

Enhancing Personalized Trip Recommendation with Attractive Routes

Jiqing Gu,^{1,3,4} Chao Song,¹ Wenjun Jiang,² Xiaomin Wang,¹ Ming Liu¹

¹School of Computer Science and Engineering, University of Electronic Science and Technology of China

²College of Computer Science and Electronic Engineering, Hunan University

³CETC Big Data Research Institute Co.,Ltd, Guiyang

⁴Big Data Application on Improving Government Governance Capabilities National Engineering Laboratory, Guiyang
gujiqing@std.uestc.edu.cn, {chaosong, csmlu, xmwang}@uestc.edu.cn, jiangwenjun@hnu.edu.cn

Abstract

Personalized trip recommendation tries to recommend a sequence of point of interests (POIs) for a user. Most of existing studies search POIs only according to the popularity of POIs themselves. In fact, the routes among the POIs also have attractions to visitors, and some of these routes have high popularity. We term this kind of route as *Attractive Route (AR)*, which brings extra user experience. In this paper, we study the attractive routes to improve personalized trip recommendation. To deal with the challenges of discovery and evaluation of ARs, we propose a personalized Trip Recommender with POIs and *Attractive Route (TRAR)*. It discovers the attractive routes based on the popularity and the Gini coefficient of POIs, then it utilizes a gravity model in a category space to estimate the rating scores and preferences of the attractive routes. Based on that, TRAR recommends a trip with ARs to maximize user experience and leverage the tradeoff between the time cost and the user experience. The experimental results show the superiority of TRAR compared with other state-of-the-art methods.

1 Introduction

With the rapid development of mobile devices and location acquisition technologies, location-based personalized trip recommender that recommends sequential points of interest (POIs) to visitors¹ has emerged and received popularity recently. Traditional trip recommendation utilizes the popularity and user preference to evaluate the attraction of each POI, and recommends a sequential POIs with maximal user experience (Lim et al. 2017; 2015). However, in addition to POIs, the route between two POIs also has attraction to the visitors. We take Fig.1 as an example to illustrate the attractions from routes. Each capital letter refers to a POI and its number in the brackets indicates the popularity of the POI. The number on each route indicates its popularity. Traditional trip recommender only considers POIs with high popularity to achieve high user experience. Thus, by only considering the popularity of POIs, the path $O \rightarrow A \rightarrow D$ will be recommended, since POI A has the highest popularity. Although the popularity of B is less than that of A , the business events on the two routes $O \rightarrow B$ and $B \rightarrow D$ attract many visitors to increase their popularity. When a user

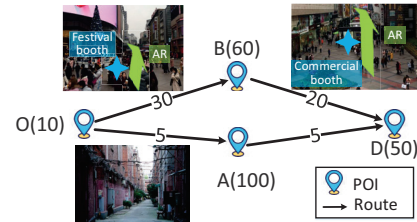


Figure 1: An example of attractive routes.

travels along the two routes, she may also visit these events and her experience will be improved. Thus, by considering the popularity of routes and POIs, the recommended path $O \rightarrow B \rightarrow D$ will bring more user experience than the previous one.

Moreover, by taking the route $B \rightarrow D$ in Fig.1 as an example, it is an incoming route to a POI with high popularity, and its popularity occupies the majority of POI's. Without depending on POI, the route has its own attractions (such as business events) for visitors. We term this kind of route as *Attractive Route (AR)*, which brings extra user experience. However, considering attractive route in personalized trip recommendation has the following challenging issues: (1) How to discover attractive routes? (2) How to evaluate the rating score and the preference of an attractive route, so as to decide if a user will visit an attractive route with extra travel time? (3) How to improve personalized trip recommendation by considering attractive routes?

In this paper, we study the user experience from the routes as an extension of the Orienteering Problem, which aims to maximize the user experience of a recommended trip with multi-constraints. To address these challenges, we propose a personalized Trip Recommender with POIs and *Attractive Route (TRAR)*. It contains AR discovery, AR evaluation, and trip recommendation. Our contributions are as follows:

- As far as we know, we are the first to investigate the attractive routes in personalized trip recommendation. TRAR discovers the attractive routes by the popularity and Gini coefficient of each POI from the travel history of users.
- In the process of AR evaluation, TRAR obtains the preference of each AR by unsupervised learning. Then, TRAR calculates the rating score of each AR by a gravity model

and SVM.

- We propose a personalized trip recommendation algorithm with attractive routes to maximize the user experience along the trip, by estimating whether a user will visit the attractive routes.
- We evaluate the performance of TRAR with real-world datasets from Foursquare, and compare it with some baselines. Experiment results show TRAR is superior to other state-of-the-art recommendation techniques.

2 Related Work

Trip Recommendation. Trip recommendation aims to suggest a sequence of POIs to visit instead of individual POIs, which is similar to our problem. Yoon et al. (Yoon et al. 2012) applied GPS trajectories to seek for a potential path by topic modeling or Markov models. Lim et al. (Lim et al. 2017) recommended personalized tours by using POI popularity and user preferences, which are automatically derived from real-life travel sequences based on geo-tagged photographs. Luan et al. (Luan et al. 2018) proposed a maximal-marginal-relevance-based algorithm in trip planning. Some works (Zhang et al. 2015) maximized user experience based on POI popularity in personalized trip recommendation. (Zhang et al. 2015) found the optimal trip that maximizes user experiences for a given time budget constraint. Lu et al. (Lu, Chen, and Tseng 2012) proposed PTR to recommend the personalized trips meeting multiple constraints by mining visitor’s check-in behaviors. All these works do not consider the attractive routes.

