

Research Article

Discovering Travel Spatiotemporal Pattern Based on Sequential Events Similarity

Juanjuan Chen ¹, Liying Huang ², Chengliang Wang ² and Nijia Zheng ¹

¹College of Computer and Information Science, Chongqing Normal University, Chongqing 401331, China

²College of Computer Science, Chongqing University, Chongqing 400044, China

Correspondence should be addressed to Juanjuan Chen; 20131036@cqnu.edu.cn

Received 2 November 2020; Revised 23 November 2020; Accepted 13 December 2020; Published 29 December 2020

Academic Editor: Chenquan Gan

Copyright © 2020 Juanjuan Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Travel route preferences can strongly interact with the events that happened in networked traveling, and this coevolving phenomena are essential in providing theoretical foundations for travel route recommendation and predicting collective behaviour in social systems. While most literature puts the focus on route recommendation of individual scenic spots instead of city travel, we propose a novel approach named City Travel Route Recommendation based on Sequential Events Similarity (CTRR-SES) by applying the coevolving spreading dynamics of the city tour networks and mine the travel spatiotemporal patterns in the networks. First, we present the Event Sequence Similarity Measurement Method based on modelling tourists' travel sequences. The method can help measure similarities in various city travel routes, which combine different scenic types, time slots, and relative locations. Second, by applying the user preference learning method based on scenic type, we learn from the user's city travel historical data and compute the personalized travel preference. Finally, we verify our algorithm by collecting data of 54 city travellers of their historical spatiotemporal routes in the ten most popular cities from Mafeng.com. CTRR-SES shows better performance in predicting the user's new city travel sequence fitting the user's individual preference.

1. Introduction

City tour has become popular in recent years as tourists may experience various food, culture, customs, and city views in this process while making use of commercial services like nice accommodation and inner-city transportation [1]. Unlike those traditional scenic spots, which are geographically isolated, a city tour combines civil resources, various facilities, and landscapes, and these form a city tour network with spatiotemporal multiplexity. Factors such as urban economy, society, and culture have an impact on the touring experience. They are coevolving through high relevance, so a city scenic spot has compound attributes of multiple labels. Furthermore, many ways of transport connect these city spots, which are geographically centered around the urban area. Thus, a city travel plan has the characteristics of personalization, flexibility, and evolving [2, 3], and the coevolving spreading dynamics of this network with multiscale structure is a great point of exploration that can apply

to the city tour recommendation system. So far, the travel recommendations given by apps and OTAs are classical routes with scenic spots ranked by the number of visitors or preferences of most travellers. Thus, the recommendations are not suitable for every visitor because of a lack of personalization [4]. When modelling and solving the tour route planning problem, most papers investigate user preference and give travel route recommendations with a fixed start and endpoints [5], not taking the spatiotemporal travel sequence, length of stay, and ways of transport into consideration [6, 7].

Because of these problems mentioned above, this paper firstly defines the user travel sequence model and various elements involved in city tour travel planning, then based on this model, we present the travel sequence similarity measurement method. The method can help measure the similarities of various city travel sequences, which combine different scenic types, time slots, and relative locations. Secondly, clustering analysis is conducted based on the

historical travel database. By using the travel sequence similarity measurement method, we compute the baseline model of an individual visitor's personalized travel preference. Finally, we propose a novel approach named City Travel Route Recommendation based on Sequential Events Similarity (CTRR-SES). CTRR-SES helps recommend personal travel routes to a new destination for users. The recommendation routes are a better fit for the user's preference as they are calculated by the travel preference baseline model and from the historical travel sequence data of the user.

