MDP-based Itinerary Recommendation using Geo-Tagged Social Media

Radhika Gaonkar[1], Maryam Tavakol[2], and Ulf Brefeld[2]

[1] Stony Brook University, NY
rgaonkar@cs.stonybrook.edu
[2] Leuphana Universität Lüneburg
tavakol,brefeld@leuphana.de

Abstract. Planning the next vacations is a complex decision problem. Many variables like the place(s) to visit, how many days to stay, the duration at each location, and the overall travel budget need to be controlled and arranged by the user. Automatically recommending travel itineraries would thus be a remedy to quickly converge to an individual trip that is tailored to a user's interests. While on a trip, users frequently share their experiences on social media platforms e.g., by uploading photos of specific locations and times of day. Their uploaded data serves as an asset when it comes to gathering information on their journey. In this paper, we leverage social media, more explicitly photo uploads and their tags, to reverse engineer historic user itineraries. Our solution grounds on Markov decision processes that capture the sequential nature of itineraries. The tags attached to the photos provide the factors to generate possible configurations and prove crucial for contextualizing the proposed approach. Empirically, we observe that the predicted itineraries are more accurate than standard path planning algorithms.

Keywords: Itinerary recommendation · Data extraction · Sequential decision making · Personalization.

1 Introduction

The Web has become an effective resource for travelers. In addition to an increasing number of travel blogs, verticals providing reviews and recommendations of places, restaurants and hotels, prove useful tools for planning trips and night outs. However, common resources do not exhaustively cover a wide range of aspects but often focus on narrow scopes to maintain a clear segregation to other content providers. Users who seek different types of information thus need to query various sites and aggregate the pieces of information themselves, which requires significant amount of time and effort.

At the same time, the rise of digital photography through widespread use of mobile devices and digital cameras has resulted in a great deal of photos being shared on the Web. Uploaded photos are mostly tagged by users with information snippets and key words to share the location, emotion, people, etc.

with others. A remarkable way of understanding itineraries is to study the photo streams of tourists in touristic zones.

In this paper, we showcase how freely available user-tagged information on the Web can be aggregated to recover trajectories of tourists in cities. Our analysis is based on the online photo streams of users that reflect (a possibly incomplete) sequence of visited locations during a trip. We thus turn photo-sharing sites into useful resources to reconstruct a user's trips. We use Flickr[3] as our main source to acquire such photo streams. Flickr proves useful to generate candidate lists of Points of Interest (POIs) for any city. Moreover, many photos already come with geographic, temporal, and/or semantic annotations. Photos annotated with geo-coordinates can be accurately placed on a map and if the user also provided semantic tags, the content can be indexed and further processed by Natural Language Processing techniques. Nevertheless, only a minority of photos possess such detailed descriptors.

A touristic trip is considered a sequential problem. At each stage of travel, a user chooses her next destination from a list of touristic points in the city. Additionally, the data provides implicit feedback on the user's preference of a touristic site by the photographs she uploads on Flickr. This partial labeling of the data fits well to the problem setting of RL-based approaches where the uncertainty of taking different actions and the resulting transitions is minimized by trading off exploration and exploitation [20]. We thus reconstruct sequence of POI visits using reverse engineering of historic user itineraries.

In our proposed approach, we take into account both the sequential nature of POI visits and the user's overall satisfaction.We learn a model of the traveler behavior as an MDP and extend it to make personalised travel recommendations. The system learns the optimal recommendation policy by observing the consequences of visiting different places by the travelers in the city and the traveler's personal preferences. Using the MDP, the user is recommended a place corresponding to the place category, which is nearest to her, both in distance and personal taste, and receives an immediate reward for taking that action. We empirically compare our approach to various path planning algorithms on data from three European touristic cities - Munich, Paris and London.

2 Related Work

Many systems have been developed to extract user-generated multimedia content and infer meaningful information for travel planning. Crandall et al. [4] is one of the earlier works in exploring the association of Flickr photos to physical locations, and apply their techniques to extract landmarks at various granularity levels that correspond to a geo-spatial hierarchy. Cao et al. [3] introduce a method that uses both logistic regression and kernel canonical correlation to enrich semantic information and location information based on image content. The tags assigned to Flickr photographs are further employed to extract place

[3] www.flickr.com

names, place coordinates, place types and popularity values [18, 15]. Y. Baba et al. [2] use co-occurrence between textual tags and geolocations to represent places related to a tag as probability distribution over the location coordinates.