Orienteering problem in operations research. The orienteering problem (OP) (Vansteenwegenaaba 2011) studied in operation research and theoretical computer science is related to our problem. In OP, there is a set of nodes and each of them has a score. The goal is to generate a path, limited in time and the specific start and end nodes, that visits some nodes and maximizes the total collected scores. However, the differences between trip recommendation and OP are: First, OP does not consider the personalized user preference and generates the same trip for all users. Second, OP has no stay time for each location, which is a key factor affecting the total travel time of a trip. Finally, we consider the attractive routes, which is absent in OP.

Gravity and scenic path in recommendation. Another relevant work is Zhang et al. (Zhang and Chow 2015) proposed a gravity model to weigh the effect of each visited location on the new location derived from the attractive force. Javed et al. (Javed, Ghani, and Elmqvist 2012) calculated a gravity vector that models the attention of current position to a visitor. The most relevant work (Quercia, Schifanella, and Aiello 2014) considered routes not only short but also emotionally pleasant in the trip recommendation, the recommended trips are more beautiful, quiet and happy, which enhances users’ experiences; (Lu and Shahabi 2015; Johnson et al. 2017; Lu et al. 2017) considered scenic or safest path while planing a trip.

Our work differs from the aforesaid studies in that we investigate the user experiences from both of routes and POIs in the personalized trip recommendation.

Table 1: Notations.

Notation	Interpretation
$\mathcal{V}, \mathcal{E}, U, \mathcal{R}$	the set of POIs, routes, users, AR
S, L	the set of AR rating score, classifiers
v, e, u	POI in \mathcal{V} , route in \mathcal{E} , user in U
$P(\cdot)/ps(\cdot)$	the function of preference / preference score
$RS(\cdot)$	the rating score
$E(\cdot)$	the function of user experience
$T(u, \cdot)$	the function of time cost of u
$t(e)$	the travel time on e
tp	a sequence of ordered POIs
η_1/η_2	the threshold of POI popularity / Gini coefficient
T_{max}	the time constraint in a trip
δ	the threshold of $E(u, tp)$
$Pop(\cdot)$	the function of popularity
m, n	the number of users and POIs

3 Problem Definition

In this section, we first introduce the system settings and the key concepts. Then we formally define the problem. Table 1 shows the notations used in this paper.

3.1 System Settings and Concepts

Definition 1: Travel Graph. For a region with n POIs, we construct a directed complete weighted travel graph $G = (\mathcal{V}, \mathcal{E})$ with n nodes being the POIs, and $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ is the set of the n POIs. Each edge $e_{ij} \in \mathcal{E}$ represents the route from v_i to v_j where $v_i, v_j \in \mathcal{V}$, and the weight of edge includes the average travel time along it. The set of m users is denoted by $U = \{u_1, u_2, \dots, u_m\}$.

Definition 2: Preference. Each user u and each POI v are associated with a preference $P(u)$ and $P(v)$ in a z dimensional category space, respectively. The preference function is denoted by $P(\cdot) = (\gamma_1, \gamma_2, \dots, \gamma_z)$, where γ_i is in the range of 0 to 1. γ_i in $P(v)$ and $P(u)$ refer to the preference degree of the i^{th} category for POI v and user u , respectively. γ_i in $P(u)$ is calculated by the ratio that the number of visits of the i^{th} category to the total number of visits, which is modeled based on its past visiting records (Li et al. 2017; Jiang, Wu, and Wang 2015).

Definition 3: Preference Score. The user u has a preference score to each POI v denoted by a function $ps(u, v)$, which is calculated by the cosine similarity between $P(u)$ and $P(v)$. In the case of $P(u) = (\alpha_1, \alpha_2, \dots, \alpha_z)$ and $P(v) = (\beta_1, \beta_2, \dots, \beta_z)$, $ps(u, v) = \frac{\sum_{i=1}^z \alpha_i \cdot \beta_i}{\sqrt{\sum_{i=1}^z \alpha_i^2} \cdot \sqrt{\sum_{i=1}^z \beta_i^2}}$.

Definition 4: Trip. A trip is orderly composed of one or several POIs denoted by $tp = (v_{k_1}, v_{k_2}, \dots, v_{k_s})$, which is also denoted as s -trip and s indicates the number of POIs in a trip (i.e., $|tp| = s$). The stay time of a user u costs on a POI v is denoted by a function $T(u, v)$, which is calculated by the historical trips of users (Lim et al. 2015). The travel time of users on e_{ij} is denoted as $t(e_{ij})$, which is estimated by the Euclidean distance and the walking speed of the user (e.g., 4 km/h). The time cost along a trip denoted by a function $T(u, tp)$ is defined as the time of a user u spending on the trip, which is calculated as: $T(u, tp) = \sum_{i=1}^{s-1} \sum_{j=2}^s t(e_{ij}) + \sum_{i=1}^s T(u, v_j)$.

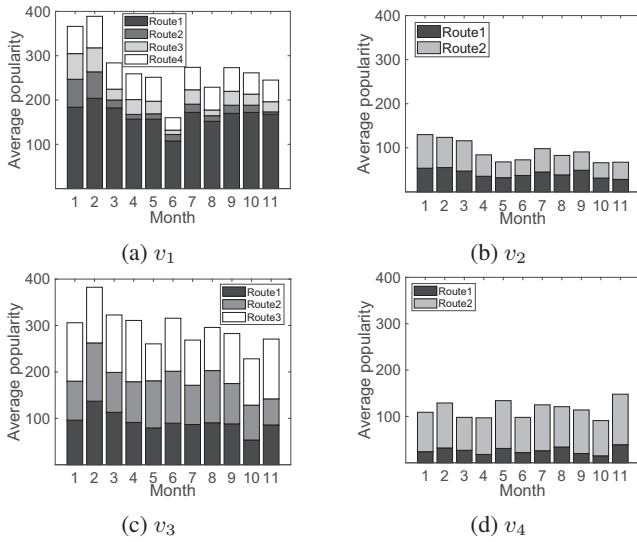


Figure 2: The popularity of incoming routes at four POIs: (a) v_1 has high popularity, more incoming routes with unequal popularity; (b) v_2 has low popularity, less incoming routes with equal popularity; (c) v_3 has high popularity, more incoming routes with equal popularity; (d) v_4 has low popularity, less incoming routes with unequal popularity.

