DEIM Forum 2018 D3-1

# ユーザ移動履歴を用いた個人化観光ツアーの推薦

孫　俊傑† 　Chenyi Zhuang† 　馬　強†

† 京都大学大学院情報学研究科社会情報学専攻　〒 6068501 京都市左京区吉田本町
E-mail: †{jj-sun,zhuang}@db.soc.i.kyoto-u.ac.jp, ††qiang@i.kyoto-u.ac.jp

**Abstract**　Travel route recommendation that recommends a set of Points-of-interest(POIs) for users is one of useful applications and a challenging task due to various of users travel preferences and constraints in location based social networks (LBSN). Conventional methods are only focusing on how to evaluate traveler's POI preferences (i.e. user-location relationships) and then recommend travel sequences with POI popularity. In this paper, we propose a general method to recommend travel routes by learning both user-location and location-location relations from tourists ' behaviors. Experimental results reveal that our framework improves conventional methods and demonstrates that transition knowledge is helpful in travel route recommendation problems.

**Key words**　Travel route recommendation, location-based social network, sightseeing

## 1. Introduction

With the development of location-based social network (LBSN) services, there are a large amount of user-location check-in behaviors via various of devices. Thus, there are many applications and research focusing on recommending individual Point of Interest (POI) that according to user ratings or check-ins [1], [2]. For example, TripAdvisor is a website that provides tour guide and recommends attractions in a city. These services can provide much more choices for tourists when they travel in a specific city, which increase the difficulty to plan their trip.

It is a significant task to help tourists plan their travel routes in an unfamiliar city, especially given the massive volume of information that is available to them. In this paper, we call it travel route recommendation problem that aims to recommend a sequence of POIs under tourists trip constraints.

Most previous studies formulate travel route recommendation problems based on the orienteering problem model [18], in which tourists earn a reward score when they visit one POI, and a travel path with a maximum reward score under several trip constraints will be recommended. The reward score can be regarded as tourist preferences in regard to POIs, e.g., [9], [14], [16]. However, these works are only focus on features of POIs while location-location relations (i.e, transition) are not considered in their models.

Since tourist preferences are very difficult to estimate [3], learning tourists' behavior patterns instead of preference will help improve travel route recommendations. Furthermore, in actual travel routes, some locations show strong connections between each other and always visit these locations in a certain order. (see also Figure 1)

Recently, [12] proposes a tour route recommendation method that learns both locations and transitions. Their results show the potential of improving recommendation performance by considering transitions. In this paper, we propose a latent factorization model that learns transition patterns and then recommend travel routes with both learned locations and transitions knowledge.

Our main contributions are as follows:

- We propose a general hybrid method, which jointly consider user-location and location-location relations to recommend travel routes, which are satisfied with tourists constraints.
- The experimental results verify our methods achieve the better performance comparing to recent studies on various of datasets.

## 2. Related Work

Travel route recommendation is a well-studied field and there are many stuedies and developed applications.

### 2.1 POI Recommendation

Most location-based recommendation falls into this category which recommend an individual POI to users according to their extral information or context by ranking a list of POIs. [1] and [2] consider the spatial influence and propose within matrix factorization and probabilistic model, respectively. [3] summarized these related works and evaluated the performance with various of methods, and the result shows that matrix factorization methods provide better performances compare to probabilistic models. [4] improves the recommendation performance by discovering the relationship between POIs.

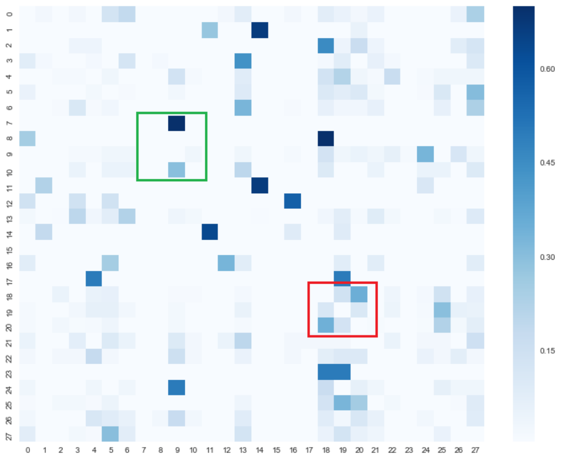


Figure 1 Transition matrix of the Toronto dataset. Each dimension represents POIs and each entry represents the observed transition probability from one POI to another. Transitions in the green rectangle are asymmetric structure while they are symmetric in the red rectangle.