The growing surge of travel data on social media platforms has resulted in many recent works on tourist place recommendations. Jiang et al. [7] enhance collaborative filtering recommendations with author topic model that consider different types of user preferences and exploit the textual metadata associated with geo-tagged pictures on Flickr. Zhang et al. [24] present an extension of the collaborative retrieval model (CRM) for POI recommendation, taking temporal information and social relations into account. [17] use Foursquare data to build a probabilistic generative framework that recommends tours based on user's preferences, peer circle, travel transitions and popularity of venues. [10] and [16] also restrict their work to the geo-tagged points on Flickr to find shortest routes with the highest satisfaction.

While all these models capture many different aspects of tourist movement, they fail to address sequentiality in travel movements. [20] and [22] propose sequential approaches to recommender systems using MDP [21]. Accordingly, probabilistic sequential approaches are used in recommending the next POI either based on location services [14, 19], or social networks [6]. In *WhereNext* [12] a T-pattern decision tree is designed to classify the trajectory patterns, and Muntean et al. [13] rank POIs using Gradient Boosted Regression Trees and Ranking SVM. Ashbrook et al. [1] apply a Markov model to GPS data in an attempt to model travel behavior. [8] combines user preference and current location into a probabilistic behavior model by combining topic and Markov models. [23] goes one step further to prune the search space and recommend sequential POIs considering their time constraints. In this paper, we propose a recommendation approach which additionally encodes the history of visited POIs into the Markov model in order to better understand the sequential patterns.

3 Data Extraction and Analysis

In order to automate the acquisition of tourist information, we make use of geo-temporal data from Flickr. The advent of digital photography and its continually increasing features of spatially and temporally annotating images in real time, has enriched photographs with useful meta-data. This results in augmenting the photographs with geographical coordinates specifying the location of the picture, as well as its date and time. Flickr has over 5 billion photographs, that many of them are time stamped. In addition, they have semantic data such as tags and titles associated with them. A small fraction of these photographs are annotated with geographical coordinates. Our system focuses on extracting and discovering a large number of trips from Flickr meta data and using these to deduce novel methods of itinerary recommendation.

3.1 Data Acquisition

Using the public API of Flickr, we collected 44051, 22970, 42104 photographs of three popular cities, Munich, London and Paris, along with their meta-data. However, a significant portion of the photographs are without geo-coordinates. Restricting ourselves to only the geo-referenced pictures would significantly decrease the coverage of our approach. Therefore, we utilize the meta-data associated with photographs to infer their locations. Nonetheless, working with such open data poses several key challenges. Most of the photographs have linguistically noisy tags or tags with no location information. For instance, the tags might only include the details of photography techniques, weather, city name, or contains semantic ambiguities. Hence, inferring POI names requires use of both the location coordinates and information gleaned from the textual tags. We propose approaches for data pre-processing which result in significant increase in the performance of our system.

3.2 From Coordinates to Places of Interest

In order to maintain a high quality of mapping from photograph location coordinates to place names, we query the free version of Google Places API and obtain a list of POIs in each city. This is with the assumption that many touristic points are already available on Google Maps and are highly reliable. We furthermore collect the textual tags of multiple photographs having the same location coordinates in the Flickr dataset. The tags are then cleaned, of stopwords, city names and Flickr specific stopwords such as camera and weather details. The place name and place type of the location coordinate is obtained by calling the API with *latitude*, *longitude*, *ranking criteria* and *search radius* around the location coordinates. To get the best place name candidate from the returned list, a fuzzy string match using Levenshtein distance [9] is used and the place with the highest fuzzy matching similarity with the textual tags of the location coordinates is assigned to that location. Despite these techniques, a small fraction of the coordinates never get a place name from the Google Places API. We thus assign their place name manually using the Google Maps interface.

3.3 Location Mapping with Tags

The user provided textual tags often contain event and geospatial information which could be used for inferring the location of non-geotagged data. Since users may define arbitrary tags, finding the relevant ones is not trivial. In addition, there is no sequential structure that could be exploited to support possibly contained geographical information using the concept of named-entity recognition or relation extraction in the text. We exploit the co-occurrence statistics of words in low dimensional vector space by using the Latent Semantic Analysis (LSA) [5] similarity between tags of a target non-geotagged photograph and each of the geo-tagged photographs. The LSA model is learned on the geo-tagged *(location, tags)* co-occurrence matrix. Using this, each new non-geotagged photograph tag is assigned a location from the highest similarity score, provided it is above a certain similarity threshold. The data points without any place information in tags are dropped for further analysis.

