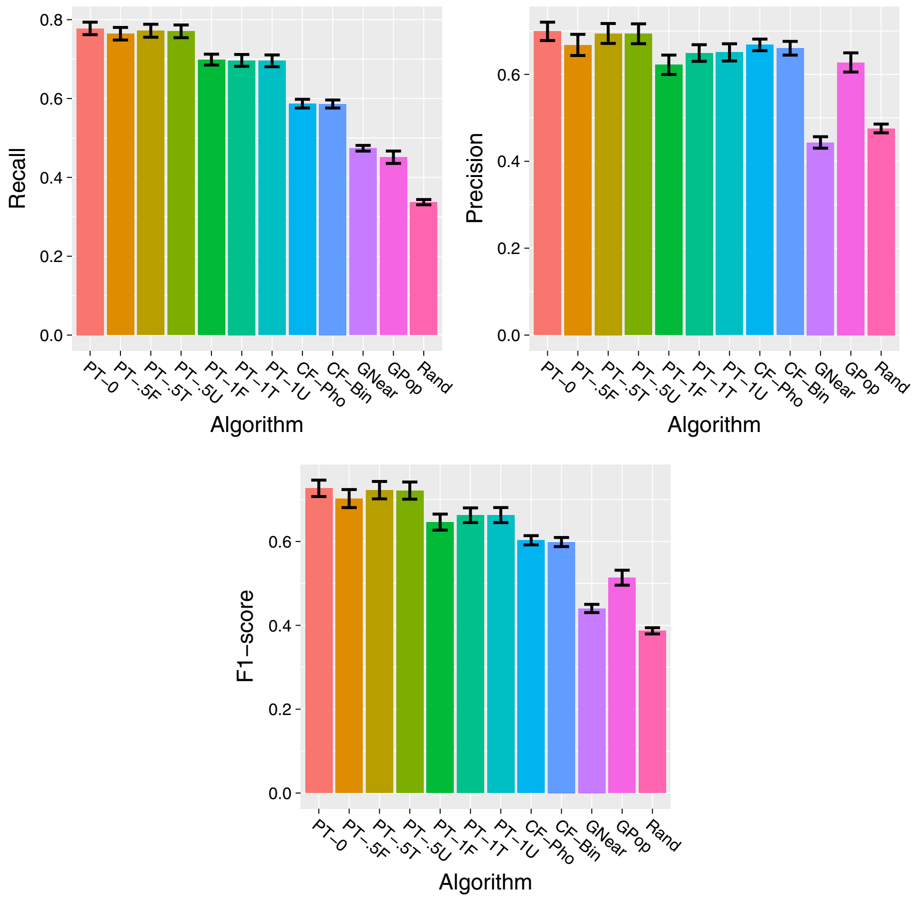

Fig. 2 Overview of results (average scores) across all ten cities, in terms of recall (T_R), precision (T_P) and F_1-score (T_{F_1})

6.1 Comparison between time-based and frequency-based user interest

Figure 2 presents an overview of results in terms of the average recall (T_R), precision (T_P) and F_1-score (T_{F_1}) across all ten cities, for the different variations of our PERSTOUR algorithm and the baselines. The results show that all variants of PERSTOUR out-perform the five baselines in terms of T_R and T_{F_1} scores. In terms of T_P scores, the PERSTOUR variants of PT- 0, PT- .5T, PT- .5U and PT- .5F out-perform all baselines, while the CF- PHO and CF- BIN out-perform the PERSTOUR variants of PT- 1T, PT- 1U and PT- 1F. We now examine the performance of our PERSTOUR algorithm and the baselines in greater detail.

Moving on to the results for individual cities, we study the performance difference between using time-based user interest and frequency-based user interest, as shown in Tables 2 and 3. Comparing the T_{F_1} scores between PT- .5T, PT- .5U and PT- .5F, and between PT- 1T, PT- 1U and PT- 1F, the results show that PERSTOUR using time-based user interest (PT- .5T, PT- .5U, PT- 1T and PT- 1U) out-performs its counterpart that uses frequency-based user interest (PT- .5F and PT- 1F), in most cases. This observation highlights the effectiveness of

Table 2 Comparison between time-based user interest (PT- .5T and PT- 1T) and frequency-based user interest (PT- .5F and PT- 1F), in terms of recall (T_R), precision (T_P) and F$_1$-score (T_{F_1})

Algo.	*Recall*	*Precision*	*F_1-score*	*Algo.*	*Recall*	*Precision*	*F_1-score*
Toronto				*Osaka*			
PT- .5F	.760 ± .009	.679 ± .013	.708 ± .012	PT- .5F	.757 ± .025	.645 ± .037	.687 ± .032
PT- .5T	***.779 ± .010***	***.706 ± .013***	***.732 ± .012***	PT- .5T	***.759 ± .026***	***.662 ± .037***	***.699 ± .033***
PT- .5U	.773 ± .009	.698 ± .013	.726 ± .011	PT- .5U	***.759 ± .026***	***.662 ± .037***	***.699 ± .033***
PT- 1F	.737 ± .010	.682 ± .013	.698 ± .012	PT- 1F	.679 ± .023	.582 ± .032	.616 ± .027
PT- 1T	***.744 ± .011***	***.710 ± .013***	***.718 ± .012***	PT- 1T	***.683 ± .025***	***.622 ± .032***	***.641 ± .029***
PT- 1U	.743 ± .011	.710 ± .013	.718 ± .012	PT- 1U	***.683 ± .025***	***.622 ± .032***	***.641 ± .029***
CF- PHO	.593 ± .007	.650 ± .009	.605 ± .006	CF- PHO	.618 ± .018	.707 ± .034	.635 ± .018
CF- BIN	.589 ± .007	.682 ± .008	.619 ± .006	CF- BIN	.618 ± .018	.736 ± .031	.652 ± .017
GNEAR	.501 ± .010	.512 ± .015	.487 ± .011	GNEAR	.478 ± .026	.433 ± .038	.441 ± .030
GPOP	.440 ± .009	.623 ± .015	.504 ± .011	GPOP	.439 ± .034	.649 ± .038	.517 ± .035
RAND	.333 ± .007	.495 ± .011	.391 ± .009	RAND	.354 ± .021	.488 ± .032	.406 ± .024
Glasgow				*Edinburgh*			
PT- .5F	.819 ± .017	.758 ± .024	.780 ± .021	PT- .5F	.740 ± .006	.607 ± .010	.654 ± .009
PT- .5T	.826 ± .017	.782 ± .022	.798 ± .020	PT- .5T	.740 ± .007	.633 ± .010	.671 ± .008
PT- .5U	***.829 ± .017***	***.783 ± .022***	***.800 ± .020***	PT- .5U	***.743 ± .007***	***.635 ± .010***	***.674 ± .009***
PT- 1F	***.748 ± .017***	.728 ± .022	.726 ± .019	PT- 1F	***.678 ± .007***	.572 ± .009	.605 ± .008
PT- 1T	.739 ± .018	.736 ± .021	.728 ± .019	PT- 1T	.668 ± .007	.601 ± .009	.618 ± .008
PT- 1U	.739 ± .018	***.739 ± .021***	***.730 ± .019***	PT- 1U	.671 ± .007	***.602 ± .009***	***.621 ± .008***
CF- PHO	.600 ± .010	.720 ± .021	.631 ± .011	CF- PHO	.561 ± .006	.648 ± .007	.581 ± .005
CF- BIN	.604 ± .011	.709 ± .022	.627 ± .011	CF- BIN	.567 ± .006	.637 ± .008	.580 ± .005
GNEAR	.498 ± .020	.519 ± .028	.490 ± .022	GNEAR	.471 ± .007	.429 ± .010	.427 ± .008
GPOP	.418 ± .015	.592 ± .024	.480 ± .017	GPOP	.486 ± .008	.640 ± .010	.539 ± .008

