International Journal of Digital Earth

ISSN: 1753-8947 (Print) 1753-8955 (Online) Journal homepage: http://www.tandfonline.com/loi/tjde20

Building a model-based personalised recommendation approach for tourist attractions from geotagged social media data

Xiaoyu Sun, Zhou Huang, Xia Peng, Yiran Chen & Yu Liu

To cite this article: Xiaoyu Sun, Zhou Huang, Xia Peng, Yiran Chen & Yu Liu (2018): Building a model-based personalised recommendation approach for tourist attractions from geotagged social media data, International Journal of Digital Earth, DOI: 10.1080/17538947.2018.1471104

To link to this article: https://doi.org/10.1080/17538947.2018.1471104

Published online: 08 May 2018.

INTERNATIONAL JOURNAL OF DIGITAL EARTH, 2018
https://doi.org/10.1080/17538947.2018.1471104

Building a model-based personalised recommendation approach for tourist attractions from geotagged social media data

Xiaoyu Sun[a,b], Zhou Huang [a,b], Xia Peng[c], Yiran Chen [a,b] and Yu Liu [a,b]

[a]Institute of Remote Sensing and Geographical Information Systems, Peking University, Beijing, People's Republic of China; [b]Beijing Key Lab of Spatial Information Integration & Its Applications, Peking University, Beijing, People's Republic of China; [c]Collaborative Innovation Centre of eTourism, Tourism College, Beijing Union University, Beijing, People's Republic of China

ABSTRACT

When travelling, people are accustomed to taking and uploading photos on social media websites, which has led to the accumulation of huge numbers of geotagged photos. Combined with multisource information (e.g. weather, transportation, or textual information), these geotagged photos could help us in constructing user preference profiles at a high level of detail. Therefore, using these geotagged photos, we built a personalised recommendation system to provide attraction recommendations that match a user's preferences. Specifically, we retrieved a geotagged photo collection from the public API for Flickr (Flickr.com) and fetched a large amount of other contextual information to rebuild a user's travel history. We then created a model-based recommendation method with a two-stage architecture that consists of candidate generation (the matching process) and candidate ranking. In the matching process, we used a support vector machine model that was modified for multiclass classification to generate the candidate list. In addition, we used a gradient boosting regression tree to score each candidate and rerank the list. Finally, we evaluated our recommendation results with respect to accuracy and ranking ability. Compared with widely used memory-based methods, our proposed method performs significantly better in the cold-start situation and when mining 'long-tail' data.

ARTICLE HISTORY
Received 22 December 2017
Accepted 25 April 2018

KEYWORDS
Recommendation system; geotagged photos; social media; model-based approach; support vector machine (SVM); gradient boosting regression tree (GBRT)

1. Introduction

With the recent development of mobile Internet technology, people are accustomed to obtaining or sharing information through mobile intelligent terminal applications whenever and wherever possible. Among such information, in addition to text, pictures and videos, there is another type of important information: geographic location. In recent years, social media applications that use location-based services have become increasingly popular. Therefore, burgeoning numbers of location-based social media applications provide various and vast amounts of social media data containing geographic location information (Huang 2016). Users can share their geographic location information in several ways. For example, Foursquare (foursquare.com) and TripAdvisor (tripadvisor.com) allow users to 'check-in' and to comment on attractions, hotels, restaurants or any other location types. Similarly, when using Twitter (twitter.com) or Facebook (facebook.com), users can post their content while attaching a geotag. Some photo-sharing social media websites, such as Flickr (flickr.com) and Instagram (instagram.com), extract and publish geographic location information

CONTACT Zhou Huang ✉ huangzhou@pku.edu.cn

from photos uploaded by users; these photos are called geotagged photos (Clements et al. 2010). The amount of geotagged social media data has become enormous. For example, Flickr has an estimated 49 million geotagged photos. These huge amounts of data include not only geographical location information but also a large amount of user information, context information, text information and other information (Majid et al. 2013). Geotagged social media data are rich in information about individual behaviours and user characteristics, which facilitates the building of more innovative services to bring convenience to people's lives.

When visiting unfamiliar cities, tourists often need help in selecting attractions that match their personal interests from a large set of options. Because of the lack of information about a strange city, tourists almost always use travel websites or travelogues to obtain more specific information about a place's attractions (Majid et al. 2013). However, because this information is fragmentary and contains the subjective attitudes of authors, tourists may not be able to effectively determine which attractions would interest them most. Moreover, tourists are only able to obtain descriptions about the most famous attractions and may miss out on some unpopular ones that they would prefer in reality. Because trip planning is a time-consuming task, there is always a need for a travel recommendation system that can offer appropriate attraction suggestions according to personal interests and spatiotemporal context (Park, Hong, and Cho 2007).

In travel recommendation systems, a simple assumption is that users have specific interests and thus visit attractions matching their preferences. Furthermore, it is assumed that users like to take photos of an attraction that they like, and the more photos they take of it, the more the users are interested in this particular attraction (Kennedy et al. 2007). Therefore, much work so far has focused on retrieving users' travel history from their geotagged photo collection and then creating a personal interest profile for a recommendation system (Clements et al. 2010; Cheng et al. 2011; Majid et al. 2013; Huang 2016; Yang et al. 2017). However, most of these systems apply only traditional recommendation methods or made limited improvements. Moreover, they demonstrate high-level sparsity and a serious imbalance of social media data.

In contrast to applying traditional methods, we focus on developing a novel model-based recommendation approach based on geotagged photos. Specifically, we used Flickr's public API to fetch geotagged photos that contain a wealth of information and reflect the travel preferences of tourists. We then rebuilt a tourist's travel history and context information by gathering weather information through the public API of Weather Underground (wunderground.com). Most recommendation methods are based on collaborative filtering; in contrast, we proposed a model-based method that is assembled using a support vector machine (SVM) and a gradient boosting regression tree (GBRT). Finally, we used offline experiments to evaluate the proposed method against a method based on collaborative filtering. The results show that our method performs better and is robust to the cold-start and sparsity problems.

2. Related work

2.1. Recommendation methods

A recommendation system explicitly or implicitly collects information about users, items and user–item interactions to provide a user with a list of items that he or she prefers. The core technology of a recommendation system is the recommendation algorithm. According to a widely accepted taxonomy, recommendation methods can be divided into memory-based and model-based approaches (Adomavicius and Tuzhilin 2005; Su and Khoshgoftaar 2009; Bobadilla et al. 2013).

Memory-based methods focus on measuring the similarity of users or items using the user–item rating matrix and generating recommendation results by matrix manipulation (Bobadilla et al. 2013). The most classic memory-based method is collaborative filtering. Recently, researchers have concentrated on developing various K-nearest-neighbour methods based on collaborative filtering. For example, Clements et al. (2010) used a user-based collaborative filtering method to analyse a geotagged Flickr dataset and predict user behaviour. Memon et al. (2015) applied the kernel density estimation to model user's geographical travel preferences and then calculated the Kullback-Leibler divergence to represent the user similarity. Huang (2016) proposed three context-aware collaborative filtering methods to provide recommendation locations appropriate to the current weather. However, memory-based methods are still inefficient because they have to maintain an extremely large matrix and perform calculations on it.

Model-based methods use a user history dataset to train a model offline and then use the trained model to generate recommendations online. Model-based methods perform better in real time and are more stable than memory-based methods (Su and Khoshgoftaar 2009; Bobadilla et al. 2013). Commonly used models include Bayesian classifiers (Park, Hong, and Cho 2007; Cheng et al. 2011; Hsu, Lin, and Ho 2012), SVMs (Oku et al. 2006; Xu and Araki 2006), and neural networks (Kongsakun and Fung 2012; Tomar et al. 2014; Covington, Adams, and Sargin 2016). Cheng et al. (2011) proposed a route recommendation system based on a Bayesian classifier trained by a Flickr dataset. Covington, Adams, and Sargin (2016) trained two deep neural networks to recommend videos on YouTube's (youtube.com) online recommendation system. They focused on using feature engineering approaches to optimise the training process and presented a two-stage method to substantially improve performance. In addition, some researchers have investigated ensemble learning models to obtain more precise results. Subramaniyaswamy et al. (2015) proposed an algorithm combining the AdaBoost (adaptive boosting algorithm) classifier with a Bayesian classifier to build a travel recommendation system. Zhang (2014) used a random forest algorithm that is based on individual decision trees to predict users' ratings on a MovieLens dataset. Because model-based methods have strong generalisation ability, better stability, and quicker response, we decided to use a model-based method to construct our recommendation system.