3.4 Itinerary Inference

After obtaining the POI names, we use the Flickr data to emulate tourist behavior. The first step is to remove all travel points falling outside the bounding box of a city. Our model aims to recommend only single-day itineraries. Therefore, the photograph sequences for more than one day are split by their datetime value into single day sequences. Additionally, it is important to differentiate between the resident and tourist in a city by checking the number of POIs covered by her. A resident would exhibit travel movements slower than a tourist. Therefore, we discard travel paths consisting of less than 3 unique POIs. Lastly, some photographers on Flickr add photographs with invalid date-time value or incorrect format. This hampers our modeling of the recommender system and is ergo removed. As a result, a total of 17904, 6000 and 9032 photographs are left for Munich, London and Paris, respectively.

4 MDP-based POI Recommendation

4.1 Preliminaries

The photos are uploaded by a set of users U in different cities. Each city c contains n_c POIs where $L_c = \{l_1, l_2, l_3, ..., l_{n_c}\}$ represents the set of POIs for that city. The photos are characterised by a set of attributes containing the timestamp of capture, the latitude and longitude of photo location, the title, the textual tag and description attached to the picture, and the category vector $\boldsymbol{cat}_l \in \mathbb{R}^m$ for each POI l. Note that each location is assigned to more than one category, i.e., for a certain category a, $\boldsymbol{cat}_l(a) = 1$ if l belongs to category a, and is zero otherwise. Our goal is to recommend an itinerary $I = (l_1, l_2, ..., l_k)$ for each user that tries to maximize her overall trip satisfaction.

An MDP is defined by a four tuple: (S, A, R, T), where S is the set of states, A is the set of actions, $R(s, a) : S \times A \to \mathbb{R}$ is the reward function that assigns a real value to each (state, action) pair, and $T(s, a, s') : S \times A \times S \to [0, 1]$ is the state-transition function, which provides the probability of a transition between every pair of states given an action from the available set. The goal of an MDP is to obtain the optimal policy, $\pi^* : S \to A$, that gives the best action for every state in order to maximise the sum of discounted reward.

In our problem, the *states* represent the history of user travels. State s_t is given by the sequence of at most k places the user has visited up to time t, $s_t = (l_1, ..., l_k)$. The *actions* are all POI categories present in the city where the user is visiting. *Transition probability* function models the probability of going to another place given the current location and the recommended place category. Each state that the user enters on taking a particular action, she gets an immediate reward from the *reward* function. A higher reward is awarded when the transition is present in the sequence of places in the training set.

4.2 The Predictive Model

We start with a simple Markov chain model to estimate the state-transition function. The transition function gives the probability of going to the next place l_{k+1}, for a user whose k recent POI visits are $(l_1, ..., l_k)$. A maximum-likelihood method is used to estimate this transition function based on the user travel data

$$N(s, s') = \frac{count(s')}{count(s)}, \tag{1}$$

where s and s' are $(l_1, l_2, ..., l_k)$ and $(l_1, l_2, ...l_{k+1})$, respectively. The *count* function gives the frequency of occurrences. We expand this model to an MDP framework which gives the probability of visiting a new POI l_{k+1} after choosing some action a, where $\boldsymbol{cat}_{l_{k+1}}(a) = 1$. The visit to this new POI depends on the place category recommended to the user at the current POI. The non-zero transitions occur when (s, s') occurs in the data set and a is a place category of s'. For each set of $\{s, a, s'\}$ transition probability is defined as

$$T(s, a, s') = \frac{N(s, s')}{\sum_{s'' \in S'} N(s, s'')}, \quad \sum_{s' \in S'} T(s, a, s') = 1, \tag{2}$$

where S' signifies the set of states that can be reached from s when action a is taken. The reward after taking action a in state s is given by the reward function $R(s, a)$, which is simply inferred by the number of occurrence of state-action sequences in the training data.