### 2.2 Next Location Recommendation

Next location recommendation problem is that try to predict users' next visit location based on users context information (e.g., users visited location, current time). The solution to this problem include applying Markov chains into a tensor factorization model [5], [6], incorporating geographical influence into topic model [7], and using recurent neural networks to predict users next visit location [8].

### 2.3 Travel Route Recommendation

In this paper, we consider the problem of travel route recommendation which recommends a travel route. Many conventional works model this problem with orienteering problem [18]: Both [9] and [14] develop applications which recommend travel routes with users manually selected preferences, and [16] proposes a framework that recommend personalized travel routes with a heuristic considerition: Users travel duration would be longer when they visit POIs that match their prefered categories. [11] and [10] consider the uncertain travel time between POIs and model users preference with matrix factorization methods to recommend routes. Traffic condition is considered in [13]. These works mainly focus on user POI preferences, and do not consider location-location relations (i.e. the transition of POIs) in their trip plannings.

[12] recommends travel routes base on both POIs and transition knowledge; their transition probability is modeled base on explicit feature-pairs such as the POI popularity and category, while the dependency of locations caused by spatiotemporal factors is not well considered. In contrast, in our work we infer transition patterns with a latent factorization model to study the connection between locations, which results in improved performance.

## 3. Preliminary

In this section, we first present two notions: *Travel Route* and *User Query*, and then define the trip recommendation problem.

**Definition 1** (Travel Route)**.** A travel route is a sequence of POIs (i.e. $p_1$, $p_2$, ..., $p_L$). Each point $p$ indicates that the tourist has visited this location and consists of {route id, POI id, category, datetime, longitude, and latitude }.

**Definition 2** (User Query)**.** A user query is a *query* $q = (p_s,\ p_e,\ L)$ in which $p_s$ and $p_e$ represent the start and end points, respectively, and $L$ represents the travel route length budget.

Assuming there are $n$ POIs, let P = { $p_1$, $p_2$, ..., $p_n$ } in the target city. In the trip recommendation problem, a travel route is recommended according to the user query by solving the following objective function:

$$\max_{x,u} \sum_{i=1}^{N-1} \sum_{j=2}^{N} x_{ij} R \tag{1}$$

$$s.t. \sum_{j=2}^{N} x_{1j} = \sum_{i=1}^{N-1} x_{iN} = 1, \sum_{i=2}^{N} x_{i1} = \sum_{j=1}^{N-1} x_{Nj} = 0 \tag{2}$$

$$\sum_{i=1}^{N-1} x_{ik} = \sum_{j=2}^{N} x_{kj} \leqq 1, \forall k = 2, \cdots, N-1 \tag{3}$$

$$\sum_{i=1}^{N-1} \sum_{j=2}^{N} x_{ij} = L - 1 \tag{4}$$

$$u_i - u_j + 1 \leqq (N-1)(1 - x_{ij}), \forall i, j = 2, \cdots, N, \tag{5}$$

where $R$ is the reward function and difined in the following sections; $N$ is the available number of POIs in one particular city; $x_{ij}$ is a binary indicator that equals 1 when users travel from POI $p_i$ to $p_j$, and equals 0 otherwise. Here we mark $p_1$ as the start location and $p_N$ as the end location. Constraint 2 specifies that a route must begin at the start location and finish at the end location, and a travel route cannot revisit the start location and travel out from the end location. Constraint 3 specifies that a POI can only be visited once. Constraint 4 is the travel budget constraint; it restricts the route length (i.e., the number of POIs that the tourist wants to visit). Constraint 5 specifies that subtours are to be avoided; this constraint was proposed in the classical traveling salesman problem [19].