2.2. Cold-start and sparsity problems

Recommendation systems always face two problems: the cold-start problem and the sparsity problem (Schafer et al. 2007). In the cold-start problem, the recommendation system is not able to give reliable results because of the initial lack of user ratings. Because the matrix is not updated frequently, it is hard for memory-based methods to measure similarity when new users or new items enter the system (Schein et al. 2002; Su and Khoshgoftaar 2009). As for model-based methods, the cold-start problem can be reduced to a certain extent because of their strong generalisation ability (Zhang 2014). A common strategy used to tackle the cold-start problem consists of adding additional information to the dataset to make recommendations based on these extra features (Bobadilla et al. 2013). The supplemental data almost always include users' personal information, item descriptions, users' historical actions on other datasets, and context information. Hence, we retrieved users' textual information, their personal resume on their homepage, and weather information that match their travel timestamp to enrich information dimensionality and mitigate the cold-start problem.

The sparsity problem often occurs when the number of users or items is huge. This causes interactions such as ratings to become extremely sparse. To reduce the sparsity problem, most existing studies use dimensionality reduction and clustering methods. These reduction methods, such as the latent factor model and singular value decomposition (Koren, Bell, and Volinsky 2009), are based on matrix factorisations. Used to improve collaborative filtering, the aim of matrix factorisation is to reduce the dimensions of the user–item matrix and enhance the calculation efficiency (Luo, Xia, and Zhu 2012, 2013). Clustering methods reduce the dimensionality of users or items by clustering algorithms such as K-means and DBSCAN, which can also partly alleviate the sparsity problem. It is typical to apply clustering methods to items. For example, Shinde and Kulkarni (2012) and Yao and Zhang (2009) both made use of item clustering methods to process high-level sparse data. Similarly, we used a spatial clustering method to find the locations of frequently photographed areas, which are treated as recommendation items (attractions) in the system.

2.3. Travel recommendation using geotagged social media data

Traditional recommendation systems, which are commonly designed for books, movies, music and news, are based on the user rating data of websites (Golbeck 2006; Ding 2010; Su et al. 2010). Although the rating data are well formed and from reliable sources, these explicit data are still updated slowly and lack a personal preference description. In contrast, geotagged social media data are implicit data that strongly reflect users' interests. Despite being noisy and unstructured, geotagged social media data are massive and take various forms. In particular, Flickr is globally one of the most popular photo-sharing websites. It maintains tens of millions of geotagged photos uploaded by travellers and photographers from all over the world. Therefore, many researchers have built travel recommendation systems based on Flickr photos.

Research on travel recommendation systems has focused on two types of recommendation: route recommendation and location recommendation. Route recommendation systems provide a route plan, which is in a time sequence, based on user histories. For example, De Choudhury et al. (2010) clustered Flickr photos to determine attractions and used a graph-based algorithm to construct recommended itineraries for tourists. Cai, Lee, and Lee (2018) applied a sequential pattern mining method on geo-tagged photos to discover users' semantic trajectory patterns. Another typical route recommendation method is developing Markov-based models (Cheng et al. 2011; Chen, Cheng, and Hsu 2013; Yamasaki, Gallagher, and Chen 2013). The authors used the Markov model to predict user's next destination based on users' attributes or spatio-temporal contexts, and finally output a route plan which consists of several destinations. The main purpose of route recommendation is to discover a user's travel pattern and then provide recommendations based on it. However, geotagged social media data are not an appropriate dataset for route recommendation. Because these geotagged photos are sparse and taken over large intervals of time, there is not enough information to determine an accurate travel pattern for most users. A GPS trajectory is much better for analysing user travel patterns because of its detail and continuousness (Cui, Luo, and Wang 2017).

In contrast, a location recommendation system generates just one recommended attraction or a list of candidate attractions that is not arranged in a time sequence. In addition, location recommendations focus on creating a personal interest profile. Several studies have described approaches based on geotagged social media data for providing location recommendation. Cao et al. (2010) recommended popular locations via discovering representative tags and images, and several memory-based methods (Clements et al. 2010; Majid et al. 2013; Memon et al. 2015; Huang 2016) were used for location recommendation. They all focus on improving collaborative filtering based on geotagged photos and context information. However, they avoid the cold-start and sparsity problems that are caused by using social media data by abandoning sparse users. In contrast to the memory-based methods mentioned above, we applied a model-based method based on geotagged Flickr photos to address the cold-start and sparsity problems.

3. Methodology

3.1. Data acquisition and preprocessing

We used Flickr's public API to retrieve geotagged photos taken in Beijing from 1 January 2005 to 1 January 2016. After removing incomplete data, we harvested a total of 213,938 geotagged photos. Figure 1 shows the spatial distributions of these photos. Each geotagged photo contains metadata fields such as 'photoID', 'userID', 'tags', 'dateTaken', 'latitude' and 'longitude'. Sample records from these geotagged photos are given in Table 1. To further clean the dataset, we removed redundant photos, which occur when a few users take many photos of an attraction within a short time. Specifically, we regarded the photos taken by the same user in the same places within one hour as just one visit. To build users' history travel context, we used the Weather Underground API to fetch the historical weather data of Beijing over the same date interval as that of the Flickr dataset. The weather

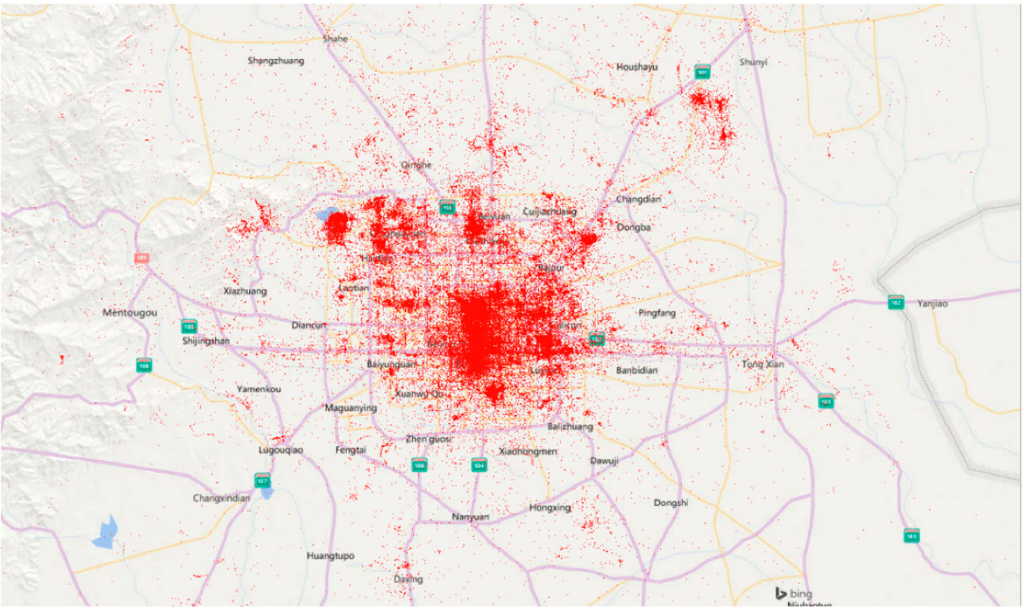

Figure 1. Spatial distribution of Flickr photos in Beijing from 1 January 2005 to 1 January 2016.

Table 1. Sample records from geotagged photos. Some attributes are not shown.

photoID	userID	tags	dateTaken	location	source	isPro
5580212429	60361833@N03	china, beijing, velvia, yuyuantanpark.	2005-04-13 12:37:17	116.316118 39.916239	San Diego, CA	True
1416289018	12772163@N06	china, beijing, forbiddencity.	2005-01-09 11:35:22	116.385726 39.917372	Melbourne, Australia	False

information includes the features ‘date’, ‘average_temperature’, ‘max_temperature’, ‘min_temperature’, ‘dew_point’, ‘precipitation’, ‘pressure’, ‘airspeed’, ‘view’ and ‘statement’. Table 2 shows several sample weather records gathered from Weather Underground.

When detecting travel attractions, we need to spatially cluster the photo collection. To find the popular tourist attractions, we applied a new adaptive spatial clustering algorithm proposed by Peng and Huang (2017). As a result, 90 clusters with more than 50 photos were extracted from the Flickr dataset. These clusters, as shown in Figure 2, are treated as the attractions used in the recommendation system. Each photo matches a cluster centre and is labelled by the corresponding attraction index. Then, we filtered out noisy points using a threshold for point density. The final dataset contains 80,440 visits of 11,173 users.