3.2 Attractive Route

The number of visitors at a POI is affected by the attraction from either the POI or the routes to it. Thus, inspired by (Lim et al. 2017), we define the concept of popularity as follows:

Definition 5: Popularity. Popularity is the number of visits of a POI or a route, which is denoted by a function $Pop(\cdot)$. The POI popularity denoted by $Pop(v)$ is estimated by the number of visits of v . The route popularity denoted by $Pop(e)$ is estimated by the number of visitors who have traveled along the route e . The relationship between $Pop(e)$ and $Pop(v)$ is: $Pop(v) = \sum_{e \in \mathcal{E}(v)} Pop(e)$, where $\mathcal{E}(v)$ is the set of incoming routes of v .

We make a statistic of the routes from Foursquare dataset (Luan et al. 2018). For each POI and its incoming routes, we obtain the average number of visits based on their popularity of each month. We take four POIs as an example to illustrate the route popularity and Fig.2 shows the popularity of incoming routes at 4 different POIs. We observe that some POIs has popularity inequality of its incoming routes. Among these incoming routes, the one with the highest popularity has advantages to attract visitors, such as business events, etc. However, traditional trip recommendation neglects such routes and recommends users to visit the incoming route with low popularity and may reduce user experience. Therefore, we introduce a new concept of attractive route as follows:

Definition 6: Attractive Route (AR). The attractive route is an incoming route to a POI with high popularity, and its popularity occupies the majority of POI's. Let \mathcal{R} denote the set of attractive routes and $\mathcal{R} \subset \mathcal{E}$.

We take Fig. 1 as an example to illustrate the effect of AR.

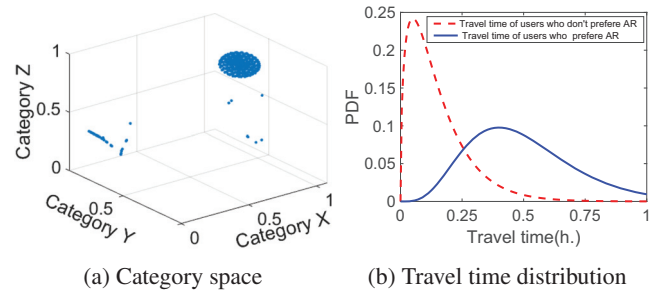


Figure 3: Analysis of AR in Foursquare.

Fig. 1 shows that by considering the popularity of routes and POIs, the path $O \rightarrow B \rightarrow D$ will be recommended to bring more user experience. Many kinds of events can affect the popularity of routes, such as the sales promotions, movie promotions, temporal shows and so on. These events attract visitors, however, they also increase their travel time. Taking Fig.1 for instance, it shows a commercial booth and a festival booth occur on the attractive routes. The green arrow represents the pedestrian flow on the route, and the blue sign represents the business event on the route. Although the events improve user experience, they take longer travel time for visiting. Therefore, we should estimate whether a user will visit an attractive route by considering the extra travel time, which is a challenging issue.

For an attractive route, we extract the preferences of past visitors based on their visiting history from the Foursquare² dataset. A user preference has the following 8 coarse-grained dimensions³: arts & entertainment; college & university; food; great outdoors; home, work, other; nightlife spot; shop; travel spot. We take route1 of v_1 mentioned in Fig.2 as an example to analyze attractive route. We reduce the preferences of 500 users who visit route1 to three dimensions (Category X, Category Y, Category Z) for visualization by t-SNE (Maaten L 2008), and Fig.3(a) shows the distribution of these user preferences in the category space. For the users who visit the attractive route, we find the vast majority of these users' preferences cluster together, which indicates the preferences of these users are similar to each other; some preferences of users are scattered, which indicates these users have different user preference with the clustered users. Then we select 50 attractive routes and analyze the travel time of users on these routes. The users who prefer AR spend more time to on it. Fig.3(b) shows the travel time distribution of users who have visited those attractive routes. It indicates that some of them prefer attractive routes and others not. The users who prefer the attractive routes will cost longer travel time, such as visiting the event of sale promotion on it. The users who don't prefer AR will treat AR as an ordinary route and travel along the AR quickly, so users spend less time on this AR. Thus, the preference of an attractive route is defined as follows:

Definition 7: Preference of AR. Preference of AR is measured by the cluster center of the preferences of users

²<https://foursquare.com/>

³<https://developer.foursquare.com/>

who prefer and visited this AR. Each attractive route $e \in \mathcal{R}$ is associated with a preference $P(e)$. Based on the above analysis, the preference score of e for u denoted by $ps(u, e)$ is also calculated by the cosine similarity between $P(u)$ and $P(e)$. The travel time of u spends on e is denoted as $T(u, e)$.

3.3 Problem Formulation

The objective of personalized trip recommendation is to recommend user trips while maximizing the user experience (UE). Inspired by (Luan et al. 2018), the user experience for u to v is based on the preference between u and v , which is defined as: $E(u, v) = ps(u, v) \cdot RS(v)$, where $RS(v)$ denotes the rating score of v with 1-5 scales obtained from Yelp (Hu and Ester 2013). The user experience for visiting an attractive route e by u is calculated as: $E(u, e) = ps(u, e) \cdot RS(e)$, where $RS(e)$ refers to the rating score of e and it is estimated in Section 4.1. The total user experience of u to tp denoted by $E(u, tp)$ includes the user experience generated from both of POIs and attractive routes as follows:

$$E(u, tp) = \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (E(u, v_j) + E(u, e_{ij})) \quad (1)$$

In Eq.1, if a recommended trip involves traveling from POI v_i to v_j , $x_{ij} = 1$; Otherwise, $x_{ij} = 0$.