$$R(s, a) = \frac{count(s, a)}{count(s)} \tag{3}$$

4.3 Optimisation

The resulting MDP can be optimised using reinforcement learning methods such as value iteration. Value iteration learns the state-value function, $V(s)$, and converges to an optimal policy in a discounted finite MDP [21]. The policy is defined as the category recommendation for the traveller. An optimal policy π^* gives the highest expected utility through the traveller's movements. The utility of a state $V(s)$ is defined as the expected sum of discounted rewards that the agent obtains by starting from state s and following policy π. The standard update rule of value iteration with discount factor γ is given by:

$$V(s) = max_{a \in A(s)}[R(s, a) + \gamma \times \sum_{s'} T(s, a, s')V(s')]. \tag{4}$$

When the value function $V(s)$ converges to an optimal value function $V^*(s)$, the state-action values $Q(s, a)$ is derived

$$Q(s, a) = R(s, a) + \gamma \sum_{s'} T(s, a, s')V^*(s'). \tag{5}$$

The Q-values are proportional to the probability that the user visits a POI of place category a, given the sequence of visited POIs in s. Hence, a high $Q(s, a)$ indicates a higher likelihood of observing the transition from s with action a.

4.4 Multi-step Place Recommendations

We use the *softmax* function to approximate a probability distribution over the place categories from the Q-values

$$P(A = a|s) = \frac{exp\{Q(s,a)\}}{\sum_{a'} exp\{Q(s,a')\}}. \tag{6}$$

The action with the highest probability $a^* = \arg\max_a P(A = a|s)$, is recommended at each state. However, the system must consider various places associated with this category to recommend a specific place. We include the distance factor of POIs to predict the next place. Considering all places corresponding to the optimal policy, the recommended place l_{rec} is the place closest in distance to the current state. Since each state consists of a sequence of places $(l_1, l_2, ..., l_k)$, the distance from the last place l_k is considered. The recommendation score is calculated by the Euclidean distance between the last place l_k and the places in L_c corresponding to the place category of the optimal action.

$$l_{rec} = \arg \min_{\substack{l_x \in L_c \\ \boldsymbol{cat}_{l_x}(a^*)=1}} dist(l_k, l_x), \tag{7}$$

4.5 Online Personalisation

In order to personalise the recommendation model, we apply two technices for inferring user preferences from her travel history; duration based user interests as introduced in [11] and frequency based user interests.

Duration Based User Preference. Each location l_x in the user travel history contains an arrival time $t^a_{l_x}$ and departure time $t^d_{l_x}$. The duration-based user preference $\rho^{dur}_u(a)$, for user $u \in U$ and category a, is given by the fraction of time he spent at each of the POIs from category a in her travel history,

$$\rho^{dur}_u(a) = \sum_{\substack{l_x \in L_u \\ \boldsymbol{cat}_{l_x}(a)=1}} (t^d_{l_x} - t^a_{l_x}), \tag{8}$$

where L_u contains all the locations visited by user u. These preferences are then normalized to a $[0, 1]$ scale for each user. The more time a user spends at POI of a place category, the more likely it is that the user is interested in that category.

Frequency Based User Preference. In this method, the user preferences are inferred from the number of times a user visited POIs of a certain category [11]

$$\rho^{frq}_u(a) = \sum_{\substack{l_x \in L_u \\ \boldsymbol{cat}_{l_x}(a)=1}} count(l_x), \tag{9}$$

which is also normalized for each user.

The preference values obtained from either of techniques, form a preference vector for each user, e.g., $\boldsymbol{\rho_u} = \{a_1 : 0.6, a_2 : 0.01, \cdots, a_m : 0.3\}$. We incorporate

Path Length	1	2	3	4	5	6
History 1	0.041	0.041	0.042	0.042	0.041	0.034
History 2	0.098	0.090	0.096	0.106	0.100	0.103
History 3	0.097	0.090	0.093	0.105	0.090	0.087
History 4	0.089	0.084	0.083	0.094	0.077	0.060
History 5	0.074	0.071	0.058	0.072	0.070	0.058

Table 1: Variation of partial path accuracy@7 with user history

the individual preferences into our model at the time of recommending places for the optimal category a^*. We assign a score to each place proportional to the weighted sum of distance and the preference associated with its other categories,

$$l_{rec} = \arg \max_{\substack{l_x \in L_c \\ \boldsymbol{cat}_{l_x}(a^*)=1}} \left((1-\alpha) \times \frac{1}{dist(l_k, l_x)} + \alpha \times (\boldsymbol{\rho_u} \cdot \boldsymbol{cat}_{l_x}) \right), \tag{10}$$

where α is the personalisation coefficient. Hence, a place is recommended which is closer in distance, and belongs to the other categories that are preferred by the user.