3.2. System overview

The following definitions are used to formalise the task in the proposed recommendation system.

Definition 1 (geotagged photo): Each geotagged photo p is defined as a tuple $p = (u, loc, t, tag)$, where u represents the user who uploaded photo p, loc is the location

Table 2. Sample weather records gathered from Weather Underground.

date	temp_avg (°C)	dew (°C)	prec (mm)	pres (hpa)	air_speed (km/h)	statement
2005-02-17 00:00:00	−2	−6	1	1032.94	5	snow
2014-10-08 00:00:00	16	12	0.2	1019.05	4	rain

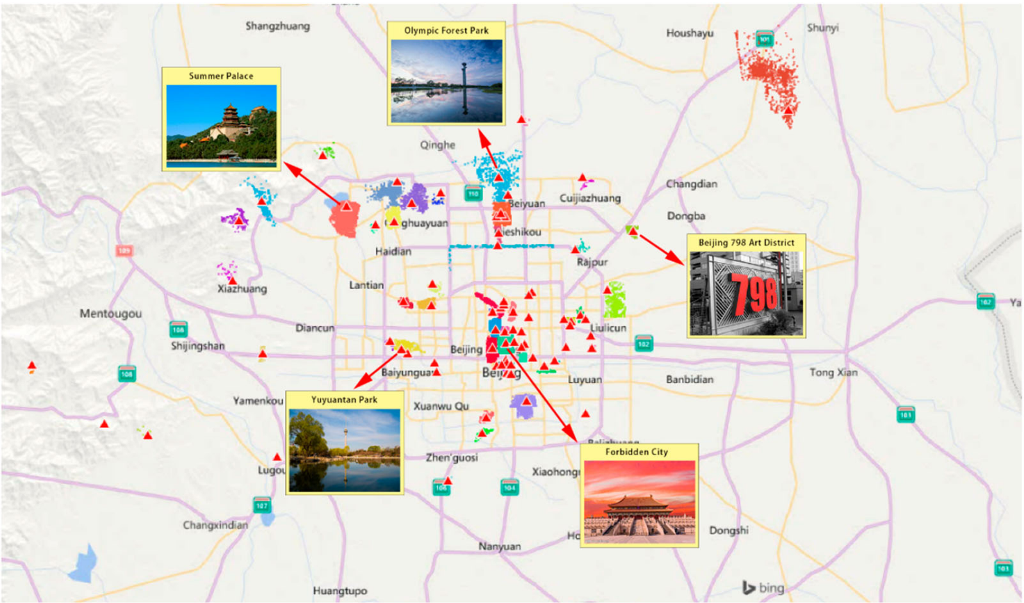

Figure 2. Popular tourist attractions in Beijing extracted from the Flickr dataset.

information consisting of the longitude and latitude, t is the upload timestamp, and *tag* is the textual tag information added by the user. All the geotagged photos form a photo collection P. In particular, P_u denotes the photo collection of individual user u.

Definition 2 (attraction collection): $L = \{l_1, l_2, \ldots, l_N\}$ denotes an attraction collection of size N. Every attraction l represents a clustering centre extracted from photo collection P containing location information *loc*. L_u denotes the attraction collection that user u has visited. Here attraction is not only a traditional tourist spot, but could also be a garden, a shopping centre, or even just a street area. For instance, we extracted Sanlitun Village (a famous pedestrian street) and Factory 798 (a well-known art district) from the Flickr photo collection, which are minor attractions compared with traditional popular spots such as Tiananmen Square and the Forbidden City.

Definition 3 (context information): A specific photo p matches an attraction l that is generated by clustered photo collection P. Using the timestamp t of photo p, the associated weather information can be gathered from the Weather Underground API. Combining the timestamp with weather information, it is able to construct context information cx, which includes the following:

(1) *Season*: spring (March–May), summer (June–August), autumn (September–November) and winter (December–February).
(2) *Time of day*: morning (6:00–12:00), afternoon (12:00–18:00), night (18:00–24:00) and early morning (0:00–6:00).
(3) *Temperature*: cold (<5°C), cool (5–15 °C), warm (15–25 °C) and hot ($\geq$ 25°C).
(4) *Weather*: sunny, lightly rainy or snowy, stormy, or heavy snow.

Definition 4 (user visit): A single photo p uploaded by user u is regarded as a visit v of u. A visit made by u is represented as a tuple $v = (u, l, cx, tag, t)$, where l is the attraction matching p, cx is the context information of photo p, tag is the textual tag attached to p, and t is the timestamp. In addition, all visits of user u are denoted as V_u.

Using the above definitions, the task of a personalised touristic attraction recommendation system can be formalised as follows: given a photo collection P_u of user u, extract all visits V_u with their contexts and recommend a set of attractions L_{Result}, where $L_{Result} \subseteq L - L_u$. To reach this goal, we proposed a model-based recommendation system architecture. As shown in Figure 3, our method uses a classic two-stage information retrieval process: candidate generation (or matching) and candidate ranking.

First, a multiclass SVM classifier (Cortes and Vapnik 1995) is designed to generate candidates from the collection of all attractions. The travel history data of a user is used as the input data, and the matching model is expected to provide a list of probable candidates after training with the selected dataset. The prediction process of the multiclass SVM classifier is similar to collaborative filtering; both perform a top-n recommendation. However, the former is based on a learning model, and the latter is memory based. In addition, the matching process generates only broadly personalised recommendations according to user travel histories and disregards many other types of information such as the weather context.

To consider more features, a regression model using GBRT (Schonlau 2005) is employed in the ranking process. Context data including weather information and textual tags posted by users are used to enrich the description of user interests. The ranking model outputs the candidate result in a fine-grained way by rating and then reranking all the candidates. In this part, we focus on adding more features into the model to discover subtler and deeper user preferences.

This SVM-GBRT model allows us to gather a small subset of attractions from a large corpus and then rerank this candidate list to improve the personalisation ability of the system. Moreover, the two parts of this model respectively focus on user travel histories and context information. For this

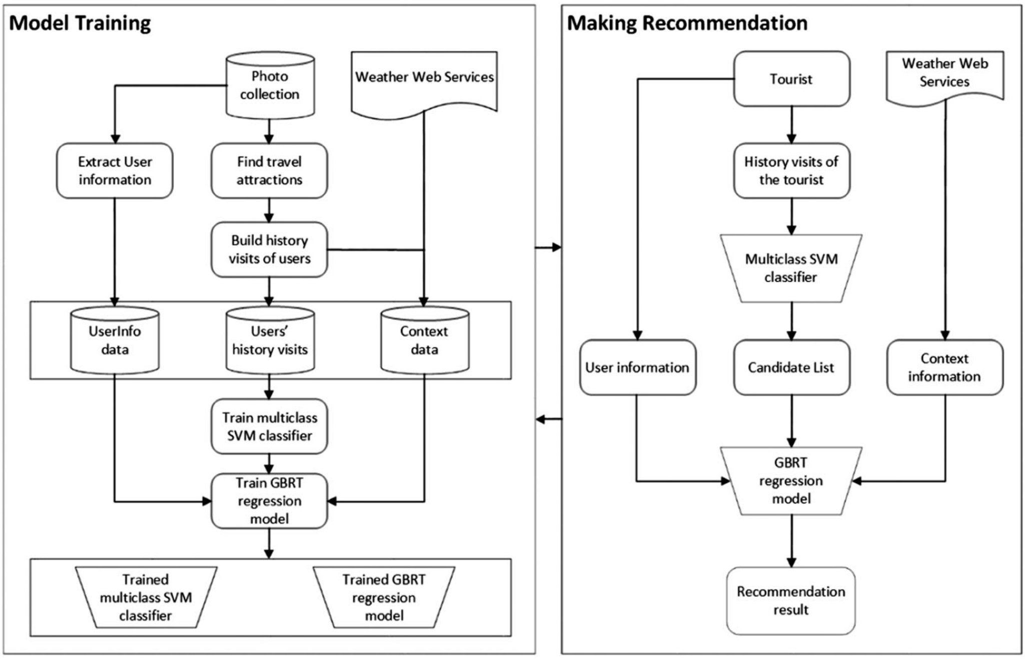

Figure 3. Architecture of the proposed SVM-GBRT recommendation method.

reason, we can train the two models separately. Hence, when new features are entered into the data, there is no need to retrain the whole model.