## 4. Travel Route Recommendation

### 4.1 Location Based Recommendation

To recommend travel routes, we can first assign rewards to all the POIs in the city which represent the attraction of POIs, and a user recieve the reward if s/he visits this POI. A naive approach is that to recommend travel routes by ranking a list of POIs based on the popularity of POIs. We take it as a baseline method called PoiPop.

It is possible to assign a POI with a score to each query by leveraging all the other route features, as mentioned in [12]. All the features used in this approach are extracted from category, popularity (number of distinct visitors), total number of visits, average visit duration for each POI and location neighbourhood by grouping POIs into 5 clusters using K-means according to their geographical locations.

With the set of features described above, scores of POIs can be assigned by learning a ranking of POIs using rankSVM, with linear kernel and L2 loss [15]. The objective function is:

$$\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C \sum_{p_i, p_j \in \mathcal{P},\ q \in \mathcal{Q}} max\big(0, 1 - \mathbf{w}^T(\phi_{i,q} - \phi_{j,q})\big)^2,$$

where $\mathbf{w}$ is the parameter vector; $C > 0$ is a regularisation constant; $\mathcal{P}$ is the set of POIs to rank; $\mathcal{Q}$ represent all the queries respect to routes in training set; $\phi$ is the feature vector for the POI $p$ respect to the query $q$. The ranking score is computed as follows:

$$R_{i,q} = \mathbf{w}^T \phi_{i,q} \tag{6}$$

Taking $R_{i,q}$ into Eq. 1 and this approach is called PoiRank which only consider the ranking score respect to a particular query.

### 4.2 Transition Based Recommendation

Conventioanl work only considers the rewards on locations, while we think location-location relationship should also be considered in trvel route recommendation problem. We regard transition probabilities, which represent the likelihood of going from one POI to another POI, as one of rewards on transitions, and recommend travel routes by maximizing the transition likelihood. One naive idea would be to directly normalize the observed transition matrix and use it to represent transition probabilities. Then, the transition probabilities between POIs could be regarded as the reward of each transition. However, this idea has several drawbacks:

- The transition data that we can observe are not complete; i.e., some locations that have transitions may not be observed in our empirical data.
- Such a model would face the cold start problem; i.e., no transitions can be observed when new POIs are discovered and added to datasets.

Therefore, we need to infer transition probabilities from observed transitions. We recommend travel routes by taking transition rewards

$$R(p_j|p_i) = \hat{T}_{i,j} \tag{7}$$

into $R$ in Eq. 1; and $\hat{T}_{i,j}$ is the inferred transition probability that defines in the following subsections.

#### 4.2.1 Transition Matrix

From users travel routes in a particular city, what we can observe are not only users check-in behaviors on locations, but also transitions between locations (i.e. moving from POI $p_i$ to $p_j$). In this way, we capture a transition matrix represented by $T^{'} \in \mathbb{R}^{|V|\times|V|}$, where each entry $T^{'}_{i,j}$ denotes the observed transition frequency between POI $p_i$ and $p_j$ and $V$ is the number of POIs in a particular city. Then $T$ equals the row normalized $T'$, which can be regarded as the transition probability from $p_i$ to $p_j$.

Similar to the idea of collaborative filtering, in which POIs may have common attractions or connections to other POIs owing to common features that match very well, the matrix factorization model can be considered for this problem. One reasonable solution for inferring the transition matrix from observed data is to factorize the observed weighted transition matrix $T$ as below:

$$T \approx V_s M V_t^T, \tag{8}$$

where $V_s \in \mathbb{R}^{|V|\times k}$ and $V_t \in \mathbb{R}^{|V|\times k}$ represent latent features of source and destination points, respectively. $M \in \mathbb{R}^{k\times k}$ is the interaction matrix that represents the relationship between locations, and $k$ specifies the number of latent features. We learn these latent factors by solving the following optimization function:

$$\min_{V_s, V_t, M} I \odot \left\| T - V_s M V_t^T \right\|^2 + \lambda \Big[ \|V_s\|^2 + \|M\|^2 + \|V_t\|^2 \Big], \tag{9}$$

where $I$ is a binary weighted matrix whose entry $I_{s,t}$ indicates whether there is observed transition from $s$ to $t$. With learned latent factors, we infer a complete transition matrix and normalize to get the transition probability $\hat{T}^{\text{TMF}}_{i,j}$ by Eq.8. Then we can take it into Eq. 1 and recommend travel routes only based on the inferred transition probability. We call this approach Tmf.

#### 4.2.2 Spatial Influence

Athough we can infer a complete transition matrix by Tmf, there is still room to improve the performance. From the factorization of the transition matrix, some of information that the latent space can represent are easy to infer and understand, such as categories, popularity and so on. However, there is still no evidences showing that the latent space has included geographical information [1], which is a very important measurement when considering the relationship of locations. Hence, we embed the spatial influence explicity to our transiton matrix factorization model.

Firstly, we compute the distance between POIs with Haversine formula, then take the reciprocal and normalize each entry with the maximum value in the matrix to construct the POIs spatial influence matrix $G$. We embed the spatial features of POIs $V_g$ by factorizing $G$ as below:

$$G \approx V_g V_g^T. \tag{10}$$

The reason of such construction is that each entry represents the confidence of location-location spatial influence; in other words, the distance is smaller and the influence is larger.

We embed the spatial feature $V_g$ by factorizing $G$ as below:

$$\min_{V_g} \left\| G - V_g V_g^T \right\|^2 + \lambda \left\| V_g \right\|^2. \tag{11}$$

After embedding the spatial influence feature, our weighted transition matrix factorization model is represented as follows:

$$T \approx V_s M V_t^T + V_g M V_g^T, \tag{12}$$

where interaction matrix $M$ represents relationships between locations while incorporating the location features and spatial features.

Then we learn the latent factors by solving the following optimization function:

$$\min_{V_s, V_t, M} I \odot \left\| T - V_s M V_t^T - V_g M V_g^T \right\|^2 + \lambda \left[ \left\| V_s \right\|^2 + \left\| M \right\|^2 + \left\| V_t \right\|^2 \right]. \tag{13}$$

Similarly, taking the row normalize of the inferred transition matrix to get the transition probability $\hat{T}^{\mathrm{GTMF}}$, and recommend travel route based on this is called GTMF.

### 4.3 Hybrid Approach

After assigning rewards to POIs and transitions, we now aim to combine knowledge on locations and transitions to recommend travel routes. For locations, we obtain POI ranking scores according to user queries, and we use the following softmax function to transform the ranking scores into point rewards:

$$R_P(p_j|q) = \frac{\exp(R_{j,q})}{\sum_i \exp(R_{i,q})}, \tag{14}$$

where $R_{j,q}$ and $R_{i,q}$ are computed with Eq. 6.

For transitions, we take each entry $R_T(p_j|p_i) = \hat{T}_{i,j}$ from the transition probability matrix as the transition reward, which can be regarded as the probability of a user moving from a source to a destination location; thus, this process considers the visiting order of POIs.

We now combine point and transition rewards. In this case, the reward $R$ in Eq. 1 is represented as follows:

$$R(p_j|p_i, q) = \alpha R_P(p_j|q) + (1-\alpha) R_T(p_j|p_i), \tag{15}$$

where $R(p_j|p_i, q)$ is the combined reward and $\alpha \in (0, 1)$ is a trade-off parameter that indicates the importance between point and transition rewards which can be tuned using cross validation in practice.

### 4.4 Optimization and Latent Variable Learning

We solve our travel route recommendation objective function Eq.1∼5 with Gurobi optimization package [注1]. We minimize the objective function Eq. 9, Eq. 11, and Eq. 13 with a gradient decent approach by iteratively optimizing the latent variables $V_s$, $M$, $V_t$, and $V_g$, which is supported by Theano framework.