2. Research Background

Travel route recommendation system gives the user city travel routes that match user's preferences, satisfying the user's real needs and expectations. Exploiting historical data of users to make future prediction lives at the heart of building effective recommender systems [8]. For e-commerce, some personalized recommendation strategies can be designed to promote the diffusion of products [9]. However, city tourism is a new product of modern social arrangements as tourists spend time in pursuit of recreation, relaxation, and pleasure in cities. City tourism is featured by social media posts and marks of hot city attractions. Many types of research investigate tour preference by studying traveller's social media posts and tag data. Based on Geo-tagged photos, some research on the correlation of several Geo-tagged images with an actual number of visitors [10], some on traveller's spatiotemporal behaviour [11], some on travel route recommendation system algorithm [12], and some on city impressions and big events and their combined impact on travel decision [13]. However, those papers do not fully consider the features of new city tourists and their touring preference sequences. Hence, they are unable to explore the unique traits of city travellers.

Big data about travel knowledge is generated each day on the Internet and various platforms. Large amounts of structured or semistructured datasets are produced by visitors who share their travel experiences, skills, or feedback through communication technology and mobile appliances. Upon travel route recommendation algorithm studies, Sun et al. use Knowledge Graph to build a travel database by extracting traveling information from the content submitted by the users, to represent personalized touring routes [14]. Li et al. present a new approach for designing tourist routes for tourists visiting Gulangyu island by applying the Stated Preference method [15].

However, those papers only study the recommendations of scenic spots, while they do not analyze the sequential order of spots in visitors' historical touring routes. We believe the sequential order plays an important role in measuring tourist preference. For example, Sequence 1 represents user A who visits urban Chongqing city, given as Ciqikou-Hongyadong-Jiefangbei-Sichuan Fine Art Institute-Eling Park. The sequence of user B is given as Sichuan Fine Art Institute-Eling Park-Hongyadong-Jiefangbei-Ciqikou. If we consider scenic spots as the plain factor to impact the visitor's preference, then it is obvious to give both

A and B the same recommendation of route sequence. However, users A and B visit those scenic spots in a different sequence, which indicates that user A prefers to spend daytime in spots tagged as shopping or fine food and night time for city sights, yet user B prefers to visit city sights in the day time and shop at night. We believe that travel route recommendations should include not only the user's preference for the scenic type but also the visiting sequence and time slot. Then the recommendation system may give users their personal travel routes matching their individual preferences.

This paper constructs the attraction of tourist city preference model based on the city attraction knowledge base and user's historical touring sequences. And a data mining algorithm is proposed to discover the city attraction label set. Traveller's historical touring events are analyzed to find clusters mostly reflecting traveller's preferences. By comparing the similarities of various travel event sequences, we aim to provide highly personalized travel recommendations that satisfy the traveller's real needs.

3. Preliminaries

Before the problem statement, we give the definitions of these concepts as follows.

Definition 1 (Attraction p). represents city places where visitors previously visited or are interested in visiting, and it could be a natural landscape, folk culture, historical landscape, civic landscape, or consuming place.

Definition 2 (Attraction labels **Profile** (p)). Given p as a city attraction, we define the labels of p as a sequence of Profile (p) = $\{p_{id}, p_{name}, p_{type}, p_{position}, p_{score}\}$.

Definition 3 (Travel history r). Given p as a city attraction, the travel history in p is given as a set of $r = (p, t_p^s, t_p^e)$, in which t_p^s as the time arriving p and t_p^e as the leaving time.

Definition 4 (Touring sequence L). We define the touring sequence as a time-ordered sequence of a user visiting multiple city attractions, given as $L = \{r_1, r_2, \dots, r_n\} = \{(p_1, t_{p_1}^s, t_{p_1}^e), (p_2, t_{p_2}^s, t_{p_2}^e), \dots, (p_n, t_{p_n}^s, t_{p_n}^e)\}$. where n represents the total number of attractions that have been visited. The time interval of visiting two adjacent attractions is no longer than a threshold value, denoted as $t_{p_{x+1}}^s - t_{p_x}^e < \varepsilon$. Considering the characteristics of city travel, we set a reasonable time interval threshold ε as 1 hour.