3.3. Matching process

During matching, several candidates that may be of interest to a user are retrieved from the overall attraction collection. Specifically, the task is defined as follows: according to individual travel history attraction collection L_u, which is extracted from visits V_u of user u, we generate a candidate list $L_{candidate}$ containing the n attractions that are most likely to be visited by user u. We pose this process as a multiclass classification problem where the recommendation process is to predict n class labels using attraction collection L_u as input. In particular, each class label matches an attraction l. The goal of our training process is to develop a classifier, $L_{candidate} = Classifier\,(L_u)$.

3.3.1. SVM classifier and multiclass classification

Formally, an SVM separates data points by a hyperplane or sets of hyperplanes, which are represented as $\omega^T x + b = 0$, where ω^T is the vector normal to the hyperplane and b is the offset of the hyperplane. The SVM training process can be represented as an optimisation problem of finding the hyperplane that ensures the distance between different kinds of data points is as large as possible. For linearly separable data, the objective function of this problem is as follows:

$$\max \frac{1}{||\omega||}, \ \text{s.t.}, \ y_i(\omega^T x_{\mathrm{i}} + b) \geq 1, \quad i = 1, \ldots, n \tag{1}$$

Here, under the constraint condition $y_i(\omega^T x + b) \geq 1, \ i = 1, \ldots, n$, the value of margin $1/||\omega||$ should be maximised. To solve the optimisation problem with the constraint, the Lagrangian multiplier method is applied to the objective function, and a slack variable is introduced to allow errors and approximation. The Lagrangian dual of Equation (1) is as follows:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j,=1}^{n} \alpha_i \alpha_j y_i y_j x_i, x_j \tag{2}$$

$$\text{s.t.}, \ 0 \leq \alpha_i \leq C, \quad i = 1, \ \ldots, \ n$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

where α_i is the Lagrangian multiplier of each data point, C is the slack variable to penalise the wrong classified data and control the maximum margin, and $\langle x_i, x_j \rangle$ is the inner product of vector x_i and vector x_j. Moreover, we can apply a kernel trick on the inner product $\langle x_i, x_j \rangle$ to make the linear SVM appropriate for nonlinear classification. To solve the dual Equation (2) and determine the hyperplane, we only need to calculate each value of α using several widely used algorithms.

Clearly, the SVM is a binary classifier that divides data into two classes by a hyperplane. However, in the top-n recommendation application scenario, there is a need to provide several items as output. Therefore, a multiclass classifier has to be applied here rather than a binary classifier. Hence, the one-vs.-rest (OvR) strategy is used to reduce a multiclass classification problem into multiple binary classifications. For an N-class problem, the OvR strategy involves training a single binary classifier for each class, with samples of that class as positive and the rest of the samples of all the other classes as negative. Moreover, this strategy requires the binary classifier to produce a confidence score for the result rather than just a class label. When predicting, the final decision consists of those classes that report the highest confidence score. In addition, when the amount of data is overwhelming, it is easy to accelerate by training each binary classifier in parallel.

3.3.2. Training the SVM classifier and predicting

Before training, it is necessary to preprocess the raw data. The training process is described by Algorithm 1. For the set of all training users, which is denoted as U_{train}, we use the history visits V_u of each user $u \in U_{train}$ to derive the specific attraction collection L_u (line 2). The training dataset is formed as $DataSet_{SVM,\ train} = \{(L'_u, label)|\ u \in U_{train}, L'_u = L_u - \{label\},\ label \in L_u\}$, which means that for a training user u, each attraction l from collection L_u is denoted as the label of the sample and the subset formed by the other attractions is taken as input data L'_u (line 5). For example, if a training user visited three attractions, there should be three training samples for him or her. The test dataset $DataSet_{SVM,\ test}$ is generated in the same way as the training dataset.

ALGORITHM 1 Generating the Training Dataset for the Multiclass SVM Classifier

```
Input: $U_{train}$, $V$
Output: $DataSet_{SVM,\ train}$
1.  Foreach $u$ in $U_{train}$ Do
2.      EXTRACT $L_u$ From $V_u$
3.      Foreach $l$ in $L_u$ Do
4.          $label = l$
5.          $L'_u = L_u - \{l\}$
6.          Add $(L'_u,\ label)$ to $DataSet_{SVM,\ train}$
7.      End Foreach
8.  End Foreach
```

Note that the SVM classifier takes input with a fixed dimensionality; hence, a user's attraction collection L'_u should be encoded as a fixed-dimensional vector Vec. Specifically, one-shot encoding is used to represent an attraction l as a binary vector that has only one single high bit (all the others are low). For instance, a group of three attractions can be represented as $\{[1, 0, 0], [0, 1, 0], [0, 0, 1]\}$. Then, the travel history can be vectorised by the sum of the encoded attraction vectors. If a user visited the first and the third attractions in this group, his or her travel history L_u would be $[1, 0, 1]$.

In the training process, which is shown in Algorithm 2, if the data include N attractions, N SVM classifiers must be trained in line with the OvR strategy. Each classifier corresponds to a specific attraction l. For each training sample, if $label = l$, the training sample can be rewritten as $(L'_u,\ 1)$, where L'_u is an N-dimensional vector (lines 4–5). If $label\ \neq l$, it can be denoted as $(L'_u,\ 0)$ (lines 6–7). Each SVM classifier is trained separately (line 10), and finally the classifiers are combined into a multiclass SVM classifier (line 11).

ALGORITHM 2 Training the Multiclass SVM Classifier

```
Input: $U_{train}$, $L$, $DataSet_{SVM,\ train}$
Output: $Multiclass\_SVM\_Classifier$
1.  Foreach $l$ in $L$ Do
2.      Foreach $(L'_u,\ label)$ in $DataSet_{SVM,\ train}$ Do
3.          Encode $L'_u$ into $Vec$
4.          If $label == l$
5.              Add $(Vec,\ 1)$ to $DataSet_l$
6.          Else
7.              Add $(Vec,\ 0)$ to $DataSet_l$
8.          End If
9.      End Foreach
10.     $SVM\_Classifier_l$ fit $DataSet_l$
11.     Add $SVM\_Classifier_l$ to $Multiclass\_SVM\_Classifier$
12. End Foreach
```

When predicting the candidate attractions, $Prob_l$ (i.e. N prediction confidence values for each attraction) can be obtained after inputting the test user's historical attraction collection L'_u. A candidate list $L_{candidate}$ is then generated by the top-n values of $Prob_l$ and the corresponding attractions.

3.4. Ranking process

In Section 3.3, a multiclass SVM classifier is applied to generate the candidate list $L_{candidate}$, which includes each candidate attraction l and its confidence value $Prob_l$. However, this matching process depends only on the users' travel history L_u, which is inadequate for reflecting personal interests. This is because we need to rerank the candidate list by using many more features that describe the user and the attraction. We pose the ranking process as a regression problem. In other words, a regression model is trained to rate each candidate and rerank the list. Formally, the ranking process is as follows: a regression model is trained to predict the preference of specific user u to each attraction l in the candidate list, i.e. $Pref(u, l) = Regression(Prob_l, cx, tag, \ldots)$, where $l \in L_{candidate}$.

3.4.1. Modelling user preferences

A regression model is often used to obtain a numerical prediction because it is an important part of machine learning. However, as explained in Section 3.3, the SVM classifier also provides confidence value $Prob_l$, which is the by-product of classification and is only based on travel histories without much more information. Hence, we must determine which numerical data we want to predict and measure preference $Pref(u,\ l)$.

Covington, Adams, and Sargin (2016) described an approach in a modelling regression problem for YouTube video recommendation. They predicted the expected video watched time and used it to rank candidate videos. In this study, the number of photos taken by a user at an attraction is used to infer the user's interest in this attraction. Hence, the preference is defined as follows:

$$Pref(u, l) = \frac{Num\,(P_{u,\,l})}{\sum_{i \in L_u} Num\,(P_{u,\,i})} \tag{3}$$

where $Num\,(P_{u,\,i})$ represents the photo count of user u for attraction i. Because it is in the form of a ratio, the user preference can be calculated without the effects caused by the difference between frequent photographers and occasional photographers.