Table 2 continued

Algo.	*Recall*	*Precision*	F_1*-score*	*Algo.*	*Recall*	*Precision*	F_1*-score*
RAND	.340 ± .012	.462 ± .017	.386 ± .013	RAND	.336 ± .006	.479 ± .009	.384 ± .006
Budapest				*Perth*			
PT- .5F	.679 ± .008	.550 ± .011	.596 ± .010	PT- .5F	.798 ± .030	.697 ± .045	.735 ± .039
PT- .5T	***.688 ± .008***	***.587 ± .011***	***.624 ± .009***	PT- .5T	***.809 ± .029***	***.725 ± .043***	***.757 ± .037***
PT- .5U	.688 ± .008	.586 ± .011	.623 ± .009	PT- .5U	.792 ± .024	.723 ± .032	.749 ± .028
PT- 1F	***.633 ± .008***	.526 ± .010	.562 ± .009	PT- 1F	.746 ± .032	.660 ± .043	.691 ± .038
PT- 1T	.624 ± .009	***.571 ± .010***	***.587 ± .009***	PT- 1T	***.746 ± .030***	.674 ± .040	.699 ± .035
PT- 1U	.620 ± .009	.569 ± .010	.584 ± .009	PT- 1U	.736 ± .024	***.685 ± .030***	***.702 ± .027***
CF- PHO	.542 ± .007	.598 ± .008	.550 ± .006	CF- PHO	.612 ± .022	.681 ± .024	.634 ± .019
CF- BIN	.558 ± .007	.589 ± .008	.554 ± .006	CF- BIN	.605 ± .017	.621 ± .026	.601 ± .016
GNEAR	.434 ± .007	.403 ± .011	.398 ± .008	GNEAR	.463 ± .030	.432 ± .047	.428 ± .035
GPOP	.408 ± .007	.538 ± .011	.450 ± .008	GPOP	.427 ± .029	.561 ± .038	.477 ± .031
RAND	.300 ± .005	.442 ± .009	.349 ± .006	RAND	.365 ± .024	.543 ± .039	.428 ± .028

The bold italic values refer to the best performing values among its group of comparison algorithms

Table 3 Comparison between time-based user interest (PT- .5T and PT- 1T) and frequency-based user interest (PT- .5F and PT- 1F), in terms of recall (T_R), precision (T_P) and F1-score (T_{F_1})

Algo.	*Recall*	*Precision*	*F_1-score*	*Algo.*	*Recall*	*Precision*	*F_1-score*
Vienna				*Delhi*			
PT- .5F	.711 ± .008	.600 ± .011	.636 ± .010	PT- .5F	.800 ± .033	.718 ± .045	.748 ± .040
PT- .5T	.713 ± .009	.630 ± .011	.656 ± .010	PT- .5T	.807 ± .036	.746 ± .045	.769 ± .041
PT- .5U	***.714 ± .009***	***.632 ± .011***	***.658 ± .010***	PT- .5U	***.813 ± .035***	***.746 ± .045***	***.770 ± .041***
PT- 1F	***.661 ± .007***	.559 ± .010	.589 ± .008	PT- 1F	.671 ± .031	.611 ± .038	.631 ± .034
PT- 1T	.651 ± .008	***.596 ± .010***	***.610 ± .009***	PT- 1T	.674 ± .032	.632 ± .039	.648 ± .036
PT- 1U	.651 ± .008	.593 ± .010	.607 ± .009	PT- 1U	***.674 ± .032***	***.636 ± .041***	***.648 ± .036***
CF- PHO	.523 ± .007	.656 ± .009	.561 ± .006	CF- PHO	.598 ± .027	.711 ± .048	.611 ± .026
CF- BIN	.515 ± .007	.661 ± .009	.559 ± .006	CF- BIN	.593 ± .028	.700 ± .049	.603 ± .027
GNEAR	.469 ± .007	.429 ± .011	.426 ± .008	GNEAR	.506 ± .028	.422 ± .038	.449 ± .031
GPOP	.404 ± .008	.584 ± .012	.465 ± .009	GPOP	.544 ± .032	.773 ± .039	.624 ± .032
RAND	.309 ± .006	.461 ± .010	.361 ± .007	RAND	.327 ± .020	.433 ± .026	.368 ± .021
Tokyo				*London*			
PT- .5F	.842 ± .014	.799 ± .018	.815 ± .016	PT- .5F	.737 ± .006	.625 ± .009	.664 ± .008
PT- .5T	***.852 ± .014***	***.813 ± .017***	***.828 ± .016***	PT- .5T	***.746 ± .007***	***.658 ± .009***	***.690 ± .008***
PT- .5U	.849 ± .014	.813 ± .018	.826 ± .016	PT- .5U	.744 ± .007	.657 ± .009	.688 ± .008
PT- 1F	.755 ± .014	.720 ± .017	.732 ± .016	PT- 1F	.679 ± .006	.581 ± .008	.612 ± .007
PT- 1T	.763 ± .015	.736 ± .017	.745 ± .016	PT- 1T	***.675 ± .007***	***.614 ± .008***	***.632 ± .007***
PT- 1U	***.765 ± .015***	***.739 ± .018***	***.747 ± .016***	PT- 1U	.674 ± .007	.612 ± .008	.631 ± .007
CF- PHO	.634 ± .012	.696 ± .017	.648 ± .011	CF- PHO	.589 ± .005	.612 ± .008	.573 ± .005
CF- BIN	.624 ± .009	.677 ± .017	.632 ± .009	CF- BIN	.588 ± .005	.590 ± .009	.559 ± .005
GNEAR	.469 ± .015	.459 ± .021	.454 ± .017	GNEAR	.450 ± .006	.396 ± .009	.402 ± .007
GPOP	.524 ± .014	.706 ± .021	.592 ± .016	GPOP	.421 ± .006	.609 ± .009	.488 ± .007
RAND	.355 ± .011	.495 ± .017	.407 ± .012	RAND	.353 ± .005	.458 ± .007	.389 ± .005

The bold italic values refer to the best performing values among its group of comparison algorithms

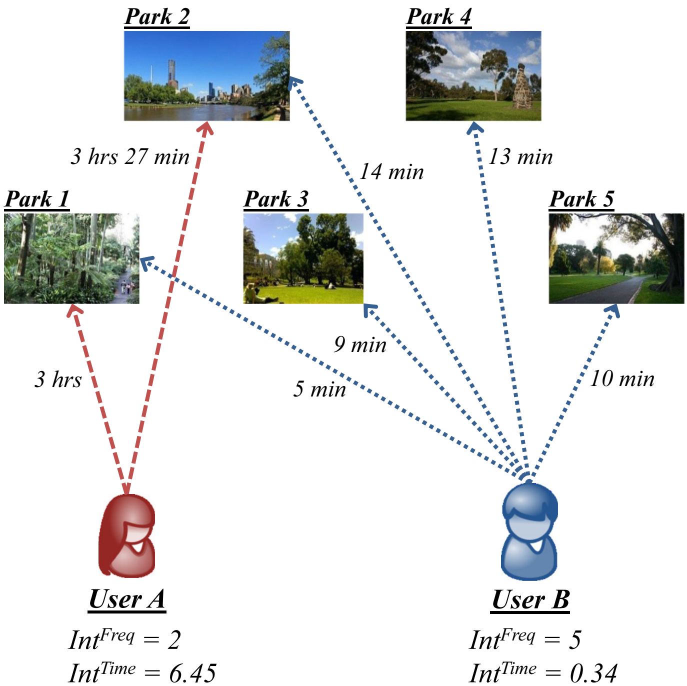

Fig. 3 Toy example illustrating the difference between time-based user interest and frequency-based user interest

time-based user interest in recommending tours that more accurately reflect real-life tours of users, compared to using frequency-based user interest. While PT- 1T and PT- 1U under-perform PT- 1F in terms of T_R for some cities, we focus more on the T_{F_1} scores as it provides a balanced representation of both T_R and T_P. Moreover, for all cities, PT- .5T, PT- .5U, PT- 1T and PT- 1U (time-based user interest) result in higher T_P scores, compared to its PT- .5F and PT- 1F counterparts (frequency-based user interest). Another observation is that all PERSTOUR variants also out-perform the five baselines for all cities, in terms of T_{F_1} scores.