Based on the above settings, the problem of personalized trip recommendation with POIs and attractive routes is defined as follows:

Problem 1: Personalized trip recommendation with POIs and Attractive Route (PEAR). Given a travel graph $G = (\mathcal{V}, \mathcal{E})$, a visitor u who requests a trip recommendation has a specific starting location, constraint time T_{max} and a predefined user experience demand δ . We recommend a trip tp such that user experience is maximized meeting four constraints: (1) the trip starts at a specific POI, (2) the recommended trip comprises POIs connected as a trip without loops, (3) the travel time of this trip should be in the constraint traveling time T_{max} , (4) the user experience is larger than the predefined user experience demand δ . Thus, the problem is formally defined as follows:

$$\text{Maximize } E(u, tp) \quad (2)$$

$$s.t. \quad \sum_{i=2}^n x_{1i} = 1 \quad (3)$$

$$\sum_{i=1}^{n-1} x_{ik} = \sum_{j=2}^n x_{kj} \leq 1 \quad (4)$$

$$\sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (T(u, e_{ij}) + T(u, v_j)) \leq T_{max} \quad (5)$$

$$\sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} E(u, v_j) \geq \delta \quad (6)$$

Eq.2 is the objective function merges the user experience from POIs and ARs in the trip. Eq.s 3 – 6 correspond to the four constraints. The recommender generates a trip which consists of a sequence of POIs and ARs.

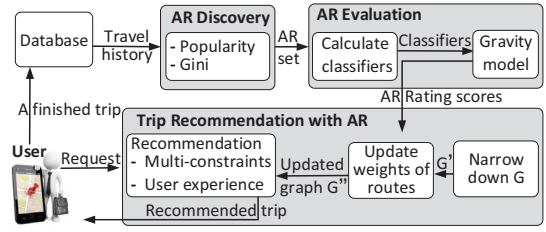


Figure 4: The framework of TRAR.

The hardness. We present the following simplified decision version of the PEAR problem, which is NP-complete, by ignoring the stay time and the predefined demand of user experience. In particular, we assume that $T(u, v) = 0$, $E(u, e_{ij}) = 0$, $\delta = 0$, $E(u, v_j) = 1$ for all POIs $v_j \in \mathcal{V}$ (except the source O and destination D), the source and the destination are the same POI, i.e., $O = D$, and the traveling time cost $T(u, e_{ij})$ is a fixed constant. The simplified decision PEAR problem is: Given a travel graph $G = (\mathcal{V}, \mathcal{E})$, a user u with the source and destination O , and a time budget T_{max} , the task is to decide whether there is any trip tp such that $E(u, tp) \geq \delta$ and the trip is completed within constraint time T_{max} .

Theorem 1. *The simplified decision version of PEAR problem is NP-complete.*

Proof. We reduce the decision version of the traveling salesman problem (TSP) (Applegate et al. 2007), which is known to be NP-hard, to the simplified decision version of PEAR. Since the simplified decision PEAR is NP-complete, the optimization PEAR problem is at least NP-hard. Due to the limited space, we omit the detailed proof. \square

4 Trip Recommendation with ARs

We propose a personalized Trip Recommender with POIs and Attractive Route (TRAR) in this section, which includes AR discovery, AR evaluation and trip recommendation. Fig.4 shows the framework of TRAR.

4.1 AR Discovery

There is no records about ARs in the real dataset, so it is necessary to discover ARs from dataset; We discover ARs to improve the user experience from a recommended trip.

We have discussed the popularity inequality of the incoming routes of a POI in Fig.2. In this section, we introduce Gini coefficient (Barro 2000) to measure such inequality. We define $\vec{p}_j = [p_{1j}, p_{2j}, \dots]$ as a vector with ascending order elements, and each element p_{ij} indicates the popularity proportion of an incoming route e_{ij} calculated by $p_{ij} = \frac{Pop(e_{ij})}{Pop(v_j)}$. Let $Gini(\vec{p}_j)$ denote the Gini coefficient of POI v_j . It is calculated as follows:

$$Gini(\vec{p}_j) = 1 - \frac{1}{|\vec{p}_j|} \left(2 \cdot \sum_{i=1}^{|\vec{p}_j|} \sum_{k=1}^i p_{kj} - 1 \right) \quad (7)$$

As shown in Fig.4, TRAR utilizes the travel history of users such as check-in data from Foursquare to discover attractive

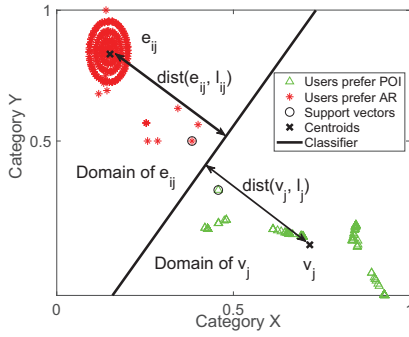


Figure 5: AR evaluation with gravity model

routes. We select one incoming route with the highest popularity for each POI. The process of discovering attractive routes is as follows: We initialize $\mathcal{R} \leftarrow \emptyset$; For each $v_j \in \mathcal{V}$, TRAR first calculates the popularity and the Gini coefficient of v_j , and then it selects the POIs satisfying $Pop(v_j) > \eta_1$ and $Gini(\vec{p}_j^i) > \eta_2$, where η_1 and η_2 are two thresholds set by the empirical studies; Finally, it selects e_{ij} with maximal $Pop(e_{ij})$ as an AR of v_j and inserts the selected AR into \mathcal{R} .