5 Empirical Study

In this section, we first analyse the path accuracy of our recommended path by varying the amount of user history encoded in each state. On obtaining the optimal length of user history, we further compare the performance across three cities against several baselines. Moreover, we compute accuracy@k when the recommended places are among the top-k closets places to the current POI. All the experiments are conducted first without the effects of user preference and then when user preferences are included.

We use a time-series leave-one-out cross-validation method for tuning the parameters of our model. For the users with multi-day itineraries, we use the last day's travel sequence as the test set and the remaining as training set. For users that have only traveled on one day, the travel sequence is split into 60%-20%-20% of POIs into train, validation and test sets. For evaluation, paths of length z are obtained from the test set, where $z \in \{1, 2, 3, 4, 5, 6\}$. A path of length 1 would contain two place locations (l_1, l_2) and so on. The performance criteria evaluate how many of the recommended places are present in the test paths. The exact match accuracy of total n paths from test set is given by

$$Acc_{exact} = \frac{1}{n} \sum_{x=path_1}^{path_n} \sum_{z=1}^{length(x)} \frac{h(z)}{length(x)}, \tag{11}$$

where $h(z) = 1$ if subpath of size z from path x is predicted correctly from the model, and is zero otherwise. The overall accuracy is given by averaging the accuracy of all the n paths. Additionally, we compute the partial path accuracy

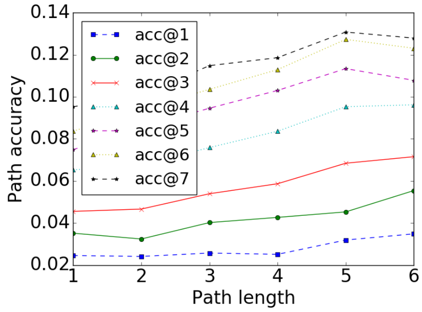


Fig. 1: Variation in partial path accuracy with k in accuracy@k

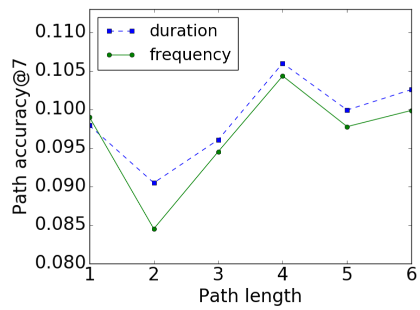


Fig. 2: Partial path accuracy for personalisation techniques

which assigns a score of 100% if at least one subpath of test path matches the recommended pair

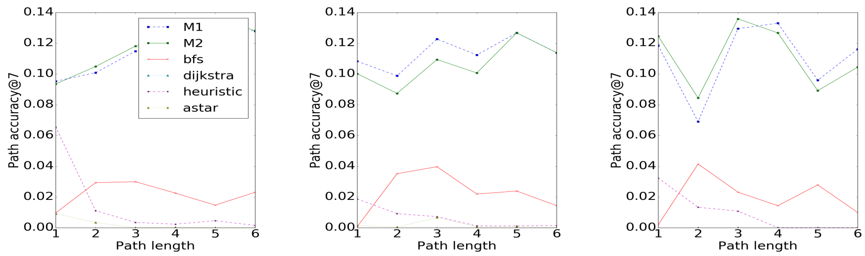
$$Acc_{partial} = \frac{1}{n} \sum_{x=path_1}^{path_n} \sum_{z=1}^{length(x)} \frac{\psi(h(z))}{length(x)}, \quad \psi(h(z)) = \begin{cases} length(x), \exists z \quad h(z) \geq 1 \\ 0, \quad otherwise \end{cases} \tag{12}$$

5.1 Baseline Comparison

We compare our approach with standard graph search algorithms as baselines and the non-personalised MDP policy. We start from the simplest, Breadth First Search (BFS) and evaluate more sophisticated algorithms of Dijkstra, Heuristic Search and A^*. For Dijkstra and A^*, the edge cost is given by the distance between the locations. For each of the baseline algorithms, we look for paths starting from l_{start} corresponding to the starting POI in the test set and iteratively choose a next POI to visit, till the last POI l_{end} in the itinerary is found. The heuristic used in A^* and heuristic search is the *Manhattan Distance* between the current place node and the goal node.