The reason for the more accurate recommendations of time-based user interest compared to frequency-based user interest is due to its use of POI visit durations instead of POI visit frequency. Figure 3 illustrates a toy example that highlights the difference between time-based user interest and frequency-based user interest. Consider user A who only visited two parks but spent three or more hours at each of them and user B who visited five parks but spent less than 15 minutes at each of them. Frequency-based interest incorrectly classifies user B as having more interest in the parks category due to his/her five visits, compared to user A's two visits. On the other hand, time-based interest more accurately determines that user A has a higher interest in the parks category due to his/her long visit duration, despite user A only visiting two parks. Furthermore, time-based interest can more accurately capture a user's level of interest based on how much longer this user spends at a POI compared to the average user (e.g., a user is more interested if he/she spends 3 h at a POI when the average time spent is only 30 minutes). With the availability of user interest levels, we can better personalize POI visit duration for each unique user, which we evaluate next.

6.2 Comparison between personalized and non-personalized visit durations

The T_{RMSE} scores in Tables 4 and 5 show that our recommendation of a personalized POI visit duration (Definitions 5 and 8) out-performs the non-personalized version in 62 out of 70 cases, based on a smaller error in the recommended POI visit durations. This result shows that personalizing POI visit duration using time-based user interests more accurately reflects the real-life POI visit duration of users, compared to the current standard of simply using average POI visit duration. Apart from recommending accurate POIs (T_{F_1} scores), recommending the appropriate amount of time to spend at each POI is another important consideration in tour recommendation, which has not been explored in earlier works that also aim to recommend tour itineraries.

Previously, we have observed how time-based interest results in more accurate POI recommendations based on the T_{F_1} scores. Our personalized POI visit duration builds upon this success by customizing the POI visit duration to each unique user based on his/her relative interest level, i.e., spend more time in a POI that interests the user and less time in a POI that the user is less interested in. Accurate POI visit durations have another important implication in tour recommendation, where spending less time at un-interesting POIs frees up the time budget for more visits to POIs that are more interesting to the user. Similarly, a user might prefer to spend more time visiting a few POIs of great interest, compared to visiting many POIs of less interest to the user.

6.3 Comparison of popularity and interest

We now present an overview of the results in terms of the average popularity (T_{Pop}), interest (T_{Int}) and rank (T_{Rk}) score for all ten cities, as shown in Fig. 4. In particular, we are most interested in the T_{Rk} score that is derived from the average rank of an algorithm based on its T_{Pop} and T_{Int} scores, compared to other algorithms. For T_{Rk} scores, a value of 1 indicates the best performance, while 12 indicates the worst performance. Based on this T_{Rk} score, all variants of PersTour out-perform the five baselines, with PT- .5U being the best performer. We observe a similar performance in terms of T_{Int} scores, where all variants of PersTour out-performing the baselines. In terms of T_{Pop} scores, the PersTour variants of PT- 0, PT- .5T, PT- .5U and PT- .5F out-perform all baselines, while PT- 1T, PT- 1U and PT- 1F under-performs the GPop baseline. This performance is understandable as the PT- 1T, PT- 1U and PT- 1F algorithms emphasize fully on user interest preferences, while the GPop baseline focuses on the most popular POIs thus maximizing the T_{Pop} scores for the latter. Next, we provide a more in-depth discussion of the performance among the various PersTour variants.

Based on the T_{Rk} scores in Tables 6 and 7, we observe that PT- .5U (time-based user interest with weighted updates) is overall the best performer, and PT- .5T (time-based user interest) is the second best performer.[7] In addition, we also observe that PT- 1U (time-based user interest with weighted updates) out-performs its PT- 1F counterpart (frequency-based user interest) for eight out of ten cities, with the same performance for the remaining two cities. These results show the benefits of applying weighted updates to user interests (PT- .5U and PT- 1U), compared to simply using time-based user interest without weighted updates (PT- .5T and PT- 1T).

Next, we examine how PersTour (with and without weighted updates) performs against the various baselines. Both PT- .5U and PT- .5T out-perform all baselines as well as its PT- .5F counterpart that uses frequency-based user interest. Similarly, PT- 1T (time-based user

[7] PT- .5T out-performs PT- .5U in only one city, performs the same in five cities and under-performs in the remaining four cities.

Table 4 Comparison between personalized and non-personalized visit durations, in terms of RMSE (T_{RMSE})

Algo.	*Visit duration*	*RMSE*	*Algo.*	*Visit duration*	*RMSE*
Toronto			*Osaka*		
PT- 0	Personalized	***147.57 ± 10.85***	PT- 0	Personalized	***51.35 ± 11.41***
	Non-personalized	152.44 ± 9.84		Non-personalized	54.91 ± 11.91
PT- .5F	Personalized	***146.33 ± 10.85***	PT- .5F	Personalized	***56.71 ± 12.43***
	Non-personalized	152.61 ± 10.09		Non-personalized	60.06 ± 13.09
PT- .5T	Personalized	***143.56 ± 10.89***	PT- .5T	Personalized	57.09 ± 12.39
	Non-personalized	150.65 ± 10.09		Non-personalized	***55.84 ± 13.18***
PT- .5U	Personalized	***143.75 ± 10.77***	PT- .5U	Personalized	57.69 ± 12.39
	Non-personalized	151.67 ± 10.19		Non-personalized	***55.84 ± 13.18***
PT- 1F	Personalized	***137.07 ± 11.40***	PT- 1F	Personalized	***56.62 ± 13.21***
	Non-personalized	145.54 ± 10.78		Non-personalized	62.24 ± 14.60
PT- 1T	Personalized	***145.20 ± 11.79***	PT- 1T	Personalized	***53.44 ± 13.05***
	Non-personalized	148.18 ± 11.29		Non-personalized	58.88 ± 14.63
PT- 1U	Personalized	***141.53 ± 11.75***	PT- 1U	Personalized	***54.12 ± 13.06***
	Non-personalized	148.64 ± 11.21		Non-personalized	58.88 ± 14.63
Glasgow			*Edinburgh*		
PT- 0	Personalized	***75.98 ± 11.53***	PT- 0	Personalized	***91.08 ± 4.85***
	Non-personalized	85.76 ± 12.07		Non-personalized	113.15 ± 5.21
PT- .5F	Personalized	***88.20 ± 13.03***	PT- .5F	Personalized	***84.56 ± 4.96***
	Non-personalized	92.71 ± 12.92		Non-personalized	99.54 ± 5.14
PT- .5T	Personalized	***76.40 ± 11.34***	PT- .5T	Personalized	***89.76 ± 5.85***
	Non-personalized	90.33 ± 12.35		Non-personalized	100.15 ± 5.27
PT- .5U	Personalized	***77.14 ± 11.52***	PT- .5U	Personalized	***87.19 ± 5.69***
	Non-personalized	90.33 ± 12.35		Non-personalized	101.29 ± 5.30
PT- 1F	Personalized	***79.67 ± 12.27***	PT- 1F	Personalized	***69.61 ± 5.04***
	Non-personalized	86.24 ± 12.85		Non-personalized	78.89 ± 5.31
PT- 1T	Personalized	***73.29 ± 11.94***	PT- 1T	Personalized	***72.11 ± 6.09***
	Non-personalized	91.06 ± 13.45		Non-personalized	74.48 ± 5.29
PT- 1U	Personalized	***74.08 ± 12.14***	PT- 1U	Personalized	***71.54 ± 5.89***
	Non-personalized	90.04 ± 13.44		Non-personalized	78.01 ± 5.41
Budapest			*Perth*		
PT- 0	Personalized	***66.67 ± 5.35***	PT- 0	Personalized	***51.12 ± 15.58***
	Non-personalized	68.32 ± 3.46		Non-personalized	87.03 ± 14.47
PT- .5F	Personalized	***64.79 ± 5.56***	PT- .5F	Personalized	***54.15 ± 16.62***
	Non-personalized	67.36 ± 3.59		Non-personalized	73.23 ± 13.80
PT- .5T	Personalized	***66.40 ± 5.38***	PT- .5T	Personalized	***54.71 ± 16.08***
	Non-personalized	68.61 ± 3.78		Non-personalized	73.78 ± 13.61
PT- .5U	Personalized	***67.51 ± 5.56***	PT- .5U	Personalized	85.80 ± 19.31
	Non-personalized	68.25 ± 3.75		Non-personalized	***69.88 ± 14.57***
PT- 1F	Personalized	***64.61 ± 5.71***	PT- 1F	Personalized	***48.78 ± 16.54***
	Non-personalized	67.79 ± 3.92		Non-personalized	75.46 ± 17.24