For example, we calculate the Gini coefficient of four popular POIs shown in Fig.2. $\vec{p}_1 = [0.1, 0.1, 0.2, 0.6]$ indicates that there are 4 incoming routes of v_1 , and the elements in \vec{p}_1 are the popularity proportion of the incoming routes. Similarly, $\vec{p}_1 = [0.1, 0.1, 0.2, 0.6]$, $\vec{p}_2 = [0.4, 0.6]$, $\vec{p}_3 = [0.3, 0.3, 0.4]$, $\vec{p}_4 = [0.2, 0.8]$. According to Eq.7, $Gini(\vec{p}_1) = 0.4$, $Gini(\vec{p}_2) = 0.1$, $Gini(\vec{p}_3) = 0.067$, $Gini(\vec{p}_4) = 0.3$. Thus, v_1 has a high popularity and Gini coefficient, and its incoming route with the maximal popularity proportion 0.6 is an attractive route.

The popularity inequality of incoming routes by Gini coefficient has some characteristics: (1) Gini coefficient ranges between (0, 1); while the Gini coefficient is approaching to 0, the popularity proportions of incoming routes are close to equal; (2) The fewer the number of incoming routes is, the lower the Gini coefficient is; (3) With the same number of incoming routes, the more unbalanced the popularity proportion is, the higher the Gini coefficient is.

Complexity. AR discovery traverses the travel graph to discover ARs and computes Gini coefficient of each POI costs $O(1)$ due to the limited incoming routes, so the complexity is $O(n)$, where n is the number of POIs.

4.2 AR Evaluation

The attraction between an AR and its related POI is different, TRAR utilizes a gravity model to evaluate the preference and rating score of each AR. In the gravity model, we take the AR (denoted by e_{ij}) in Fig.3(a) as an example to illustrate the process of evaluation. For an attractive route e_{ij} , TRAR selects the users who have visited v_j through e_{ij} from the travel history, and maps them into a category space with their preferences. Then, it also maps the POI v_j into the category space with its preference. TRAR clusters the preferences of users in the z -dimensional space by applying k -means clustering (Kanungo et al. 2002), which is chosen due to its computational efficiency. Here we set $k = 2$ to

distinguish the users who prefer the AR or not. Fig.3 has shown that the travel time of users who prefer the attractive routes is longer than that of other, because an AR attracts the users to cost longer time for visiting. Therefore, we utilize the travel time of a route to distinguish the users who prefer it or not. The users take a longer time along e_{ij} are labeled as preferring e_{ij} , and others are not.

In the category space, we treat the centroid of users' preferences who prefer e_{ij} as $P(e_{ij})$. Based on the labeled user preferences, a linear classifier l_{ij} is trained by SVM (Ben-Hur et al. 2001). It partitions the category space into two domains to distinguish the users who prefer v_j and the users who prefer e_{ij} . A linear classifier is a hyperplane in the category space, and it is determined by the support vectors as shown in Fig.5. The calculated l_{ij} is put into a classifier set L . TRAR learns a classifier for each attractive route, so that the attractive route e_{ij} is associated with the trained classifiers l_{ij} .

The gravity model is utilized to estimate the rating score of an attractive route. Let $dist(v_j, l_{ij})$ denote the distance from v_j to l_{ij} in the category space, and $dist(e_{ij}, l_{ij})$ denote the distance from e_{ij} to l_{ij} . $RS(v_j)$ is obtained from Yelp⁴, which represents the mass of v_j in the category space. We assume that the attraction from v_j and e_{ij} are balanced on the hyperplane. According to the gravity model (Zhang and Chow 2015), $RS(v_j)$ and $RS(e_{ij})$ are treated as different masses in the space, and we apply the force balance on the hyperplane to calculate $RS(e_{ij})$, as follows:

$$\frac{RS(v_j)}{(dist(v_j, l_{ij}))^2} = \frac{RS(e_{ij})}{(dist(e_{ij}, l_{ij}))^2} \quad (8)$$

Then $RS(e_{ij})$ is normalized to 0-5, which is consistent with the range of POI rating score. The calculated $RS(e_{ij})$ is put into the classifier set S .

Complexity. AR evaluation traverses the attractive routes to evaluate ARs. The complexity of this process is $O(kn)$, where k is a small constant, n is the size of POI set.

4.3 Trip Recommendation

Since the problem of PEAR is NP-hard, we propose a heuristic method to recommend a personalized trip for a user. For a region with n POIs, we construct a travel graph $G(\mathcal{V}, \mathcal{E})$ with n nodes being the POIs; these n nodes are available to visit under the time constraint of the user, we take the temporal information into consideration of POI (Yuan et al. 2013). The user experience and travel time of the route e_{ij} is initialized as: $E(u, e_{ij}) = 0, T(u, e_{ij}) = t(e_{ij})$. When a new user u arrives, according to the user preference $P(u)$ and the preference of each POI $P(v)$, the recommender calculates the preference score between the user and each POI. To narrow down the searching range in G , the recommender selects the top- k POIs by their preference scores. A new graph $G' = (\mathcal{V}', \mathcal{E}')$ is generated by these selected POIs as \mathcal{V}' and the routes among them as \mathcal{E}' , which is a subgraph of G .

The recommender maps the preference of user u $P(u)$ in the category space for each attractive route in $\mathcal{E}' \cap \mathcal{R}$. If $P(u)$ falls in the domain of e_{ij} , the recommender updates

⁴<http://www.yelp.com/>

the user experience and time cost $T(u, e_{ij})$ on e_{ij} in G' . The attraction of AR is utilized to heighten the weight of the route. The user experience on e_{ij} is updated by $E(u, e_{ij})$, and the time cost $T(u, e_{ij})$ is updated by:

$$T(u, e_{ij}) = t(e_{ij}) + \mathbb{I}_{u \rightarrow e_{ij}} \Delta T(u, e_{ij}) \quad (9)$$

where $\mathbb{I}_{u \rightarrow e_{ij}}$ is an indicator function which returns 1 when u prefers to e_{ij} , otherwise returns 0. As discussion in Fig.3, $\Delta T(u, e_{ij})$ is the value that the average travel time of users who prefers to e_{ij} subtracts the average travel time of users who do not. The updated travel graph is denoted as $G''(u)$, which is personalized for the user u .