3.4.2. GBRT regression model

GBRT is an ensemble method based on the classification and regression tree (CART) algorithm (Breiman et al. 1984), which is a weak learner. Like other boosting methods, GBRT trains several CART base learners through multiple iterations and finally produces a strong learner as a linear combination of these weak learners. The CART algorithm uses recursive binary splitting to generate a binary decision tree. In the splitting step, for discrete values, the data space is divided according to whether the data equal x or not. For continuous values, the split point value is calculated as the middle value of a range of data values, i.e. $(x\,[i] + x\,[i+1])/2$. To determine the best split value, the CART algorithm uses the Gini impurity, which measures the probability that a randomly chosen element from the set would be incorrectly labelled if it was randomly labelled according to the distribution of labels in the subset (Tan, Steinbach, and Kumar 2005).

A typical boosting algorithm is AdaBoost, which focuses on the data samples misclassified by weak learners and then updates the weight of samples to train subsequent learners. Analogously, GBRT uses weak learners to fit the residual of the previous step. The loss of samples can be iteratively reduced, and the final boosted learner is complete. In the n-th iteration, the combined model of the previous step is $f_{t-1}(x)$, and its loss function is $L(y, f_{t-1}(x))$. Our goal in the n-th step is to fit a CART weak learner $h_t(x)$ to minimise loss function $L(y, f_{t-1}(x) + h_t(x))$ and generate a new model $f_t(x)$ as a linear combination of $f_{t-1}(x)$ and $h_t(x)$.

Because GBRT is based on the CART algorithm, it is insensitive to data type. Both continuous and discrete data can be input into it. In contrast, as an ensemble learning method, GBRT performs better than any other weak learners, and its generalisation ability is stronger.

3.4.3. Training GBRT and predicting

Algorithm 3 presents the method for generating a training dataset for the GBRT ranking model.

In the preprocessing stage, a topic model is first used to deal with the feature *tag*, which is the textual tag information of a user's photo collection P_u. Each document is formed by all tag texts of a specific user u, and the document collection is collected from all training users. The latent Dirichlet allocation (LDA) model is applied to analyse the document collection and then extract the topic vector *topic* of each user u when training and predicting (line 11).

As described in Section 3.3, a user's travel history is represented as an N-dimensional vector L'_u, which denotes the visited attraction as 1 and the others as 0. In the ranking process, the travel history information is expected to be at a fine level of detail. Therefore, we should not arbitrarily use the binary encoding; instead, the ratio of user u's photos taken of attraction P_l to the total number of photos P_u he or she has taken is denoted as the travel interest in each attraction. In addition, an N-dimensional vector *Vec* is formed to represent a user's travel history including his or her preference for each attraction (line 2).

For a fine-level regression, as many features as possible are used to describe users and attractions. For users, the following features are chosen: *isLocal* indicates whether the user is a native user, *isPro* indicates whether the user is a professional photographer, and $Num\,(P_u)$ indicates the total number of photos uploaded by the user on Flickr. For attractions, all photos taken at attraction l are counted, which is indicated by $Num\,(P_l)$. Note that continuous features must be normalised because the gradient descent step is sensitive to the scaling and distribution of input data. A raw feature x with distribution f could be transformed by cumulative distribution normalisation, which is $x^* = \int_{-\infty}^{x} df$ (line 13).

When training GBRT, the user preference $Pref(u,\ l)$ of each attraction l in the candidate list $L_{candidate}$ is calculated, which is taken as the training label (line 3). The training data samples can be presented as follows: $DataSet_{GBRT,\,train} = \{([Vec, Prob_l, cx, topic,\ isLocal,\ isPro,\ Num(P_u),\ Num(P_l)], Pref(u,\ l))|u \in U_{train}, l \in L_{candidate}\}$.

ALGORITHM 3 Generating the Training Dataset for GBRT

```
Input: $DataSet_{SVM,\,train}$, $Multiclass\_SVM\_Classifier$, trained LDA model, WeatherData
Output: $DataSet_{GBRT,\,train}$
1.  Foreach ($L'_u$, $l$) in $DataSet_{SVM,\,train}$ Do
2.      Encode $L'_u$ into Vec
3.      $Pref(u,\ l) = Num(P_{u,\,l})/Num(P_u)$
4.      $L_{candidate} = Multiclass\_SVM\_Classifier(L_u)$
5.      If $l$ in $L_{candidate}$
6.          $Prob_l = L_{candidate}[l].confidenceValue$
7.      Else
8.          $Prob_l = 0$
9.      End If
10.     Collect $doc_u$ from $P_u.tag$
11.     $topic = LDA(doc_u)$
12.     Extract cx from WeatherData by timestamp
13.     Normalize $Num(P_u)$, $Num(P_l)$
14.     Add ($[Vec, Prob_l, cx, topic,\ isLocal,\ isPro,\ Num(P_u), Num(P_l)], Pref(u,\ l)$)
          to $DataSet_{GBRT,\,train}$
15. End Foreach
```

In contrast, in the prediction process, for each candidate attraction l in $L_{candidate}$, the prediction value of user preference $Pref(u,\ l)$ is obtained after inputting $[Vec, Prob_l, cx, topic,\ isLocal,\ isPro, P_{u,\,all}, P_{l,\,all}]$. Then, we can rerank $L_{candidate}$ by $Pref(u,\ l)$ and obtain the final result list L_{result} for target user u.

4. Experimental evaluation and discussion

4.1. Experimental setup and evaluation metrics

We partitioned the Flickr dataset by users: 80% were assigned to training users set U_{train}, and the remaining 20% were assigned to test user set U_{test}. As discussed in Section 3.3.2, because we extracted 90 attractions from the dataset, we embedded user travel histories into a 90-dimensional vector L_u, where visited attractions are marked as 1 and the others are 0. After determining each visited attraction l of user uB in order, the attraction index of l is regarded as a sample label, and the rest of the 90-dimensional vector L'_u is regarded as the input. In total, a user can provide $Num(L_u)$ samples for training and testing. Each sample is in the form of $(L'_u, label)$. All samples of U_{train} constitute the training dataset $DataSet_{SVM,\ train}$. This dataset was used to train the SVM models using the OvR strategy to obtain the final 90 SVM binary classifier corresponding to specific attraction l.

Before training the regression model, we first processed the feature *tag*. All the textual tag information of a user was used to form a document, and there are $Num\,(U_{train})$ documents for $Num\,(U_{train})$ users. After tokenisation and vectorisation of this document collection, we trained an LDA model to find the textual topic of the user and provided a vector *topic* for training and predicting.

When training the GBRT regression model, we first predicted a training sample in $DataSet_{SVM,\ train}$. Next, $L_{candidate}$ and its confidence values $Prob_l$ were extracted. As discussed in Section 3.4.1, for each candidate attraction l, we calculated the ratio of the photo count in attraction l to all photos, which was modelled as user preference $Pref(u, l)$. Combined with other multisource features, the final training sample was formed as $([Vec, Prob_l, cx, topic, isLocal, isPro, Num\,(P_u), Num\,(P_l)], Pref(u, l))$. We used the training dataset $DataSet_{GBRT,\ train}$ of all training samples and set the number of training iterations of 100 to train a GBRT regression model.

In the test process, for a test sample, we first used an SVM classifier to generate candidate list $L_{candidate}$ and kept the top five candidates with the highest confidence values. Then, we applied the trained GBRT model to predict the user's preference for each candidate and rerank $L_{candidate}$ as L_{result}. Finally, we evaluated the performance of the whole model by *label* of the samples and L_{result}.

For the evaluation, we need to choose appropriate metrics. Although precision and recall are widely accepted metrics for information retrieval system evaluation, Herlocker et al. (2004) argued that recall is impractical for evaluating top-n recommendations. Moreover, both precision and recall are incapable of measuring the ability of the ranking. Hence, in addition to precision, we used the mean average precision (MAP), which is a widely used metric to evaluate the ranking effectiveness of a recommendation system. For a top-n recommendation, $MAP@n$ is defined as follows:

$$MAP@n = \frac{\sum_{i=1}^{N_q} AP_i}{N_q}, AP_i = \frac{1}{r} \tag{4}$$

where N_q is the query count, AP_i is the average precision of the i-th query, and r is the rank of the correct label in the output list. Using this experimental setup, we built and evaluated our proposed SVM-GBRT model.

4.2. Results and discussion

We discuss the results from two aspects: (i) system performance on different users and (ii) performance for each attraction. To prove the effectiveness of our model, the following three methods are taken as baselines:

- Kullback-Leibler divergence Collaborative Filtering (KLCF) model: The KLCF model was proposed by Memon et al. (2015). Unlike the traditional collaborative filtering method, KLCF focuses on the geographic distribution of users' travel histories. The authors applied 2-dimensional kernel density estimation to model user's travel preferences, and then calculated the user similarity by the Kullback-Leibler divergence between users.