Table 4 continued

Algo.	*Visit duration*	*RMSE*	*Algo.*	*Visit duration*	*RMSE*
PT- 1T	Personalized	***68.07*** ± ***5.95***	PT- 1T	Personalized	***52.84*** ± ***16.51***
	Non-personalized	70.55 ± 4.31		Non-personalized	78.74 ± 16.49
PT- 1U	Personalized	***68.84*** ± ***6.28***	PT- 1U	Personalized	85.85 ± 21.75
	Non-personalized	70.32 ± 4.30		Non-personalized	***82.07*** ± ***14.86***

The bold italic values refer to the best performing values among its group of comparison algorithms

Table 5 Comparison between personalized and non-personalized visit durations, in terms of RMSE (T_{RMSE})

Algo.	*Visit duration*	*RMSE*	*Algo.*	*Visit duration*	*RMSE*
Vienna			*Delhi*		
PT- 0	Personalized	***70.71*** ± ***3.64***	PT- 0	Personalized	***29.57*** ± ***6.59***
	Non-personalized	73.81 ± 3.70		Non-personalized	30.60 ± 6.47
PT- .5F	Personalized	***64.87*** ± ***3.24***	PT- .5F	Personalized	***27.58*** ± ***5.73***
	Non-personalized	68.73 ± 3.61		Non-personalized	31.12 ± 6.61
PT- .5T	Personalized	***69.14*** ± ***4.07***	PT- .5T	Personalized	***26.83*** ± ***5.92***
	Non-personalized	70.22 ± 4.55		Non-personalized	33.92 ± 6.83
PT- .5U	Personalized	***69.68*** ± ***4.63***	PT- .5U	Personalized	***27.25*** ± ***5.73***
	Non-personalized	70.19 ± 3.64		Non-personalized	33.92 ± 6.83
PT- 1F	Personalized	***59.92*** ± ***3.88***	PT- 1F	Personalized	***29.83*** ± ***6.85***
	Non-personalized	61.37 ± 4.10		Non-personalized	31.85 ± 7.26
PT- 1T	Personalized	64.64 ± 4.41	PT- 1T	Personalized	***30.02*** ± ***7.06***
	Non-personalized	***62.96*** ± ***4.98***		Non-personalized	35.51 ± 7.76
PT- 1U	Personalized	65.26 ± 5.06	PT- 1U	Personalized	***30.13*** ± ***7.05***
	Non-personalized	***62.99*** ± ***3.87***		Non-personalized	35.51 ± 7.76
Tokyo			*London*		
PT- 0	Personalized	***130.14*** ± ***14.14***	PT- 0	Personalized	***24.67*** ± ***1.80***
	Non-personalized	142.51 ± 10.22		Non-personalized	27.10 ± 1.84
PT- .5F	Personalized	***117.78*** ± ***10.19***	PT- .5F	Personalized	***25.08*** ± ***1.86***
	Non-personalized	146.38 ± 10.22		Non-personalized	26.64 ± 1.91
PT- .5T	Personalized	***127.01*** ± ***13.85***	PT- .5T	Personalized	***25.56*** ± ***1.88***
	Non-personalized	144.51 ± 10.36		Non-personalized	26.91 ± 1.98
PT- .5U	Personalized	***130.25*** ± ***14.07***	PT- .5U	Personalized	***25.41*** ± ***1.90***
	Non-personalized	146.27 ± 10.29		Non-personalized	26.92 ± 1.98
PT- 1F	Personalized	***112.26*** ± ***10.05***	PT- 1F	Personalized	***24.19*** ± ***1.94***
	Non-personalized	144.63 ± 10.52		Non-personalized	25.19 ± 2.00
PT- 1T	Personalized	***100.93*** ± ***9.20***	PT- 1T	Personalized	25.78 ± 2.16
	Non-personalized	138.26 ± 10.46		Non-personalized	***22.74*** ± ***1.84***
PT- 1U	Personalized	***106.84*** ± ***9.54***	PT- 1U	Personalized	26.27 ± 2.21
	Non-personalized	139.03 ± 10.42		Non-personalized	***22.83*** ± ***1.83***

The bold italic values refer to the best performing values among its group of comparison algorithms

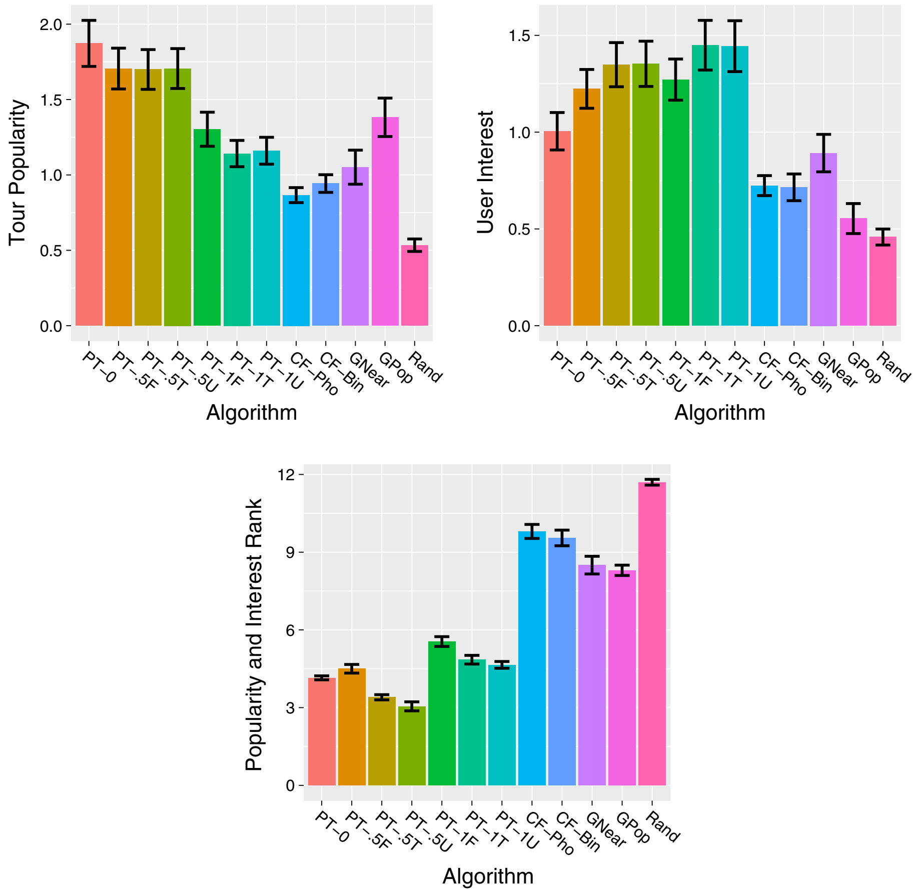

Fig. 4 Overview of results (average scores) across all ten cities, in terms of popularity (T_{Pop}), interest (T_{Int}) and rank (T_{Rk}). *Number within brackets* indicates the rank based on popularity and interest scores, where 1 = best and 12 = worst

interest) out-performs PT- 1F (frequency-based user interest) for six out of ten cities, with the same performance for the remaining four cities. These results show the effectiveness of time-based user interest (both with and without weighted updates) over frequency-based user interest, based on the T_{Rk} scores.