Based on the definitions above, we define our city travel route recommendation problem as follows. Given all users' historical touring sequences in the set $U = \{L_1, L_2, \dots, L_n\}$, input the historical touring sequence set $U_{userA} = \{L_{A1}, L_{A2}, \dots, L_{An}\}$ of user A, in which An denotes his/her total number of touring sequences. Then input city B. Our goal is to determine the best personalized city travel route recommendation for the user A from the travel sequence set of city B. The strategies are given as follows:

- (1) Learning from the user's historical city tour sequences, identify the user's city travel preference model
- (2) Based on all travel sequences in a given city and the user's city travel preference model, determine the best personalized city tour route recommendation for the user

4. Recommendation Algorithm

City tour recommendation is challenging to satisfy the visitor's preference and real needs when a tourist visits a new city. To meet this challenge, we propose a novel approach named City Travel Route Recommendation based on Sequential Events Similarity (CTRR-SES) by measuring the similarities of various city travel routes in a given city and learning from the user's historical city touring sequences.

4.1. Travel Route Recommendation Framework. There are three building blocks in our CTRR-SES, as indicated in Figure 1, which are Travel History/Sequences Construction, Scenic Type-based User Preferences Baseline Modelling, and Route Recommendation System. Travel History/Sequences Construction and User Preferences Baseline Model Learning are processed offline. By analyzing the user's open travel posts, we can obtain the user's historical city touring sequences. Then we may compute the baseline model from the travel history using clustering analysis. Route Recommendation is processed online. Firstly, CTRR-SES computes the feature vectors which represent user's travel characteristics from the city travel historical sequences and the preferences baseline model. Then it recommends the most similar touring sequence, which matches the user's personal preference from existing travel sequences.

4.2. Travel Knowledge Base and Touring Sequence Construction. Using data mining technology, we construct the travel knowledge base by obtaining big data from platforms like Baidu, Mafengwo, TripAdvisor, and Booking. Attraction information is comprised of attributes of Name, Geographic Position, Type and Rating, etc. Each attraction is also labelled with a category of one or many of the following, i.e., city park, garden, arboretum, natural landscape, architecture, church, temple, museum, college campus, historical sites, food and beverage, shopping site, amusement, art performance, etc. City tour transportation modes include Taxi, Bus, Subway, and Walk. Learning from the user's past space-time trajectory, travel sequences are generated by consecutively extracting data of geographic position, attraction label, visit duration, transportation mode, and time spent in transportation.

4.3. User Travel Preference Baseline Model Learning

4.3.1. Travel Sequence Similarity Measure. The travel sequence similarity measure is the measure by the proper algorithm of how much like multiple sequences are, which

then derive similar clusters. In this paper, we present the travel sequence similarity measurement method using the Needleman-Wunsch (NW) algorithm. Moreover, we improve the traditional NW algorithm by integrating time information in the Score Function.

Definition 5 (Attraction touring history similarity **W**). Given p as a city attraction, $r_i = (p_i, t_{p_i}^s, t_{p_i}^e)$ and $r_j = (p_j, t_{p_j}^s, t_{p_j}^e)$ are two variables in the travel sequence L . Then the similarity formula between r_i and r_j is given as follows:

$$W(r_i, r_j) = \begin{cases} u_1 S_{\text{poi}} + u_2 S_{\text{time}} (p_i \text{ and } p_j \text{ belong to a class}), \\ d_1 (p_i \text{ and } p_j \text{ do not belong to a class}), \\ d_2 (p_i \text{ or } p_j \text{ aligns to a gap}). \end{cases} \quad (1)$$

S_{poi} indicates the similarity between two POIs, and S_{time} indicates the similarity between the time visiting the two spots. u_1 and u_2 are different weights put on S_{poi} and S_{time} , which can adjust the sensitivity of S_{poi} and S_{time} . $u_1 + u_2 = 1$; d_1 and d_2 are customized scores.