## 5. Experiments

### 5.1 Experimental Setup

To evaluate our proposed approaches, we apply our methods to the public location-based social network datasets that provided by [16], [12]. These datasets are users travel routes extracted from Flickr photos in five cities, and the statistics is shown in Table 1.

In our travel route recommendation settings, we randomly divide city travel routes with lengths ≧ 2 into five folds. We use leave-one-out cross validation to evaluate our approaches, which means that when testing on one fold of the dataset, we use other folds of data and short routes (i.e., routes that only have one POI) to train our models. For comparison, we test the following listed methods on each city dataset.

- PoiPop: Recommending travel routes only base on POI popularity.
- PoiRank, Markov, Rank+Markov: Proposed in [12], these three methods recommend travel routes based on the ranking of POIs rewards, transition probabilities by factorising explicit feature matrices, and combine PoiRank with Markov, respectively.
- Tmf, GTmf: Recommending travel routes only base on rewards of transitions which factorize transition matrix, and transition matrix with additional geographical influences, respectively.
- Rank+Tmf, Rank+GTmf: Combine PoiRank with our proposed Tmf and GTmf to recommend travel routes based on location rewards and transition rewards, respectively.

Table 1 **Statistics of datasets**

| Dataset | #Photos | #Check-ins | #Routes | #Users |
|---|---|---|---|---|
| Edinburgh | 82,060 | 33,944 | 5,028 | 1,454 |
| Glasgow | 29,019 | 11,434 | 2,227 | 601 |
| Melbourne | 94,142 | 23,995 | 5,106 | 1,000 |
| Osaka | 392,420 | 7,747 | 1,115 | 450 |
| Toronto | 157,505 | 39,419 | 6,057 | 1,395 |

(注1)：Gurobi Optimization. http://www.gurobi.com

Table 2 $F_1$ scores. The best method for each dataset (i.e., city) is shown in bold, the second best is shown in italic.

| | Edinburgh | Glasgow | Melbourne | Osaka | Toronto |
|---|---|---|---|---|---|
| PoiPop | $0.697 \pm 0.158$ | $0.678 \pm 0.120$ | $0.606 \pm 0.143$ | $0.659 \pm 0.130$ | $0.678 \pm 0.120$ |
| PoiRank | $0.679 \pm 0.145$ | $0.749 \pm 0.164$ | $0.625 \pm 0.149$ | $0.724 \pm 0.160$ | $0.749 \pm 0.164$ |
| Markov | $0.667 \pm 0.152$ | $0.711 \pm 0.151$ | $0.573 \pm 0.155$ | $0.688 \pm 0.152$ | $0.711 \pm 0.151$ |
| Rank+Markov | *0.705 ± 0.162* | *0.756 ± 0.165* | *0.626 ± 0.154* | $0.720 \pm 0.166$ | $0.743 \pm 0.165$ |
| Tmf | $0.684 \pm 0.158$ | $0.716 \pm 0.168$ | $0.575 \pm 0.155$ | $0.728 \pm 0.161$ | $0.654 \pm 0.147$ |
| GTmf | $0.658 \pm 0.170$ | $0.734 \pm 0.191$ | $0.567 \pm 0.162$ | $0.737 \pm 0.169$ | $0.722 \pm 0.175$ |
| Rank+Tmf | $0.691 \pm 0.156$ | $0.746 \pm 0.178$ | $0.624 \pm 0.153$ | *0.751 ± 0.171* | *0.751 ± 0.164* |
| Rank+GTmf | **0.709 ± 0.163** | **0.764 ± 0.175** | **0.630 ± 0.156** | **0.764 ± 0.174** | **0.764 ± 0.172** |

Table 3 Pairs-$F_1$ scores. The best method for each dataset (i.e., city) is shown in bold, the second best is shown in italic.