The effects of the η parameter can be observed in the T_{Pop} and T_{Int} scores. A value of $\eta = 0$ (PT- 0) results in the best performance in T_{Pop} and worst performance in T_{Int}, while a value of $\eta = 1$ (PT- 1F, PT- 1T and PT- 1U) results in the opposite. While we include the T_{Pop} and T_{Int} scores for completeness, we are more interested in T_{Rk} as it gives a balanced measurement of both T_{Pop} and T_{Int}.

6.4 Comparison of PersTour with adaptive weights

To evaluate the effectiveness of using adaptive weights, we compare PersTour without adaptive weights (PT- .5U and PT- 1U) against PersTour with adaptive weights (PT- AS and PT- AC). We focus mainly on the top and bottom 15% of users of each city, based on

Table 6 Comparison of PersTour (PT) against baselines, in terms of popularity (T_{Pop}), interest (T_{Int}) and rank (T_{Rk})

Algo.	*Popularity*	*Interest*	*Rk*	*Algo.*	*Popularity*	*Interest*	*Rk*
Toronto				*Osaka*			
PT- 0	2.204 ± .069 (1)	0.904 ± .048 (7)	4	PT- 0	1.263 ± .094 (1)	0.791 ± .166 (8)	4.5
PT- .5F	2.053 ± .063 (2)	1.088 ± .060 (6)	4	PT- .5F	1.126 ± .095 (4)	1.151 ± .213 (5)	4.5
PT- .5T	1.960 ± .064 (4)	1.223 ± .061 (3)	***3.5***	PT- .5T	1.144 ± .093 (3)	1.171 ± .206 (4)	3.5
PT- .5U	1.972 ± .063 (3)	1.195 ± .060 (4)	***3.5***	PT- .5U	1.144 ± .093 (2)	1.176 ± .206 (3)	***2.5***
PT- 1F	1.583 ± .048 (5)	1.137 ± .061 (5)	5	PT- 1F	0.809 ± .075 (8)	1.137 ± .211 (6)	7
PT- 1T	1.419 ± .044 (9)	1.351 ± .069 (1)	5	PT- 1T	0.737 ± .067 (9)	1.205 ± .211 (2)	5.5
PT- 1U	1.420 ± .043 (8)	1.319 ± .069 (2)	5	PT- 1U	0.735 ± .066 (10)	1.207 ± .211 (1)	5.5
CF- Pho	0.926 ± .027 (11)	0.807 ± .042 (8)	9.5	CF- Pho	0.823 ± .078 (7)	0.707 ± .136 (9)	8
CF- Bin	1.121 ± .028 (10)	0.572 ± .033 (10)	10	CF- Bin	0.953 ± .076 (5)	0.661 ± .125 (10)	7.5
GNear	1.424 ± .049 (7)	0.773 ± .054 (9)	8	GNear	0.500 ± .059 (11)	0.853 ± .183 (7)	9
GPop	1.566 ± .050 (6)	0.443 ± .029 (12)	9	GPop	0.837 ± .062 (6)	0.223 ± .066 (12)	9
Rand	0.581 ± .032 (12)	0.467 ± .037 (11)	11.5	Rand	0.433 ± .055 (12)	0.305 ± .089 (11)	11.5
Glasgow				*Edinburgh*			
PT- 0	1.701 ± .101 (1)	0.459 ± .069 (8)	4.5	PT- 0	2.269 ± .046 (1)	1.047 ± .053 (7)	4
PT- .5F	1.562 ± .089 (4)	0.563 ± .091 (5)	4.5	PT- .5F	2.016 ± .042 (2)	1.383 ± .068 (6)	4
PT- .5T	1.601 ± .089 (2)	0.625 ± .084 (4)	***3***	PT- .5T	2.012 ± .043 (3)	1.579 ± .069 (3)	***3***
PT- .5U	1.594 ± .088 (3)	0.626 ± .084 (3)	***3***	PT- .5U	2.003 ± .043 (4)	1.575 ± .070 (4)	4
PT- 1F	1.128 ± .069 (6)	0.562 ± .090 (6)	6	PT- 1F	1.541 ± .038 (6)	1.430 ± .070 (5)	5.5
PT- 1T	1.001 ± .052 (7)	0.676 ± .096 (2)	4.5	PT- 1T	1.336 ± .034 (8)	1.722 ± .076 (1)	4.5
PT- 1U	0.978 ± .050 (8)	0.682 ± .096 (1)	4.5	PT- 1U	1.355 ± .034 (7)	1.720 ± .076 (2)	4.5
CF- Pho	0.914 ± .046 (9)	0.434 ± .059 (9)	9	CF- Pho	0.941 ± .023 (11)	0.752 ± .032 (9)	10
CF- Bin	0.874 ± .045 (10)	0.519 ± .071 (7)	8.5	CF- Bin	1.056 ± .023 (10)	0.740 ± .032 (10)	10
GNear	0.874 ± .064 (11)	0.339 ± .070 (10)	10.5	GNear	1.269 ± .033 (9)	0.939 ± .054 (8)	8.5
GPop	1.399 ± .075 (5)	0.217 ± .049 (12)	8.5	GPop	1.775 ± .039 (5)	0.577 ± .033 (11)	8

Table 6 continued

Algo.	*Popularity*	*Interest*	*Rk*	*Algo.*	*Popularity*	*Interest*	*Rk*
RAND	0.483 ± .048 (12)	0.229 ± .041 (11)	11.5	RAND	0.656 ± .025 (12)	0.526 ± .033 (12)	12
Budapest				*Perth*			
PT- 0	2.921 ± .075 (1)	1.366 ± .075 (7)	4	PT- 0	1.854 ± .154 (1)	1.338 ± .206 (8)	4.5
PT- .5F	2.619 ± .070 (3)	1.596 ± .081 (6)	4.5	PT- .5F	1.732 ± .146 (4)	1.426 ± .209 (7)	5.5
PT- .5T	2.614 ± .069 (4)	1.859 ± .087 (4)	4	PT- .5T	1.744 ± .152 (3)	1.518 ± .209 (4)	3.5
PT- .5U	2.622 ± .069 (2)	1.877 ± .088 (3)	***2.5***	PT- .5U	1.773 ± .127 (2)	1.566 ± .180 (3)	***2.5***
PT- 1F	2.090 ± .064 (6)	1.708 ± .090 (5)	5.5	PT- 1F	1.317 ± .136 (6)	1.490 ± .218 (5)	5.5
PT- 1T	1.687 ± .050 (9)	2.076 ± .091 (2)	5.5	PT- 1T	1.170 ± .131 (8)	1.663 ± .219 (1)	4.5
PT- 1U	1.687 ± .051 (8)	2.109 ± .096 (1)	4.5	PT- 1U	1.313 ± .121 (7)	1.640 ± .189 (2)	4.5
CF- PHO	1.243 ± .026 (11)	0.919 ± .045 (10)	10.5	CF- PHO	0.824 ± .065 (11)	0.923 ± .094 (10)	10.5
CF- BIN	1.309 ± .027 (10)	1.114 ± .051 (9)	9.5	CF- BIN	0.942 ± .071 (10)	0.926 ± .116 (9)	9.5
GNEAR	1.746 ± .057 (7)	1.148 ± .068 (8)	7.5	GNEAR	0.958 ± .115 (9)	1.430 ± .186 (6)	7.5
GPOP	2.209 ± .053 (5)	0.900 ± .050 (11)	8	GPOP	1.401 ± .115 (5)	0.851 ± .115 (11)	8
RAND	0.805 ± .032 (12)	0.572 ± .040 (12)	12	RAND	0.529 ± .077 (12)	0.617 ± .103 (12)	12

Number within brackets indicates the rank based on popularity and interest scores, where 1 = best and 12 = worst

Table 7 Comparison of PERSTOUR (PT) against baselines, in terms of popularity (T_{Pop}), interest (T_{Int}) and rank (T_{Rk})