- Markov model with Seasonal and Temporal information (MST): The Markov model has been employed to predict the next destination by many studies (Cheng et al. 2011; Chen, Cheng, and Hsu 2013; Yamasaki, Gallagher, and Chen 2013). Although most of them focus on route planning, the Markov model could be applied on location recommendation as well. We used the MST model described by Yamasaki, Gallagher, and Chen (2013) as the baseline, which is developed for intra-city recommendation.
- Context-aware Collaborative Filtering (CACF) model: The CACF model was proposed by Huang (2016), which is a memory-based method developed from the classic collaborative filtering algorithm. The CACF model considers not only users' travel histories but also temporal and weather contexts.

4.2.1. Performance evaluation for different users

The number of attractions visited by user u is denoted as $Num\,(L_u)$, and a summary of user travel histories is shown in Table 3. We divided users into three different groups: 'short-route users' ($Num(L_u) < 2$), 'medium-route users' ($2 \leq Num(L_u) < 5$), and 'long-route users' ($Num(L_u) \geq 5$). Figure 4 shows the different spatial distributions of the three kinds of users.

We applied the SVM-GBRT and baseline models to these three user groups and evaluated the performance of the recommendations, as shown in Figure 5 and Tables 4 and 5. In general, as the length of the user route grows, the precision and $MAP@5$ of KLCF, CACF and SVM-GBRT both increase. This result illustrates that when a user's route is longer, more descriptions of the user are available to the recommendation system; therefore, the performance is improved. Moreover, this is consistent with the intuition that the more we know about a person's interests, the more likely we will be able to select items he or she prefers. In contrast, the MST model performs worse on long-route users, which indicates its lack of personalisation.

When predicting attractions for short-route users, there is only one attraction available for him or her. Marking this attraction as the sample label, the input 90-dimensional vector is an all-zero vector. In other words, the recommendation system is asked to provide results without any travel history of

Table 3. Summary of users and their average route length.

	$Num(L_u) < 2$	$2 \leq Num(L_u) < 5$	$Num(L_u) \geq 5$	All
$Num(U)$	3894	3551	3728	11,173
$AVG(Num(L_u))$	1.00	2.74	17.92	7.20

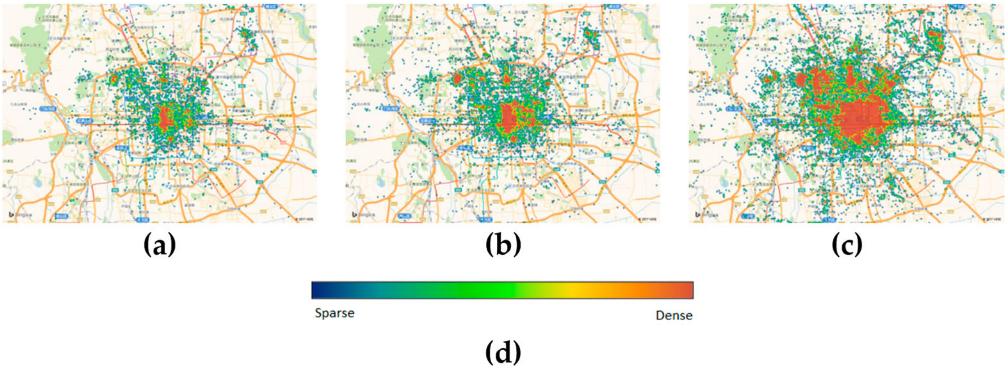

Figure 4. Spatial distribution heat map of three different kinds of users: (a) short-route user distribution, (b) medium-route user distribution, (c) long-route user distribution; (d) heat map legend, where cool colours represent sparse areas and warm colours represent dense areas.

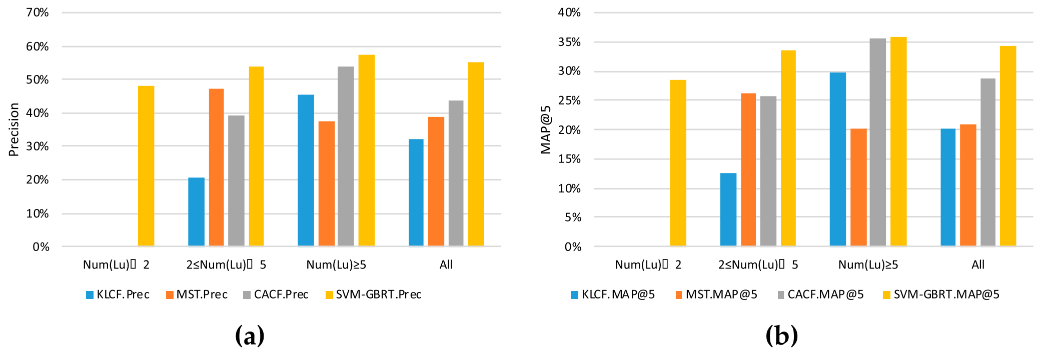

Figure 5. Performance comparison of the SVM-GBRT model with the baseline models: (a) precision and (b) MAP@5.

Table 4. Precision comparison of the SVM-GBRT and baseline models.

	KLCF.Prec	MST.Prec	CACF.Prec	SVM-GBRT.Prec
$Num(L_u) < 2$	/	/	/	47.94%
$2 \leq Num(L_u) < 5$	20.45%	47.02%	39.03%	54.05%
$Num(L_u) \geq 5$	45.36%	37.58%	54.02%	57.23%
All	31.99%	38.83%	43.64%	55.31%

Table 5. MAP@5 comparison of the SVM-GBRT and baseline models.

	KLCF.MAP@5	MST.MAP@5	CACF.MAP@5	SVM-GBRT.MAP@5
$Num(L_u) < 2$	/	/	/	28.42%
$2 \leq Num(L_u) < 5$	12.50%	26.25%	25.80%	33.48%
$Num(L_u) \geq 5$	29.83%	20.27%	35.63%	35.79%
All	20.27%	21.03%	28.79%	34.29%

the user. In such a scenario, the KLCF and CACF methods could not measure user similarity based on travel histories, and they generate a random result list. Similarly, the MST model could not calculate the transition probability without a starting point. However, for the SVM-GBRT model, the all-zero vector is also a feature vector that may be included in the training process. A partly reliable result list could be given, even though the history information is inadequate.

For medium-route users, the SVM-GBRT model achieves significantly better results than the KLCF, MST and CACF models, with improvements of 33.60%, 7.03% and 15.02% respectively for precision, and 20.98%, 7.23% and 7.68% respectively for *Map@5*. The reason why KLCF achieves such poor results is that the kernel density estimation could not model users' geographical preferences very well with sparse point data. In contrast, the MST model performs much better. Because medium-route users are traditional travellers, and most of them often choose the popular attractions. The MST model leans toward selecting such generally popular attractions, because the transition probabilities of them are much higher.

For long-route users, the difference between SVM-GBRT and CACF is not as notable, and the KLCF model also performs well. It can be inferred that for users who provide a rich history of information, the interest profile can be built by both methods. However, the MST model performs even worse on long-route users than on medium-route users. As discussed above, the MST model calculates the transition probability from one location to another. This Markov model only utilises the last location to predict the next one, ignoring the user's continuous travel history. Hence a lack of personalisation reduces its performance on long-route users.

In conclusion, the trajectory pattern mining method represented by MST is not suitable for personalised location recommendation, and it tends to discover common patterns rather than personalised preferences. Memory-based methods represented by KLCF and CACF are extremely dependent on the richness of the history information and can give acceptable results only when a user's history data are plentiful. In contrast with the memory-based method affected by the cold-start problem, the model-based method represented by SVM-GBRT demonstrates better generalisation ability and better adaptation of multisource information.

4.2.2. Performance evaluation for different attractions

A recommendation system is always expected to give novel and personalised results rather than generally popular results. To ensure that the proposed SVM-GBRT model is able to meet that expectation, we analysed the statistics for each attraction, which are shown in Figure 6.