| | Edinburgh | Glasgow | Melbourne | Osaka | Toronto |
|---|---|---|---|---|---|
| PoiPop | $0.429 \pm 0.253$ | $0.507 \pm 0.296$ | $0.303 \pm 0.177$ | $0.360 \pm 0.193$ | $0.384 \pm 0.201$ |
| PoiRank | $0.399 \pm 0.226$ | $0.437 \pm 0.252$ | $0.327 \pm 0.203$ | $0.468 \pm 0.281$ | $0.512 \pm 0.293$ |
| Markov | $0.385 \pm 0.233$ | $0.503 \pm 0.298$ | $0.272 \pm 0.180$ | $0.416 \pm 0.261$ | $0.442 \pm 0.256$ |
| Rank+Markov | *0.446 ± 0.265* | *0.526 ± 0.299* | *0.333 ± 0.212* | $0.470 \pm 0.292$ | $0.502 \pm 0.292$ |
| Tmf | $0.413 \pm 0.251$ | $0.464 \pm 0.292$ | $0.273 \pm 0.177$ | $0.478 \pm 0.283$ | $0.363 \pm 0.227$ |
| GTmf | $0.412 \pm 0.277$ | $0.508 \pm 0.332$ | $0.268 \pm 0.188$ | $0.496 \pm 0.298$ | $0.476 \pm 0.296$ |
| Rank+Tmf | $0.413 \pm 0.156$ | $0.519 \pm 0.315$ | $0.329 \pm 0.208$ | *0.521 ± 0.305* | *0.517 ± 0.294* |
| Rank+GTmf | **0.461 ± 0.277** | **0.563 ± 0.330** | **0.339 ± 0.222** | **0.544 ± 0.313** | **0.539 ± 0.303** |

### 5. 2 Metrics and Evaluation

$F_1$ score is a common metric for evaluating POI and travel route recommendation [1], [12], [16] to measure whether POIs are correctly recommended. The $F_1$ score is defined as follows:

$$F_1 = \frac{2Precision \cdot Recall}{Precision + Recall},$$

Furthermore, since $F_1$ score on points ignores the visiting order between POIs, we adopt another measurement, pairs-$F_1$ which considers both the POI identity and visiting order [12]. This metric measures the $F_1$ score of every pair of POIs,

$$pairs - F_1 = \frac{2P_{PAIR}R_{PAIR}}{P_{PAIR} + R_{PAIR}},$$

where $P_{PAIR}$ and $R_{PAIR}$ are the precision and recall of ordered POI pairs, respectively. Pairs-$F_1$ takes values between 0 and 1 (higher is better). A perfect pairs-F1 is achieved if and only if both the POIs and their visiting order in the recommended route are exactly the same as those in the ground truth.

### 5. 3 Results

The travel route recommendation performance of various of approaches are summarised in Table 2 and Table 3, evaluated with $F_1$ score and pairs-$F_1$, respectively.

From the results, we can see that our hybrid approach Rank+GTmf, which recommende travel routes with both location and transition knowledge with spatial influence, outperforms all the baseline methods on different datasets.

The recommendation only base on popularity provides not bad performances which indicates that POI popularity is always an important feature for POI recommendation or travel route recommendation. PoiRank improves the performance upon PoiPop by leveraging more features than popularity.

Performances of approaches that recommend travel routes only base on transition rewards varied on different datasets. There are no distinct differences between explicit factorization method Markov and our proposed latent factorization method Tmf, which indicates the latent factor model includes those features already. The performance get significant improvement when additional spatial influence is considered.

## 6. Conclusion and Future Work

In this paper, to given travel routes data, we propose a method to recommend travel tours according to users query based on user-location and location-location relationship. Our expirical results demonstrate that transition knowledgement is helpful in travel route recommendation problem and our approaches outperfom conventional methods on different datasets.

Since there are still some rooms to improve our methods' performance when additional information can be provided, such as POI check-in distribution and social influences, a more aggregated and reliable framework should be considered in our future work. Furthermore, POIs that clustered from geo-tagged photos maybe not precise enough. Testing our approach upon real GPS travel datasets is also necessary.