5.2 Results and Discussion

We first study the impact of path history on the prediction accuracy. Note that history length is the number of visited POIs encoded in the state, while path length is the number of next consecutive POIs to recommend. Path length of one hence stands for step-by-step recommendation. Table 1 captures the relation of path history to the performance of the system. There is a jump in performance as we change the path history from 1 to 2. Nonetheless, the performance shows very less improvement as the path history is increased up to length 5. This is primarily due to the fact that many of the travel sequences do not cover places more than three on a single day. Moreover, we observe that as the path history increases, the number of successors in the transition decreases.



(a) Munich (b) Paris (c) London

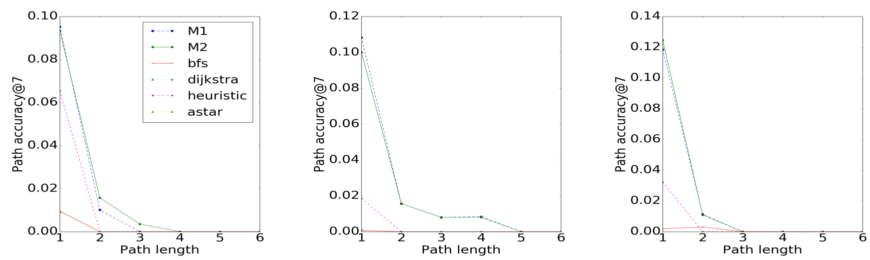
Fig. 3: Personalised Recommendation vs Baselines (Partial path accuracy)

In addition, we demonstrate the increase in performance by recommending the k closest places in Figure 1. All these places correspond to the optimal place category obtained through the value iteration algorithm. Thus, the more flexible a traveler is to multiple recommendation options at her current POI, the higher the likelihood of our system of recommending the best possible place entailing her travel preferences. We also compare the performance of the two personalising techniques, i.e., duration based and frequency based user interest as shown in Figure 2. The duration based personalisation consistently outperforms its counterpart that uses frequency based personalisation. This highlights the effectiveness of duration based personalisation in more accurately reflecting real-life tours of users, compared to the frequency based personalisation. Additionally, the personalisation factor α can be varied to balance the distance from the current state and the user personal interest. A value of $\alpha = 0.35$ gave the highest partial path accuracy during cross validation.

Furthermore, Figure 3 and 4 show the performance compare to baselines in terms of partial and exact accuracy, respectively. M1 denotes the personalised itinerary recommender system and M2 its non-personalised counterpart. Partial path accuracy@7 is used as evaluation measure. There is an average improvement of 10.5% of M1 over the path planning baselines, across the three cities. The effects of personalisation in M1 over the non-personalised recommendations in M2 is still not very significant in our experiments. However, there is a slight improvement for the shorter tour recommendations in Paris. One of the reasons for less impact of personalisation is the lack of data points in the training set as well as per user. Despite the limited quality data, sparsity of transitions, and minimal manual intervention in data processing, the computationally inexpensive MDP-based personalised recommender system beats the robust path planning algorithms and serves as a promising technique for modeling user behavior for travel recommendation.

6 Conclusion

We presented an MDP-based itinerary recommendation approach which took the sequential travel histories and preferences of users into account. An end-to-end itinerary planning system was constructed that uses both photo-sharing sites (Flickr) as well as the large abundance of geographical information on web-mapping services to extract supplementary knowledge for modeling a decision system based on reinforcement learning techniques. As opposed to many of systems proposed earlier, our model is not restricted to the geo-tagged pictures on Flickr. Our proposed approach tracks tourist movements from the time-stamps extracted from data, and recommends travel plans emulating the trip plan of a tourist. The empirical study showed that our proposed approach outperforms standard path planning algorithms.