Algo.	*Popularity*	*Interest*	*Rk*	*Algo.*	*Popularity*	*Interest*	*Rk*
Vienna				*Delhi*			
PT- 0	1.781 ± .045 (1)	1.067 ± .069 (7)	4	PT- 0	1.744 ± .148 (1)	0.628 ± .157 (7)	4
PT- .5F	1.550 ± .043 (4)	1.385 ± .083 (6)	5	PT- .5F	1.620 ± .142 (2)	0.839 ± .208 (6)	4
PT- .5T	1.563 ± .043 (3)	1.559 ± .085 (4)	3.5	PT- .5T	1.610 ± .133 (4)	0.954 ± .252 (3)	***3.5***
PT- .5U	1.571 ± .043 (2)	1.595 ± .086 (3)	***2.5***	PT- .5U	1.620 ± .135 (3)	0.945 ± .248 (4)	***3.5***
PT- 1F	1.234 ± .041 (6)	1.476 ± .088 (5)	5.5	PT- 1F	1.142 ± .119 (6)	0.923 ± .238 (5)	5.5
PT- 1T	1.011 ± .032 (9)	1.676 ± .087 (2)	5.5	PT- 1T	1.129 ± .089 (8)	1.000 ± .257 (1)	4.5
PT- 1U	1.030 ± .033 (8)	1.711 ± .087 (1)	4.5	PT- 1U	1.136 ± .093 (7)	0.964 ± .252 (2)	4.5
CF- PHO	0.676 ± .018 (11)	0.521 ± .031 (11)	11	CF- PHO	0.773 ± .062 (10)	0.605 ± .132 (9)	9.5
CF- BIN	0.687 ± .017 (10)	0.382 ± .023 (12)	11	CF- BIN	0.773 ± .058 (11)	0.618 ± .134 (8)	9.5
GNEAR	1.040 ± .037 (7)	0.957 ± .070 (8)	7.5	GNEAR	1.056 ± .106 (9)	0.524 ± .120 (10)	9.5
GPOP	1.399 ± .037 (5)	0.605 ± .038 (9)	7	GPOP	1.167 ± .102 (5)	0.419 ± .094 (11)	8
RAND	0.470 ± .021 (12)	0.536 ± .040 (10)	11	RAND	0.431 ± .059 (12)	0.356 ± .117 (12)	12
Tokyo				*London*			
PT- 0	1.396 ± .051 (1)	1.256 ± .108 (7)	4	PT- 0	1.592 ± .034 (1)	1.191 ± .055 (7)	4
PT- .5F	1.335 ± .049 (4)	1.379 ± .118 (6)	5	PT- .5F	1.442 ± .032 (2)	1.426 ± .062 (6)	4
PT- .5T	1.341 ± .049 (3)	1.420 ± .114 (3)	***3***	PT- .5T	1.405 ± .031 (4)	1.578 ± .065 (3)	***3.5***
PT- .5U	1.345 ± .048 (2)	1.409 ± .115 (4)	***3***	PT- .5U	1.412 ± .031 (3)	1.567 ± .065 (4)	***3.5***
PT- 1F	1.098 ± .119 (5)	1.388 ± .118 (5)	5	PT- 1F	1.088 ± .029 (5)	1.464 ± .062 (5)	5
PT- 1T	1.023 ± .089 (7)	1.451 ± .119 (1)	4	PT- 1T	0.904 ± .023 (9)	1.672 ± .067 (1)	5
PT- 1U	1.042 ± .051 (6)	1.434 ± .119 (2)	4	PT- 1U	0.909 ± .023 (8)	1.656 ± .066 (2)	5
CF- PHO	0.737 ± .033 (10)	0.725 ± .055 (10)	10	CF- PHO	0.804 ± .015 (10)	0.845 ± .031 (10)	10
CF- BIN	0.948 ± .036 (9)	0.695 ± .058 (11)	10	CF- BIN	0.766 ± .013 (11)	0.922 ± .035 (9)	10

Table 7 continued

Algo.	*Popularity*	*Interest*	*Rk*	*Algo.*	*Popularity*	*Interest*	*Rk*
GNEAR	0.694 ± .044 (11)	0.905 ± .101 (8)	9.5	GNEAR	0.953 ± .026 (7)	1.050 ± .052 (8)	7.5
GPOP	1.006 ± .043 (8)	0.820 ± .077 (9)	8.5	GPOP	1.063 ± .023 (6)	0.481 ± .024 (12)	9
RAND	0.356 ± .028 (12)	0.396 ± .045 (12)	12	RAND	0.597 ± .019 (12)	0.579 ± .030 (11)	11.5

Number within brackets indicates the rank based on popularity and interest scores, where 1 = best and 12 = worst their number of total POI visits. The reason for choosing these users is that adaptive weights are most beneficial to such outlier users as we can recommend more personalized tours to users with many POI visits and popular tours to users with little POI visits.

Our main evaluation metrics are the T_R, T_P and T_{F_1} scores as they indicate the effectiveness of adaptive weights in recommending tours that correspond to real-life visits. Table 8 shows that PT- AS has the overall best performance as indicated by the highest T_R, T_P and T_{F_1} scores for seven, five and six cities, respectively, out of all ten cities. These results show the effectiveness of implementing adaptive weights for different users, i.e., a different level of emphasis between POI popularity and user interest preferences for different users.

7 Conclusion and future work

We modeled our tour recommendation problem based on the Orienteering problem and proposed the PERSTOUR algorithm for recommending personalized tours. Our PERSTOUR algorithm considers both POI popularity and user interest preferences to recommend suitable POIs to visit and the amount of time to spend at each POI. In addition, we implemented a framework where geo-tagged photographs can be used to automatically detect real-life travel sequences and determine POI popularity and user interest, which can then be used to train our PERSTOUR algorithm. Our work improves upon earlier tour recommendation research in three main ways: (i) we introduce *time-based user interest* derived from a user's visit durations at specific POIs relative to other users, instead of using a frequency-based user interest based on POI visit frequency; (ii) we *personalize POI visit duration* based on the relative interest levels of individual users, instead of using the average POI visit duration for all users or not considering POI visit duration at all; and (iii) we introduce two adaptive weighting methods to automatically determine the emphasis on POI popularity and user interest preferences.

Using a Flickr dataset across ten cities, we evaluate the effectiveness of our PERSTOUR algorithm against various collaborative filtering and greedy-based baselines, in terms of tour popularity, interest, precision, recall, F_1-score and RMSE of visit duration. In particular, our experimental results show that: (i) using time-based user interest results in tours that more accurately reflect the real-life travel sequences of users, compared to using frequency-based user interest, based on precision and F_1-score; (ii) our personalized POI visit duration more accurately reflects the time users spend at POIs in real-life, compared to the current standard of using average visit duration, based on the RMSE of visit duration; (iii) PERSTOUR and its variants out-perform all baselines in most cases, based on tour popularity, interest, precision, recall and F_1-score; and (iv) our adaptive weighting methods further improve the performance of PERSTOUR, based on precision, recall and F_1-score.

In this work, we focused mainly on recommending tours that are personalized to individual users based on their *time-based user interest*. Some possible directions for future work are:

- Modeling uncertainty in POI visit duration based on the day of the week and time of the day. The main consideration for this work is to incorporate some uncertainty in the amount of time recommended at various POIs due to delays caused by crowds (e.g., POIs are more crowded during weekends than weekdays, thus causing possible delays)
- Recommending tour itineraries that utilize multiple types of transport (e.g., walking, bus, train, taxi, car), instead of a single type of transport. The main motivation of this future work would be to offer users the flexibility to switch between different modes of transport, while excluding certain types (e.g., either bus, train or taxi but no walking).