For all 90 attractions, the distribution of visits to each attraction shows that there are a few generally popular attractions that are visited much more frequently than most other attractions. For example, more than 4000 users visited the Forbidden City, whereas only 39 users visited the National Art Museum of China. The significant imbalance in data is a significant challenge in recommendation. As shown in Figure 6, when user history information is not rich, the results are mainly occupied by generally popular attractions that are visited by more than 1000 users. As a user's route length increases, the proposed model gradually provides some novel attractions such as the Purple Bamboo Park and Prince Gong's Mansion. The SVM-GBRT model recommended 46 of 90 attractions, but KLCF and CACF recommended only 35 and 36 attractions respectively. How to mitigate the impact of generally popular items continues to be a problem for the methods based on collaborative filtering. Due to a lack of personalisation, the MST model recommended even only 11 popular attractions. However, because of the better generalisation ability, our proposed model-based method is more sensitive to long-tail data.

4.2.3. Summary and discussion

In the results shown in Section 4.2.1, the SVM-GBRT model performs better than the baseline models with respect to both precision and MAP. When users' history information is rich enough,

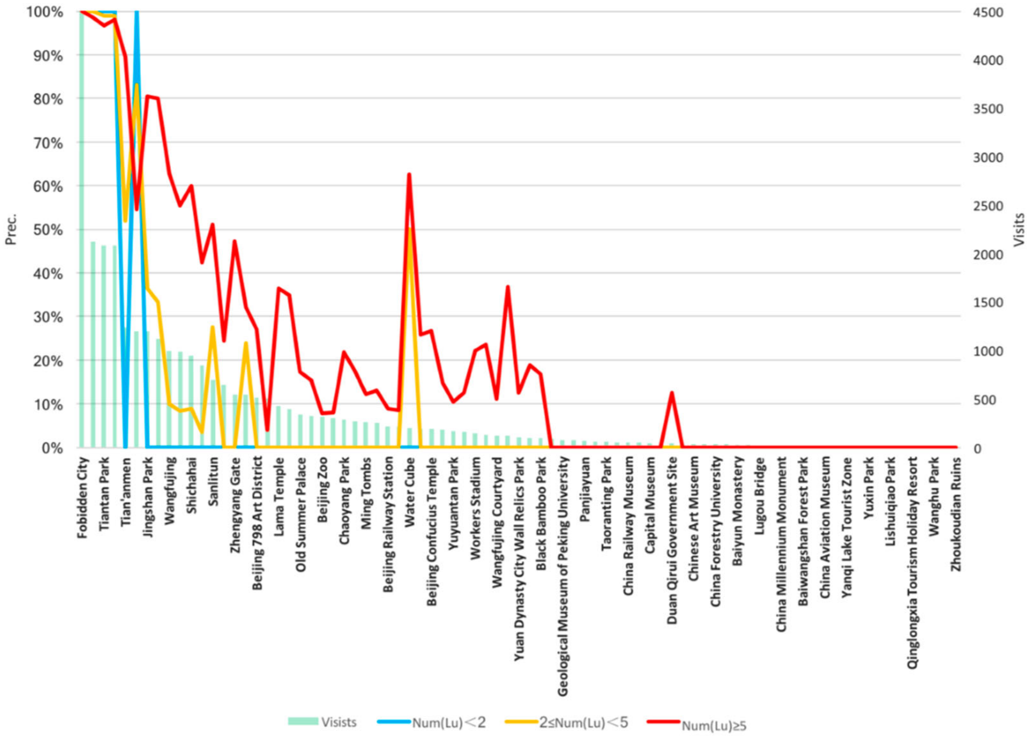

Figure 6. Precision of the SVM-GBRT model and visit distribution for each attraction.

the CACF model performs similarly to the SVM-GBRT model. As shown in Table 3, the average route length of long-route users is 17.92; however, it is 2.74 for medium-route users. This means that only when the length of a user route is substantially longer than that of other users can the CACF model infer a user's interest. Similarly, the KLCF model performs better on long-route users, which is consistent with the discussions on the cold-start problem by Bobadilla et al. (2013). Because memory-based methods rely much more on the volume and quality of samples, they perform poorly when sample data are not sufficient. As for the MST model, it is a feasible solution for mining common travel patterns of the whole group, rather than a personalised recommendation method. However, our proposed model-based method is able to make reliable recommendations because of its generalisation ability and stability.

Furthermore, most methods based on collaborative filtering are essentially in the form of the K-nearest-neighbour algorithm, and their performance particularly depends on the similarity metric. In Section 4.2.2, the results indicate that traditional methods such as CACF, KLCF and MST mostly provide generally popular attractions and are insensitive to long-tail data. To mitigate the impact of generally popular items, the CACF method uses TF-IDF. However, the introduction of such a penalty term is still subjective. If the penalty is too heavy, this significantly reduces the accuracy of the recommendation. On the other hand, if the penalty is too light, generally popular items will remain in the results. In contrast, our proposed model-based method is insensitive to generally popular items, which are treated as extreme data in the training, so that the penalty term is not needed.

The proposed method can still be improved in several ways. First, our method is based on the simple assumption that users take more photos at attractions that they like. This assumption can only provide positive samples and ignores negative samples. However, in the real world, we cannot determine arbitrarily that an attraction that has not been visited by a user is not in fact preferred by this user. Perhaps there are some reasons that led him or her to miss this attraction: for example, transportation delay, bad weather, or a tight schedule. In this study, because of the lack of negative feedback, we can only treat the interest in an attraction that the user has not visited as zero. This process can be improved by mining multisource data at a fine level; for instance, extracting negative judgements from users' textual information or applying sentiment analysis to the photos. Another limit in this study is that we only conducted an offline experiment. It would have been better to conduct an online A/B test to measure subtle changes in user feedback.

5. Conclusions

In this research, we proposed a model-based recommendation system, which has a two-stage architecture for matching and ranking. For the matching process, a multiclass SVM model is employed using the OvR strategy, and thus candidates are generated via inputting user travel histories. In the ranking process, using additional multisource information, a GBRT regression model is applied to score candidates and rerank the candidate list. Specifically, we discussed the multiclass problem in classification and described how to model the user preference of an attraction. Finally, we evaluated our model performance with respect to precision and MAP. The results indicate that our SVM-GBRT model is significantly better with respect to accuracy and ranking ability than conventional methods. Our model is less influenced by the cold-start problem and better at mining long-tail data.

However, due to the limitations in the quantity and quality of the dataset, the proposed method could have performed better. Moreover, the offline experiment could not comprehensively evaluate the recommendation performance, so that an online A/B test will be used in future work.

Acknowledgments

This research was supported by grants from the National Key Research and Development Program of China (2017YFB0503602), the National Natural Science Foundation of China (41771425, 41625003, 41501162), and the

Beijing Philosophy and Social Science Foundation (17JDGLB002). We also appreciate the detailed comments from the Editor and the anonymous reviewers.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This research was supported by grants from the National Key Research and Development Program of China [grant number 2017YFB0503602], the National Natural Science Foundation of China [grant number 41771425], [grant number 41625003], [grant number 41501162], and the Beijing Philosophy and Social Science Foundation [grant number 17JDGLB002]. We also appreciate the detailed comments from the Editor and the anonymous reviewers.

ORCID

Zhou Huang http://orcid.org/0000-0002-1255-1913
Yiran Chen http://orcid.org/0000-0001-6626-5413
Yu Liu http://orcid.org/0000-0002-0016-2902

References

Adomavicius, G., and A. Tuzhilin. 2005. "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-art and Possible Extensions." *IEEE Transactions on Knowledge and Data Engineering* 17 (6): 734–749. doi:10.1109/tkde.2005.99.

Bobadilla, J., F. Ortega, A. Hernando, and A. Gutierrez. 2013. "Recommender Systems Survey." *Knowledge-Based Systems* 46: 109–132. doi:10.1016/j.knosys.2013.03.012.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. 1984. *Classification and Regression Trees*, 17–23. Belmont: Wadsworth International Group.

Cai, G. C., K. Lee, and I. Lee. 2018. "Itinerary Recommender System with Semantic Trajectory Pattern Mining from Geo-tagged Photos." *Expert Systems with Applications* 94: 32–40.

Cao, L., J. Luo, A. Gallagher, X. Jin, J. Han, and T. S. Huang. 2010. "A Worldwide Tourism Recommendation System Based on Geotaggedweb Photos." IEEE International Conference on Acoustics Speech and Signal Processing, Dallas, TX, USA, 2274–2277.

Chen, Y. Y., A. J. Cheng, and W. H. Hsu. 2013. "Travel Recommendation by Mining People Attributes and Travel Group Types from Community-contributed Photos." *IEEE Transactions on Multimedia* 15 (6): 1283–1295.