(a) Munich (b) Paris (c) London

Fig. 4: Personalised Recommendation vs Baselines (Exact path accuracy)

References

1. Ashbrook, D., Starner, T.: Using gps to learn significant locations and predict movement across multiple users. Personal Ubiquitous Comput. **7**(5), 275–286 (Oct 2003)
2. Baba, Y., Ishikawa, F., Honiden, S.: Extraction of places related to flickr tags. In: Proceeding of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence. pp. 523–528. IOS Press (2010)
3. Cao, L., Yu, J., Luo, J., Huang, T.S.: Enhancing semantic and geographic annotation of web images via logistic canonical correlation regression. In: Proceedings of the 17th ACM International Conference on Multimedia. pp. 125–134. MM '09, ACM, New York, NY, USA (2009)
4. Crandall, D.J., Backstrom, L., Huttenlocher, D., Kleinberg, J.: Mapping the world's photos. In: Proceedings of the 18th International Conference on World Wide Web. pp. 761–770. WWW '09, ACM, New York, NY, USA (2009)
5. Dumais, S.T.: Latent semantic analysis. Annual review of information science and technology **38**(1), 188–230 (2004)
6. Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y.M., Yuan, Q.: Personalized ranking metric embedding for next new poi recommendation. In: IJCAI. pp. 2069–2075 (2015)
7. Jiang, S., Qian, X., Shen, J., Fu, Y., Mei, T.: Author topic model-based collaborative filtering for personalized poi recommendations. IEEE transactions on multimedia **17**(6), 907–918 (2015)
8. Kurashima, T., Iwata, T., Irie, G., Fujimura, K.: Travel route recommendation using geotags in photo sharing sites. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management. pp. 579–588. CIKM '10, ACM, New York, NY, USA (2010)

9. Levenshtein, V.: Binary codes capable of correcting spurious insertions and deletions of ones. Problems of information Transmission **1**(1), 8–17 (1965)
10. Lim, K.H.: Recommending and planning trip itineraries for individual travellers and groups of tourists. In: The 26th International Conference on Automated Planning and Scheduling. p. 115 (2016)
11. Lim, K.H., Chan, J., Leckie, C., Karunasekera, S.: Personalized tour recommendation based on user interests and points of interest visit durations. In: IJCAI. pp. 1778–1784 (2015)
12. Monreale, A., Pinelli, F., Trasarti, R., Giannotti, F.: Wherenext: a location predictor on trajectory pattern mining. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 637–646. ACM (2009)
13. Muntean, C.I., Nardini, F.M., Silvestri, F., Baraglia, R.: On learning prediction models for tourists paths. ACM Transactions on Intelligent Systems and Technology (TIST) **7**(1), 8 (2015)
14. Noulas, A., Scellato, S., Lathia, N., Mascolo, C.: Mining user mobility features for next place prediction in location-based services. In: Data mining (ICDM), 2012 IEEE 12th international conference on. pp. 1038–1043. IEEE (2012)
15. Popescu, A., Grefenstette, G., Moëllic, P.: Gazetiki: automatic construction of a geographical gazetteer. In: Proc. of JCDL (2008)
16. Quercia, D., Schifanella, R., Aiello, L.M.: 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. pp. 116–125. ACM (2014)
17. Rakesh, V., Jadhav, N., Kotov, A., Reddy, C.K.: Probabilistic social sequential model for tour recommendation. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. pp. 631–640. WSDM '17, ACM, New York, NY, USA (2017)
18. Rattenbury, T., Good, N., Naaman, M.: Towards automatic extraction of event and place semantics from flickr tags. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 103–110. ACM (2007)
19. Sang, J., Mei, T., Sun, J.T., Xu, C., Li, S.: Probabilistic sequential pois recommendation via check-in data. In: Proceedings of the 20th International Conference on Advances in Geographic Information Systems. pp. 402–405. ACM (2012)
20. Shani, G., Brafman, R.I., Heckerman, D.: An mdp-based recommender system. In: Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence. pp. 453–460. Morgan Kaufmann Publishers Inc. (2002)
21. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction, vol. 1. MIT press Cambridge (1998)
22. Tavakol, M., Brefeld, U.: Factored mdps for detecting topics of user sessions. In: Proceedings of the 8th ACM Conference on Recommender systems. pp. 33–40. ACM (2014)
23. Zhang, C., Liang, H., Wang, K., Sun, J.: 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. ACM (2015)
24. Zhang, W., Wang, J.: Location and time aware social collaborative retrieval for new successive point-of-interest recommendation. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. pp. 1221–1230. ACM (2015)