## 7. Acknowledgements

This work is partly supported by MIC SCOPE(172307001).

### References


[1] Lian, D., Zhao, C., Xie, X., Sun, G., Chen, E. and Rui, Y., 2014, August. GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation. Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 831-840.

[2] Liu, B., Xiong, H., Papadimitriou, S., Fu, Y. and Yao, Z., 2015. A general geographical probabilistic factor model for point of interest recommendation. IEEE Transactions on Knowledge and Data Engineering, 27(5), pp.1167-1179.

[3] Liu Y, Pham TA, Cong G, Yuan Q. An experimental evaluation of point-of-interest recommendation in location-based social networks. Proceedings of the VLDB Endowment. 2017 Jun 1;10(10):1010-21.

[4] Zhuang, C., Yuan, N.J., Song, R., Xie, X. and Ma, Q., 2017. Understanding People Lifestyles: Construction of Urban Movement Knowledge Graph from GPS Trajectory. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence

[5] Cheng, C., Yang, H., Lyu, M.R. and King, I., 2013, August. Where You Like to Go Next: Successive Point-of-Interest Recommendation. Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, Vol. 13, pp. 2605-2611.

[6] Yuan, Q., Cong, G., Ma, Z., Sun, A. and Thalmann, N.M., 2013, July. Time-aware point-of-interest recommendation. Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, pp. 363-372.

[7] Kurashima, T., Iwata, T., Hoshide, T., Takaya, N. and Fujimura, K., 2013, February. Geo topic model: joint modeling of user's activity area and interests for location recommendation. In Proceedings of the sixth ACM international conference on Web search and data mining, pp. 375-384.

[8] Liu, Q., Wu, S., Wang, L. and Tan, T., 2016, February. Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, pp. 194-200.

[9] Vansteenwegen, P., Souffriau, W., Berghe, G.V. and Van Oudheusden, D., 2011. The city trip planner: an expert system for tourists. Expert Systems with Applications, 38(6), pp.6540-6546.

[10] Zhang, C., Liang, H. and Wang, K., 2016. Trip recommendation meets real-world constraints: POI availability, diversity, and traveling time uncertainty. ACM Transactions on Information Systems (TOIS), 35(1), p.5.

[11] Zhang, C., Liang, H., Wang, K. and Sun, J., 2015, October. Personalized trip recommendation with poi availability and uncertain traveling time. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 911-920.

[12] Chen, D., Ong, C.S. and Xie, L., 2016, October. Learning points and routes to recommend trajectories. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 2227-2232.

[13] Chen, C., Zhang, D., Guo, B., Ma, X., Pan, G. and Wu, Z., 2015. TripPlanner: Personalized trip planning leveraging heterogeneous crowdsourced digital footprints. IEEE Transactions on Intelligent Transportation Systems, 16(3), pp.1259-1273.

[14] Kurata, Y., 2011, January. CT-Planner2: More flexible and interactive assistance for day tour planning. Information and Communication Technologies in Tourism, Vol. 2011, pp. 25-37.

[15] Lee, Ching-Pei, and Chih-Jen Lin，2014. "Large-scale linear ranksvm." Neural computation 26, no. 4 (2014): 781-817.

[16] Lim, K.H., Chan, J., Leckie, C. and Karunasekera, S., 2015, July. Personalized Tour Recommendation Based on User Interests and Points of Interest Visit Durations. Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence， pp. 1778-1784.

[17] Shumaker, B.P. and Sinnott, R.W., 1984. Astronomical computing: 1. Computing under the open sky. 2. Virtues of the haversine. Sky and telescope, 68, pp.158-159.

[18] Vansteenwegen, P., Souffriau, W. and Van Oudheusden, D., 2011. The orienteering problem: A survey. European Journal of Operational Research, 209(1), pp.1-10.

[19] Papadimitriou, C.H. and Steiglitz, K., 1998. Combinatorial optimization: algorithms and complexity. Courier Corporation.