Definition 6 (Travel sequence similarity **S**). Given two travel sequences $L_1 = \{r_{11}, \dots, r_{1n}\}$ and $L_2 = \{r_{21}, \dots, r_{2m}\}$, the similarity score $S(i, j)$ of $L_i = \{r_{11}, \dots, r_{1i}\}$ and $L_j = \{r_{21}, \dots, r_{2j}\}$ ($L_i \subseteq L_1, L_j \subseteq L_2$) is computed as follows:

$$S(i, j) = \max \begin{cases} S(i-1, j-1) + W(r_{1i}, r_{2j}), \\ S(i-1, j) + d_2, \\ S(i, j-1) + d_2, \end{cases} \quad (S(0, 0) = 0). \quad (2)$$

We can calculate the travel sequence similarity score matrix M . Normalization of data in the last row and column of the matrix generates the similarity scores of two travel sequences L_1 and L_2 . The pseudocode of the algorithm is given in Algorithm 1.

TSSA uses the method of iteration to calculate the similarity of two sequences L_1 and L_2 by comparing each item in the sequences. Then the value of similarity is stored in the 2-dimensional matrix M . If the lengths of the two sequences are not equal, then add a space gap to make them equal. The first to 6th lines in the TSSA initializes the similarity matrix, and the 7th to 18th lines conduct similarity calculation and fill in the matrix. $M[i][j]$ represents the similarity of two corresponding items in L_1 and L_2 , the value of which is determined by the values of $M[i-1][j]$, $M[i][j-1]$, and $M[i-1][j-1]$. Equation (2) is an iterative formula and gives three paths to calculate the values of $M[i][j]$, among which choosing the maximum value:

- (1) Obtain from above in the vertical line of $M[i][j]$. Sequences L_1 and L_2 are compared, then suppose L_1' and L_2' are generated during the comparison. Reaching the cell of $M[i][j]$ from above is equivalent to adding the corresponding items in L_2 to L_2' and

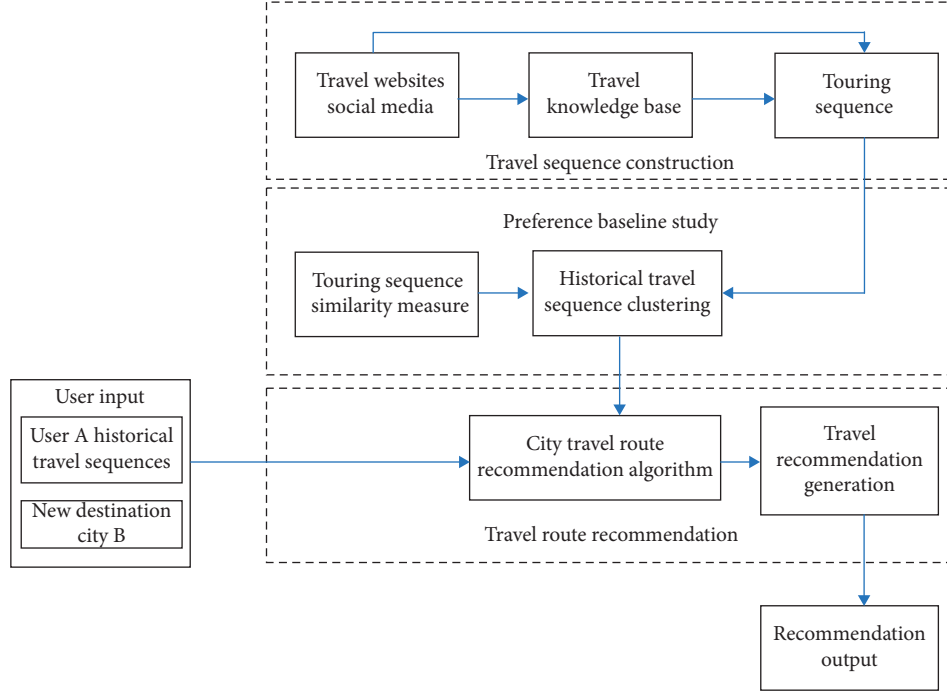


FIGURE 1: City travel route recommendation framework.