Under the constraints discussed in Section 3.3, TRAR executes a greedy algorithm on $G''(u)$ according to the time cost and user experience on each edge as follows: For the current selected POI, TRAR collects its unvisited neighbouring POIs. Then, it chooses the next POI with the maximal UE from the current POI to its unvisited neighbouring POIs. The total UE and time cost is updated. The above process is terminated until the total time cost is larger than the user time constraint if a next unvisited POI is inserted to the trip. Thus, an optimal trip with maximal user experience is generated for u .

Theorem 2. *Trip Recommendation is an $\Omega(OPT)$ approximation, with OPT as the optimal user experience.*

Proof. Consider a simple instance that all POIs and routes have unit user experience, and all POIs have no stay time, $\delta = 0$. In such instances, TRAR recommends a trip to a user which is an optimal solution. \square

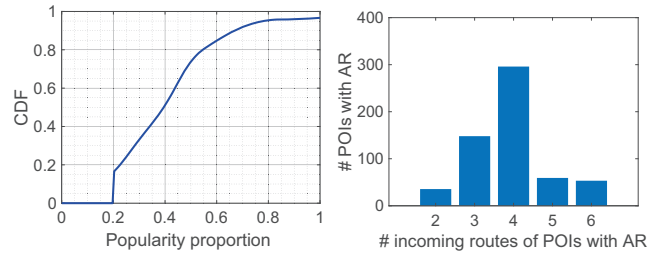
Complexity. Calculating top- k POIs costs $O(n \lg n)$, updating the travel graph costs $O(n^2)$, traversing the personalized travel graph costs $O(n)$. So the time complexity of trip recommendation is in $O(O(n \lg n) + O(n^2) + O(n)) = O(n^2)$, where n is the number of POIs.

5 Experiments

5.1 Experimental Setup

Dataset. The datasets include two parts: One is synthetic dataset to evaluate the scalability of recommender system, and it contains 20,000 users, 200 POIs, 50 attractive routes. The synthetic dataset is generated by a simulator (Lu, Chen, and Tseng 2012), and it randomly selects routes as attractive routes. Another one is Foursquare dataset (Zhang et al. 2015), which is a publicly large-scale check-in LBSNs dataset. It contains 11,326 users, 7,711 POIs, and 1,385,223 check-ins. The rating score of POIs in Foursquare is obtained from Yelp (Hu and Ester 2013).

We divide a user’s travel history by day, and separate each one into distinct trips if the check-in time between two consecutive POIs is more than a threshold. And similar to existing work (Lim 2015), we set 8 hours for this threshold in our experiments. These trips also serve as the ground truth of real-life user trips, which are subsequently used for evaluating our algorithm and baselines. From the travel history of users, we discover 592 attractive routes by AR discovery process. We select the earliest 80% trips to train the model, and use the other 20% as a testing dataset.



(a) Popularity proportion of ARs (b) Incoming routes

Figure 6: ARs in the Foursquare dataset.

Experimental Settings. η_1 and η_2 in AR discovery are set at 800 and 0.3 as the default value. In the Synthetic dataset, we treat $\Delta T(u, e_{ij})$ in Eq.9 as a random variable following a certain distribution (Zhang et al. 2015).

Evaluation Metrics. In our experiments, each user in the testing dataset is related with two trips: one is the ground truth trip from the dataset, and the other is a predicted trip by recommender under the user preference and multi-constraints. The metrics include recall, precision and F_1 (Li et al. 2015; Zhang and Chow 2015), the testing results of all trips of test users are averaged to produce the final results.

5.2 AR Analysis

Fig.6(a) shows the distribution of the popularity proportion (described in Section 3.2) of all selected ARs in Foursquare, whose popularity proportion are concentrated at 0.2 to 0.6, which shows the popularity of ARs occupy an primary part of the POI popularity. There are no ARs whose popularity proportion is less than 0.2. Fig.6(b) shows the amount of POIs with different number of incoming routes. The number of POIs of 4 incoming routes with AR is the largest of all.

5.3 Baselines

Since our work is the first to consider attractive routes, we implement the following baselines related to our work:

GreedyTSPCost (GTC): This algorithm recommends POIs with maximum user experience per unit time to visitors (Friggstad et al. 2018), which greedily chooses a POI with higher user experience and shorter travel time.

MPTR: This algorithm recommends a sequence of POIs meeting multiple constraints of users by mining user’s preference to POIs (Luan et al. 2018), which ignores the attractive routes in the recommendation process.

RandomWalk (RandW): This algorithm (Lim et al. 2015) randomly selects an unvisited POI in a trip.

5.4 Variants of TRAR

In addition to the above baselines, we also evaluate two variants of the proposed TRAR.

Sat: This variant only recommends unvisited POIs with high user experience for a user to generate a personalized trip.

Hap: This variant only chooses unvisited attractive routes with high user experience for a user as many as possible to form a personalized trip.

Table 2: Performance comparison between TRAR and baselines on different datasets.