Cheng, A. J., Y. Y. Chen, Y. T. Huang, W. H. Hsu, and H. Y. M. Liao. 2011. "Personalized Travel Recommendation by Mining People Attributes from Community-contributed Photos." International Conference on Multimedea 2011, Scottsdale, AZ, USA, November 28–December, 83–92. doi:10.1145/2072298.2072311.

Clements, M., P. Serdyukov, A. P. de Vries, and M. J. T. Reinders. 2010. "Using Flickr Geotags to Predict User Travel Behaviour." SIGIR 2010: Proceedings of the 33rd Annual International ACM Sigir Conference on Research Development in Information Retrieval, 851–852. doi:10.1145/1835449.1835648.

Cortes, C., and V. Vapnik. 1995. "Support-vector Networks." *Machine Learning* 20 (3): 273–297. doi:10.1007/bf00994018.

Covington, P., J. Adams, and E. Sargin. 2016. "Deep Neural Networks for YouTube Recommendations." ACM Conference on Recommender Systems, Boston, MA, USA, 191–198.

Cui, G., J. Luo, and X. Wang. 2017. "Personalized Travel Route Recommendation Using Collaborative Filtering Based on GPS Trajectories." *International Journal of Digital Earth* 12: 1–24. doi:10.1080/17538947.2017.1326535.

De Choudhury, M., M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, and C. Yu. 2010. "Automatic Construction of Travel Itineraries Using Social Breadcrumbs." Proceedings of the 21st ACM Conference on Hypertext and Hypermedia, Toronto, ON, Canada, 35–44.

Ding, X. 2010. "Research on the Book Intelligent Recommendation System Based on Data Mining." *Information Studies Theory & Application* 5: 29.

Golbeck, J. 2006. "Generating Predictive Movie Recommendations from Trust in Social Networks." International Conference on Trust Management, 93–104. doi:10.1007/11755593_8.

Herlocker, J. L., J. A. Konstan, K. Terveen, and J. T. Riedl. 2004. "Evaluating Collaborative Filtering Recommender Systems." *ACM Transactions on Information Systems* 22 (1): 5–53. doi:10.1145/963770.963772.

Hsu, F. M., Y. T. Lin, and T. K. Ho. 2012. “Design and Implementation of an Intelligent Recommendation System for Tourist Attractions: The Integration of EBM Model, Bayesian Network and Google Maps.” *Expert Systems with Applications* 39 (3): 3257–3264. doi:10.1016/j.eswa.2011.09.013.

Huang, H. S. 2016. “Context-aware Location Recommendation Using Geotagged Photos in Social Media.” *ISPRS International Journal of Geo-Information* 5 (11): 195. doi:10.3390/ijgi5110195.

Kennedy, L., M. Naaman, S. Ahern, R. Nair, and T. Rattenbury. 2007. “How Flickr Helps us Make Sense of the World: Context and Content in Community-contributed Media Collections.” ACM International Conference on Multimedia, Augsburg, Germany, 631–640.

Kongsakun, K., and C. C. Fung. 2012. “Neural Network Modeling for an Intelligent Recommendation System Supporting SRM for Universities in Thailand.” *WSEAS Transactions on Computers* 11 (2): 34–44.

Koren, Y., R. Bell, and C. Volinsky. 2009. “Matrix Factorization Techniques for Recommender Systems.” *Computer* 42 (8): 30–37. doi:10.1109/mc.2009.263.

Luo, X., Y. N. Xia, and Q. S. Zhu. 2012. “Incremental Collaborative Filtering Recommender Based on Regularized Matrix Factorization.” *Knowledge-Based Systems* 27: 271–280. doi:10.1016/j.knosys.2011.09.006.

Luo, X., Y. N. Xia, and Q. S. Zhu. 2013. “Applying the Learning Rate Adaptation to the Matrix Factorization Based Collaborative Filtering.” *Knowledge-Based Systems* 37: 154–164. doi:10.1016/j.knosys.2012.07.016.

Majid, A., L. Chen, G. C. Chen, H. T. Mirza, I. Hussain, and J. Woodward. 2013. “A Context-aware Personalized Travel Recommendation System Based on Geotagged Social Media Data Mining.” *International Journal of Geographical Information Science* 27 (4): 662–684. doi:10.1080/13658816.2012.696649.

Memon, I., L. Chen, A. Majid, M. Q. Lv, I. Hussain, and G. C. Chen. 2015. “Travel Recommendation Using Geo-tagged Photos in Social Media for Tourist.” *Wireless Personal Communications* 80 (4): 1347–1362.

Oku, K., S. Nakajima, J. Miyazaki, and S. Uemura. 2006. “Context-aware SVM for Context-dependent Information Recommendation.” The 7th International Conference on Mobile Data Management (MDM 2006), Nara, Japan, 109.

Park, M. H., J. H. Hong, and S. B. Cho. 2007. “Location-based Recommendation System Using Bayesian User’s Preference Model in Mobile Devices.” International Conference on Ubiquitous Intelligence and Computing, Hong Kong, China, 1130–1139.

Peng, X., and Z. Huang. 2017. “A Novel Popular Tourist Attraction Discovering Approach Based on Geo-tagged Social Media big Data.” *ISPRS International Journal of Geo-Information* 6 (7): 216. doi:10.3390/ijgi6070216.

Schafer, J. B., F. Dan, J. Herlocker, and S. Sen. 2007. *Collaborative Filtering Recommender Systems*, 291–324. Berlin: Springer.

Schein, A. I., A. Popescul, L. H. Ungar, and D. M. Pennock. 2002. “Methods and Metrics for Cold-start Recommendations.” International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, 253–260.

Schonlau, M. 2005. “Boosted Regression (Boosting): An Introductory Tutorial and a Stata Plugin.” *Stata Journal* 5 (3): 330–354.

Shinde, S. K., and U. Kulkarni. 2012. “Hybrid Personalized Recommender System Using Centering-bunching Based Clustering Algorithm.” *Expert Systems with Applications* 39 (1): 1381–1387. doi:10.1016/j.eswa.2011.08.020.

Su, X., and T. M. Khoshgoftaar. 2009. “A Survey of Collaborative Filtering Techniques.” *Advances in Artificial Intelligence* 2009 (4): 1–19.

Su, J. H., H. H. Yeh, P. S. Yu, and V. S. Tseng. 2010. “Music Recommendation Using Content and Context Information Mining.” *IEEE Intelligent Systems* 25 (1): 16–26. doi:10.1109/mis.2010.23.

Subramaniyaswamy, V., V. Vijayakumar, R. Logesh, and V. Indragandhi. 2015. “Intelligent Travel Recommendation System by Mining Attributes from Community Contributed Photos.” *Big Data, Cloud and Computing Challenges* 50: 447–455. doi:10.1016/j.procs.2015.04.014.

Tan, P. N., M. Steinbach, and V. Kumar. 2005. *Introduction to Data Mining*. 1st ed. Boston, MA: Addison-Wesley Longman.

Tomar, A., F. Godin, B. Vandersmissen, W. De Neve, and R. Van de Walle. 2014. “Towards Twitter Hashtag Recommendation using Distributed Word Representations and a Deep Feed Forward Neural Network.” 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), New Delhi, 362–368.

Xu, J. A., and K. Araki. 2006. “A SVM-based Personal Recommendation System for TV Programs.” 12th International Multi-Media Modelling Conference Proceedings, Beijing, 401–404.

Yamasaki, T., A. Gallagher, and T. Chen. 2013. “Personalized Intra- and Inter-city Travel Recommendation Using Large-scale Geotags.” Proceedings of the 2nd ACM international workshop on Geotagging and its Applications in Multimedia, Barcelona, Spain, 25–30.

Yang, L., L. Wu, Y. Liu, and C. Kang. 2017. “Quantifying Tourist Behavior Patterns by Travel Motifs and Geo-tagged Photos from Flickr.” *ISPRS International Journal of Geo-Information* 6 (11): 345. doi:10.3390/ijgi6110345.

Yao, Z., and Q. Zhang. 2009. “Item-based Clustering Collaborative Filtering Algorithm under High-dimensional Sparse Data.” International Joint Conference on Computational Sciences and Optimization (CSO 2009), Sanya, Hainan, China, 787–790.Zhang, H. 2014. “A Random Forest Approach to Model-based Recommendation.” *Journal of Information & Computational Science* 11 (15): 5341–5348. doi:10.12733/jics20104701.