Table 8 Comparison between PERSTOUR with weighted updates and PERSTOUR with adaptive weightings, in terms of recall (T_R), precision (T_P) and F$_1$-score (T_{F_1})

Algo.	*Recall*	*Precision*	*F_1-score*	*Algo.*	*Recall*	*Precision*	*F_1-score*
Toronto				*Osaka*			
PT- .5U	***.779 ± .013***	.698 ± .017	***.728 ± .015***	PT- .5U	.765 ± .034	.654 ± .056	.694 ± .047
PT- 1U	.744 ± .014	***.707 ± .017***	.716 ± .015	PT- 1U	.706 ± .035	.617 ± .048	.648 ± .042
PT- AS	.767 ± .012	.685 ± .017	.715 ± .015	PT- AS	***.765 ± .034***	***.667 ± .054***	***.702 ± .046***
PT- AC	.766 ± .013	.700 ± .017	.723 ± .015	PT- AC	.746 ± .038	.654 ± .056	.684 ± .048
Glasgow				*Edinburgh*			
PT- .5U	***.837 ± .026***	***.781 ± .036***	***.802 ± .032***	PT- .5U	.722 ± .010	.583 ± .013	.634 ± .012
PT- 1U	.732 ± .027	.718 ± .032	.715 ± .029	PT- 1U	.682 ± .010	.566 ± .012	.606 ± .011
PT- AS	.831 ± .025	.767 ± .036	.789 ± .032	PT- AS	***.736 ± .010***	***.595 ± .014***	***.646 ± .012***
PT- AC	.831 ± .026	.775 ± .035	.796 ± .031	PT- AC	.723 ± .010	.592 ± .014	.640 ± .012
Budapest				*Perth*			
PT- .5U	.695 ± .014	.573 ± .018	.617 ± .016	PT- .5U	.756 ± .027	.670 ± .037	.703 ± .032
PT- 1U	.606 ± .014	.549 ± .016	.568 ± .015	PT- 1U	.732 ± .026	.660 ± .033	.687 ± .029
PT- AS	***.696 ± .013***	.574 ± .018	***.619 ± .016***	PT- AS	***.777 ± .028***	***.695 ± .038***	***.726 ± .033***
PT- AC	.664 ± .015	***.579 ± .018***	.610 ± .016	PT- AC	.748 ± .027	.667 ± .035	.699 ± .031
Vienna				*Delhi*			
PT- .5U	.742 ± .012	.630 ± .017	.670 ± .015	PT- .5U	.750 ± .056	.639 ± .073	.677 ± .065
PT- 1U	.663 ± .012	.591 ± .015	.614 ± .013	PT- 1U	.665 ± .056	.600 ± .072	.624 ± .066
PT- AS	***.744 ± .012***	.628 ± .017	.668 ± .015	PT- AS	***.771 ± .058***	***.656 ± .078***	***.694 ± .070***
PT- AC	.730 ± .013	***.645 ± .017***	***.674 ± .015***	PT- AC	.722 ± .057	.637 ± .075	.670 ± .068
Tokyo				*London*			
PT- .5U	***.812 ± .021***	.758 ± .029	.777 ± .025	PT- .5U	.714 ± .009	***.602 ± .012***	***.643 ± .011***
PT- 1U	.758 ± .023	.717 ± .028	.732 ± .025	PT- 1U	.675 ± .009	.589 ± .011	.618 ± .010
PT- AS	.807 ± .021	.753 ± .028	.773 ± .025	PT- AS	***.718 ± .009***	.597 ± .012	.641 ± .011
PT- AC	.808 ± .022	***.764 ± .029***	***.779 ± .026***	PT- AC	.704 ± .009	.600 ± .013	.639 ± .011

The bold italic values refer to the best performing values among its group of comparison algorithms

– When using public transport (e.g., bus, train, tram), recommend tour itineraries that consider the arrival and departure times of public transport to minimize the waiting time by the tourists for their respective public transport to arrive. Furthermore, we can also model uncertainty in the arrival times, especially when there are connections between multiple transport modes.

Acknowledgements This work was supported in part by Data61. We thank the anonymous reviewers for their useful comments and suggestions.

References

1. Anagnostopoulos A, Atassi R, Becchetti L, Fazzone A, Silvestri F (2016) Tour recommendation for groups. Data Min Knowl Discov 1–32. doi:10.1007/s10618-016-0477-710
2. Baraglia R, Muntean CI, Nardini FM, Silvestri F (2013) Learnext: learning to predict tourists movements. In: Proceedings of the 22nd ACM international conference on information and knowledge management (CIKM'13), pp 751–756
3. Berkelaar M, Eikland K, Notebaert P (2004) lpsolve: open source (mixed-integer) linear programming system. http://lpsolve.sourceforge.net/
4. Brilhante I, Macedo JA, Nardini FM, Perego R, Renso C (2013) Where shall we go today? Planning touristic tours with TripBuilder. In: Proceedings of the 22nd ACM international conference on information and knowledge management (CIKM'13), pp 757–762
5. Brilhante I, Macedo JA, Nardini FM, Perego R, Renso C (2014) Tripbuilder: a tool for recommending sightseeing tours. In: Proceedings of the 36th European conference on information retrieval (ECIR'14), pp 771–774
6. Brilhante IR, Macedo JA, Nardini FM, Perego R, Renso C (2015) On planning sightseeing tours with tripbuilder. Inf Process Manag 51(2):1–15
7. Castillo L, Armengol E, Onaindía E, Sebastiá L, González-Boticario J, Rodríguez A, Fernández S, Arias JD, Borrajo D (2008) SAMAP: an user-oriented adaptive system for planning tourist visits. Expert Syst Appl 34(2):1318–1332
8. Chen C, Zhang D, Guo B, Ma X, Pan G, Wu Z (2015) TripPlanner: personalized trip planning leveraging heterogeneous crowdsourced digital footprints. IEEE Trans Intell Transp Syst 16(3):1259–1273
9. Choudhury MD, Feldman M, Amer-Yahia S, Golbandi N, Lempel R, Yu C (2010) Automatic construction of travel itineraries using social breadcrumbs. In: Proceedings of the 21st ACM conference on hypertext and hypermedia (HT'10), pp 35–44
10. Cohen R, Katzir L (2008) The generalized maximum coverage problem. Inf Process Lett 108(1):15–22
11. Crandall DJ, Backstrom L, Cosley D, Suri S, Huttenlocher D, Kleinberg J (2010) Inferring social ties from geographic coincidences. Proc Natl Acad Sci 107(52):22436–22441
12. Gavalas Damianos, Charalampos Konstantopoulos KMGP (2014) A survey on algorithmic approaches for solving tourist trip design problems. J Heuristics 20(3):291–328
13. Gionis A, Lappas T, Pelechrinis K, Terzi E (2014) Customized tour recommendations in urban areas. In: Proceedings of the 7th ACM international conference on web search and data mining (WSDM'14), pp 313–322
14. Gunawan A, Lau HC, Vansteenwegen P (2016) Orienteering problem: a survey of recent variants, solution approaches and applications. Eur J Oper Res 255(2):315–332
15. Ji R, Xie X, Yao H, Ma W-Y (2009) Mining city landmarks from blogs by graph modeling. In: Proceedings of the 17th ACM international conference on multimedia (MM'09), pp 105–114
16. Kisilevich S, Mansmann F, Keim D (2010) P-dbscan: a density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos. In: Proceedings of the 1st international conference and exhibition on computing for geospatial research and application (COMGeo'10), p 38
17. Kofler C, Caballero L, Menendez M, Occhialini V, Larson M (2011) Near2me: an authentic and personalized social media-based recommender for travel destinations. In: Proceedings of the 3rd ACM SIGMM international workshop on social media (WSM'11), pp 47–52
18. Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the 14th international joint conference on artificial intelligence (IJCAI'95), pp 1137–1145