Dataset	Method	R@3	R@5	R@10	R@15	P@3	P@5	P@10	P@15	F ₁ @3	F ₁ @5	F ₁ @10	F ₁ @15
Foursquare	RandW	0.1139	0.1237	0.1359	0.1586	0.2245	0.1840	0.1464	0.1101	0.1512	0.1480	0.1410	0.1300
	MPTR	0.1353	0.1479	0.1759	0.2355	0.2369	0.1874	0.1639	0.1428	0.1722	0.1653	0.1697	0.1778
	GTC	0.1638	0.1887	0.2735	0.3046	0.3393	0.3075	0.2678	0.1739	0.2210	0.2331	0.2706	0.2214
	TRAR	0.2979	0.3657	0.4358	0.4648	0.4178	0.3684	0.2733	0.2175	0.3478	0.3670	0.3359	0.3145
Synthetic	RandW	0.1102	0.1210	0.1405	0.1586	0.2126	0.1840	0.1326	0.1101	0.1450	0.1453	0.1362	0.1300
	MPTR	0.1200	0.1300	0.1603	0.2250	0.2105	0.1731	0.1514	0.1320	0.1500	0.1473	0.1548	0.1648
	GTC	0.1521	0.1843	0.2516	0.2724	0.3250	0.2950	0.2521	0.1619	0.2053	0.2236	0.2500	0.2009
	TRAR	0.2800	0.3511	0.4253	0.4501	0.4002	0.3500	0.2607	0.2200	0.3294	0.3500	0.3226	0.2955

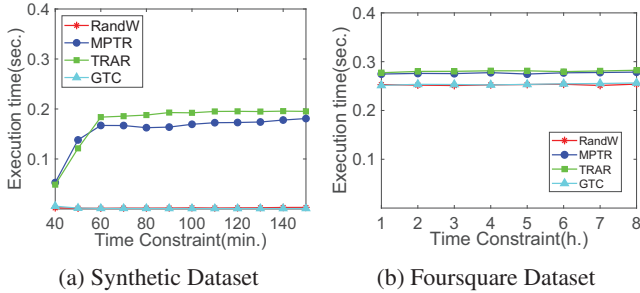


Figure 7: Execution time on different datasets.

5.5 Evaluation Results and Discussion

Execution time on different datasets We performed the evaluation of TRAR on a 3.3GHz CPU with 8GB of RAM. Fig. 7 shows average execution time of different algorithms on different datasets for test users. In the Fig. 7(a) and (b), the execution time of GTC and RandW is smaller because the two methods are easy to perform. The execution time of TRAR is slightly larger than that of MPTR, because TRAR considers both POIs and ARs in the recommend process. The execution time of TRAR is still tolerable for users. With time constraint increases, the execution time tends to be stable, because of finite number of ARs and POIs in the personalized travel graph. The execution time in Fig. 7(b) is slightly higher than that in Fig. 7(a) when the algorithm is fixed, because Foursquare dataset is larger and richer than synthetic dataset. The execution time in the experiment is sufficient for most real-life applications.

Impact of trip time constraint The travel time is not more than the trip time constraint. As shown in Fig.8(a), this experiment compares the user experiences by of TRAR, GTC, MPTR and RandW on synthetic dataset when the travel time is varied from 40 to 150 minutes. TRAR outperforms GTC, MPTR and RandW, and achieves the highest score while the time constraint is increasing. Because MPTR chooses POIs of higher rating scores and ignores attractive routes, and GTC chooses POIs with maximum user experience per unit time, which may not fully fit for user’s interest. The gap between the user experiences of TRAR and GTC is the extra user experience from attractive routes. The score of TRAR is lower than that of GTC with short time constraint, because GTC chooses POIs with higher user experience per unit time while TRAR chooses the POIs and attractive routes which fit for the preference of a user.

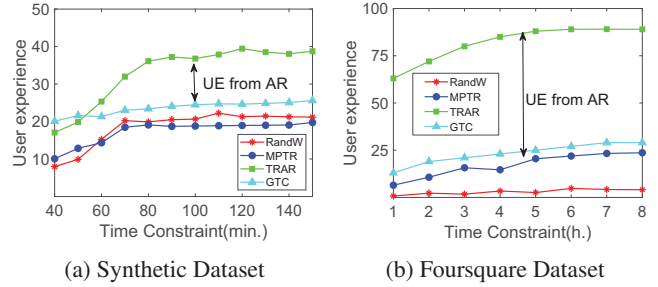


Figure 8: Impact of trip time constraint.

As shown in Fig.8(b), this experiment compares the user experience of TRAR, GTC, MPTR and RandW on Foursquare dataset when the travel time is varied from 1 to 8 hours. The user experiences of all methods increase at first and tend to be stable by increasing the time constraint (i.e., travel time), and the reason is same as that in synthetic dataset. The score is much higher than the methods in synthetic dataset, because the categories in Foursquare dataset are richer than that in synthetic dataset. TRAR achieves the highest score with increasing time, since TRAR considers the user experience from attractive routes.

Performance metrics: Recall, Precision and F₁ Table 2 summarizes the performance with the metrics Recall@k, Precision@k and F₁@k on both of Foursquare dataset and synthetic dataset. The results show that TRAR improves recommendation performance. TRAR, GTC and MPTR outperform RandW in terms of all metrics. It indicates that TRAR has a significant performance disparity in terms of top-k recall. TRAR outperforms other baseline algorithms, such as the recall@5 of TRAR in the Foursquare dataset is 0.3657, while the recall@5 of other three algorithms are 0.1139, 0.1353 and 0.1638 shown in Table 2. First, this is because TRAR applies the gravity model in category space to model the attraction of attractive routes and produce trip recommendation. Second, compared with GTC and MPTR, TRAR justifies the benefit brought by considering the similarity between the preferences of users and attractive routes. Third, TRAR performs well than other algorithms on recall and precision which implies that the visitors gain more user experience from attractive routes. The value of Recall, Precision and F₁ with Foursquare dataset are higher than that of synthetic dataset due to the rich and diverse data of Foursquare dataset, which shows that TRAR is more effective and performs well in the true dataset.

Table 3: Comparison of variants of TRAR.

Variant	Recall	Precision	F ₁
Sat	0.142	0.231	0.184
Hap	0.122	0.154	0.109

Variants of TRAR We compare the variants of TRAR on the Foursquare dataset as shown in Table 3. Sat solely selects unvisited POIs with high user experience for a user to generate a personalized trip, and consequently the results are more similar to MPTR. Sat outperforms Hap in these three metrics because Hap only selects attractive routes the user interested in as many as possible, while the number of attractive routes is far less than that of POIs. Though attractive routes improve user experience, it may not fit to the preference of user. Sat satisfies the requirement of user but lacks the user experience from attractive routes. Based on the above discussion, the performance of Sat is superior to Hap, but TRAR is better than both of them as shown in Table 2.

6 Conclusion

In this paper, we identify a novel kind of route called attractive route, and propose a novel method named TRAR for personalized recommendation trip with ARs. TRAR discovers attractive routes based on the popularity and the Gini coefficient of POIs, and we propose (1) a gravity model under the category space to estimate the rating score and preference of an attractive route, and (2) a recommendation algorithm to efficiently plan the trip which aims to maximize the user experience. As far as we know, this is the first work on personalized trip recommendation that considers the attractive routes. Through a series of experiments by different datasets, we validate the effectiveness of TRAR and show TRAR performs well under various conditions. In the future work, we will apply TRAR to more applications.

Acknowledgments

This work is supported by NSFC grants No. 61572113, 61632009; the Science and Technology Achievements Transformation Demonstration Project of Sichuan Province of China No. 2018CC0094; the Science and technology program of Hunan Province 2017XK2002; the Fundamental Research Funds for the Central Universities No. ZYGX2015J155, ZYGX2016J084, ZYGX2016J195, ZYGX2019J075, 2082604401036; and Big Data Application on Improving Government Governance Capabilities National Engineering Laboratory Open Fund Project.

References

Applegate, D. L.; Bixby, R. E.; Chvatal, V.; and Cook, W. J. 2007. The traveling salesman problem: A computational study. *Princeton university press*.

Barro, R. J. 2000. Inequality and growth in a panel of countries. *Journal of Economic Growth*.

Ben-Hur, A.; Horn, D.; Siegelmann, H. T.; and Vapnik, V. 2001. Support vector clustering. *Journal of Machine Learning Research*.

Friggstad, Z.; Gollapudi, S.; Kollias, K.; Sarlos, T.; and Tomkins, A. 2018. Orienteering algorithms for generating travel itineraries. In *ACM WSDM*.

Hu, B., and Ester, M. 2013. Spatial topic modeling in online social media for location recommendation. In *ACM RecSys*.

Javed, W.; Ghani, S.; and Elmqvist, N. 2012. Gravnav: Using a gravity model for multi-scale navigation. In *ACM AVI*.

Jiang, W.; Wu, J.; and Wang, G. 2015. On selecting recommenders for trust evaluation in online social networks. In *ACM TOIT*.

Johnson, I.; Henderson, J.; Perry, C.; Schöning, J.; and Hecht, B. 2017. Beautiful... but at What Cost?: An Examination of Externalities in Geographic Vehicle Routing. *ACM IMWUT*.

Kanungo, T.; Mount, D. M.; Netanyahu, N. S.; Piatko, C. D.; Silverman, R.; and Wu, A. Y. 2002. An efficient k-means clustering algorithm: analysis and implementation. *IEEE TPAMI*.

Li, H.; Hong, R.; Zhu, S.; and Ge, Y. 2015. Point-of-interest recommender systems: A separate-space perspective. In *IEEE ICDM*.

Li, H.; Ge, Y.; Lian, D.; and Liu, H. 2017. Learning user’s intrinsic and extrinsic interests for point-of-interest recommendation: A unified approach. In *IJCAI*.

Lim, K. H.; Chan, J.; Leckie, C.; and Karunasekera, S. 2015. Personalized tour recommendation based on user interests and points of interest visit durations. In *IJCAI*.

Lim, K. H.; Chan, J.; Leckie, C.; and Karunasekera, S. 2017. Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. In *KAIS*.

Lim, K. H. 2015. Recommending tours and places-of-interest based on user interests from geo-tagged photos. In *Acm Sigmod on Phd Symposium*.

Lu, Y., and Shahabi, C. 2015. An arc orienteering algorithm to find the most scenic path on a large-scale road network. In *ACM SIGSPATIAL*.

Lu, Y.; Jossé, G.; Emrich, T.; Demiryurek, U.; Renz, M.; Shahabi, C.; and Schubert, M. 2017. Scenic routes now: Efficiently solving the time-dependent arc orienteering problem. In *ACM CIKM*.

Lu, E. H.-C.; Chen, C.-Y.; and Tseng, V. S. 2012. Personalized trip recommendation with multiple constraints by mining user check-in behaviors. In *ACM SIGSPATIAL GIS*.

Luan, W. J.; Liu, G. J.; Jiang, C. J.; and Zhou, M. C. 2018. MPTR: A maximal-marginal-relevance-based personalized trip recommendation method. *IEEE TITS*.

Maaten L, H. G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*.

Quercia, D.; Schifanella, R.; and Aiello, L. M. 2014. The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. In *ACM HT*.

Vansteenwegenaaba, P. 2011. The orienteering problem: A survey. *European Journal of Operational Research*.

Yoon, H.; Zheng, Y.; Xie, X.; and Woo, W. 2012. Social itinerary recommendation from user-generated digital trails. In *Personal and Ubiquitous Computing*.

Yuan, Q.; Cong, G.; Ma, Z.; Sun, A.; and Thalmann, N. M. 2013. Time-aware point-of-interest recommendation. In *ACM SIGIR*.

Zhang, J.-D., and Chow, C.-Y. 2015. Spatiotemporal sequential influence modeling for location recommendations: A gravity-based approach. In *ACM TIST*.

Zhang, C.; Liang, H.; Wang, K.; and Sun, J. 2015. Personalized trip recommendation with poi availability and uncertain traveling time. In *ACM CIKM*.