19. Kurashima T, Iwata T, Irie G, Fujimura K (2010) Travel route recommendation using geotags in photo sharing sites. In: Proceedings of the 19th ACM international conference on information and knowledge management (CIKM'10), pp 579–588
20. Kurashima T, Iwata T, Irie G, Fujimura K (2013) Travel route recommendation using geotagged photos. Knowl Inf Syst 37(1):37–60
21. Leung KW-T, Lee DL, Lee W-C (2011) CLR: a collaborative location recommendation framework based on co-clustering. In: Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval (SIGIR'11), pp 305–314
22. Li J, Qian X, Tang YY, Yang L, Mei T (2013) GPS estimation for places of interest from social users' uploaded photos. IEEE Trans Multimed 15(8):2058–2071
23. Li W, Eickhoff C, de Vries AP (2014) Geo-spatial domain expertise in microblogs. In: Proceedings of the 36th European conference on information retrieval (ECIR'14), pp 487–492
24. Lim KH (2015) Recommending tours and places-of-interest based on user interests from geo-tagged photos. In: Proceedings of the 2015 SIGMOD PhD symposium (SIGMOD'15), pp 33–38
25. Lim KH, Chan J, Karunasekera S, Leckie C (2017) Personalized itinerary recommendation with queuing time awareness. In: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval (SIGIR'17)
26. Lim KH, Chan J, Leckie C, Karunasekera S (2015a) Improving location prediction using a social historical model with strict recency context. In: Proceedings of the 5th workshop on context-awareness in retrieval and recommendation (CaRR'15)
27. Lim KH, Chan J, Leckie C, Karunasekera S (2015b) Personalized tour recommendation based on user interests and points of interest visit durations. In: Proceedings of the twenty-fourth international joint conference on artificial intelligence (IJCAI'15), pp 1778–1784
28. Lim KH, Chan J, Leckie C, Karunasekera S (2016) Towards next generation touring: personalized group tours. In: Proceedings of the 26th international conference on automated planning and scheduling (ICAPS'16), pp 412–420
29. Lim KH, Wang X, Chan J, Karunasekera S, Leckie C, Chen Y, Tan CL, Gao FQ, Wee TK (2016) PersTour: a personalized tour recommendation and planning system. In: Extended proceedings of the 27th ACM conference on hypertext and social media (HT'16)
30. Lucchese C, Perego R, Silvestri F, Vahabi H, Venturini R (2012) How random walks can help tourism. In: Proceedings of the 34th European conference on information retrieval (ECIR'12), pp 195–206
31. Miller CE, Tucker AW, Zemlin RA (1960) Integer programming formulation of traveling salesman problems. J ACM 7(4):326–329
32. Popescu A, Grefenstette G, Moëllic P-A (2009) Mining tourist information from user-supplied collections. In: Proceedings of the 18th ACM conference on information and knowledge management (CIKM'09), pp 1713–1716
33. Quercia D, Schifanella R, Aiello LM (2014) The shortest path to happiness: recommending beautiful, quiet, and happy routes in the city. In: Proceedings of the 25th ACM conference on hypertext and social media (HT'14), pp 116–125
34. Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J (1994) Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM conference on computer supported cooperative work (CSCW'94), pp 175–186
35. Schedl M, Hauger D, Schnitzer D (2012) A model for serendipitous music retrieval, In: Proceedings of the 2nd workshop on context-awareness in retrieval and recommendation (CaRR'12), pp 10–13
36. Shi Y, Serdyukov P, Hanjalic A, Larson M (2011) Personalized landmark recommendation based on geotags from photo sharing sites. In: Proceedings of the fifth international AAAI conference on weblogs and social media (ICWSM'11), pp 622–625
37. Sinnott RW (1984) Virtues of the Haversine. Sky Telesc 68(158):159
38. Souffriau W, Vansteenwegen P (2010) Tourist trip planning functionalities: state-of-the-art and future. In: Proceedings of the 10th international conference on web engineering (ICWE'10), pp 474–485
39. Spyrou E, Mylonas P (2016) A survey on Flickr multimedia research challenges. Eng Appl Artif Intell 51:71–91
40. Sun Y, Fan H, Bakillah M, Zipf A (2015) Road-based travel recommendation using geo-tagged images. Comput Environ Urban Syst 53:110–122
41. Thomee B, Shamma DA, Friedland G, Elizalde B, Ni K, Poland D, Borth D, Li L-J (2016) YFCC100M: the new data in multimedia research. Commun ACM 59(2):64–73
42. Tsiligirides T (1984) Heuristic methods applied to orienteering. J Oper Res Soc 35(9):797–809
43. Vansteenwegen P, Oudheusden DV (2007) The mobile tourist guide: an OR opportunity. OR Insight 20(3):21–27

44. Vansteenwegen P, Souffriau W, Berghe GV, Oudheusden DV (2011) The city trip planner: an expert system for tourists. Expert Syst Appl 38(6):6540–6546
45. Vansteenwegen P, Souffriau W, Oudheusden DV (2011) The orienteering problem: a survey. Eur J Oper Res 209(1):1–10
46. Wang X, Leckie C, Chan J, Lim KH, Vaithianathan T (2016) Improving personalized trip recommendation by avoiding crowds. In: Proceedings of the 25th ACM international conference on information and knowledge management (CIKM'16), pp 25–34
47. Wörndl W, Hefele A (2016) Generating paths through discovered places-of-interests for city trip planning. In: Information and communication technologies in tourism. Springer International Publishing, pp 441–453
48. Yahoo! Webscope (2014) Yahoo! Flickr creative Commons 100M dataset (YFCC-100M). http://webscope.sandbox.yahoo.com/catalog.php?datatype=i&did=67
49. Yamasaki T, Gallagher A, Chen T (2013) Personalized intra-and inter-city travel recommendation using large-scale geotags. In: Proceedings of the 2nd ACM international workshop on geotagging and its applications in multimedia (GeoMM'13), pp 25–30
50. Yao L, Sheng QZ, Qin Y, Wang X, Shemshadi A, He Q (2015) Context-aware point-of-interest recommendation using tensor factorization with social regularization. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval (SIGIR'15), pp 1007–1010
51. Ye M, Yin P, Lee W-C, Lee D-L (2011) Exploiting geographical influence for collaborative point-of-interest recommendation. In: Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval (SIGIR'11), pp 325–334
52. Yuan Q, Cong G, Ma Z, Sun A, Thalmann NM (2013) Time-aware point-of-interest recommendation. In: Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval (SIGIR'13), pp 363–372

Kwan Hui Lim is currently a Ph.D. candidate at the Department of Computing and Information Systems, University of Melbourne, Australia. Previously, he was a Research Engineer at the Living Analytics Research Centre, Singapore Management University. He received his Master of Science (Research) and Bachelor of Computer Science (first-class honors) degrees from the University of Western Australia. He is a recipient of the 2016 Google Ph.D. Fellowship in Machine Learning. His research interests are in data mining, machine learning, social network analysis and social computing.

Jeffrey Chan is currently a Lecturer at the RMIT University, Melbourne, Australia. He has a B.E., B.Sc. and Ph.D., all from the University of Melbourne. He has over 50 publications in graph mining, social network analysis and data mining, and his research interests are in data analytics, analyzing graphs and social networks and learning about new and exciting research.

Christopher Leckie received the B.Sc. degree in 1985, the B.E. degree in electrical and computer systems engineering (with first-class honors) in 1987, and the Ph.D. degree in computer science in 1992, all from Monash University, Clayton, VIC, Australia. He joined Telstra Research Laboratories in 1988, where he conducted research and development into artificial intelligence techniques for various telecommunication applications. In 2000, he joined the University of Melbourne, Parkville, VIC, where he is currently a Professor in the Department of Computing and Information Systems. His research interests include scalable data mining, network intrusion detection, bioinformatics and wireless sensor networks.

Shanika Karunasekera received the B.Sc. degree in electronics and telecommunications engineering from the University of Moratuwa, Sri Lanka, in 1990 and the Ph.D. degree in electrical engineering from the University of Cambridge, Cambridge, UK, in 1995. From 1995 to 2002, she was a Software Engineer and a Distinguished Member of Technical Staff with Lucent Technologies, Bell Labs Innovations, USA. In January 2003, she joined the University of Melbourne, VIC, Australia, and currently she is a Professor in the Department of Computing and Information Systems. Her current research interests include distributed system engineering, distributed data mining and social media analytics.