Input: Travel sequence L_1 and L_2

Output: Similarity of L_1 and L_2

Initialization: Set score matrix M to 0

```

(1) for  $i \leftarrow 0$  to  $|L_1|$  do
(2)    $M[i][0] \leftarrow i * d_2$ ;
(3) end for
(4) for  $j \leftarrow 0$  to  $|L_2|$  do
(5)    $M[0][j] \leftarrow j * d_2$ ;
(6) end for
(7) for  $i \leftarrow 1$  to  $|L_1|$  do
(8)   for  $j \leftarrow 1$  to  $|L_2|$  do
(9)     if Overlap ( $L_1[i].p.type, L_2[j].p.type$ ) then, //Overlap ( $a, b$ ) means the attraction type labels of POI  $a$  and POI  $b$  overlap
(10)       $S_{poi} \leftarrow PSA (L_1[i].p, L_2[j].p)$ ;
(11)       $S_{time} \leftarrow TSA (L_1[i].t, L_2[j].t)$ ;
(12)       $sim \leftarrow u * S_{poi} + (1 - u) * S_{time}$ ;
(13)       $M[i][j] \leftarrow \max(M[i-1][j-1] + sim, M[i-1][j] + d_2, M[i][j-1] + d_2)$ ;
(14)    else
(15)       $M[i][j] \leftarrow \max(M[i-1][j-1] + d_1, M[i-1][j] + d_2, M[i][j-1] + d_2)$ ;
(16)    end if
(17)  end for
(18) end for
(19) return  $M[|L_1|-1][|L_2|-1]$ ;

```

ALGORITHM 1: Travel sequence similarity algorithm (TSSA).

adding a gap in L'_1 . Therefore, the value of $M[i][j]$ is $M[i-1][j] + d_2$.

- (2) Obtain from the left in the horizontal line of $M[i][j]$. Same as (1), reaching the cell of $M[i][j]$ from the left is equivalent to adding the corresponding items in

L_1 to L'_1 and adding a gap in L'_2 . Thus, the value of $M[i][j]$ is $M[i][j-1] + d_2$.

- (3) Obtain from the diagonal line of $M[i][j]$. By adding the corresponding items in L_1 to L'_1 and adding the corresponding items in L_2 to L'_2 , we can calculate the

similarity value of $M[i-1][j-1] + W(L_1[i], L_2[j])$. When the i th POI label in L_1 is overlapped with the j th label in L_2 , the calculation of $W(L_1[i], L_2[j])$ is written in the 10th to 12th line in the TSSA; otherwise, the value of $W(L_1[i], L_2[j])$ equals d_1 .

The pseudocode of PSA (point similarity algorithm) in the 10th line and that of TSA (time similarity algorithm) in the 11th line are given in Algorithm 2 and Algorithm 3, respectively.

The first line in the algorithm calculates the intersection of scenic labels of two POIs. The second line measures the percent of an intersection of all labels added up and takes it as the similarity value of two POIs.

In the first line of the algorithm, we set half an hour as a single unit and then build the time axis based on it, and the time range spent in two POIs is indicated by two numeric sequences. In the second and third lines of the algorithm, the Longest Common Subsequence (LCS) algorithm is applied to find the longest subsequence present in both of the two numeric sequences. LCS can be solved using Dynamic Programming by dividing the original problem into some subproblems. The time similarity is the ratio of the length of the longest common subsequence to the length of the sequence.

4.3.2. Travel Sequences Clustering. The K -means algorithm is one of the most popular and widely used methods of clustering due to its simplicity, robustness, and speed. It is an iterative algorithm meaning that we repeat multiple steps making progress each time. Among many clustering algorithms, K -Means is also comparatively well known for its robustness as it is nonsensitive to noise and isolated points. K -means algorithm can deal with data sets of different types and discover clusters that are irrelevant with the input order of data. Thus, this paper adopts the K -Means algorithm for travel sequence clustering analysis.

(1) Clustering Algorithm Description. K -means algorithm partitions the dataset, which includes the number n data, into K number of clusters. Then the clusters are positioned as points, and all observations or data points are associated with the nearest cluster, computed, adjusted, and then the process starts overusing the new adjustments until the desired result is reached. The Travel Sequence Clustering Algorithm (TSCA) is given in Algorithm 4.

K clusters and a sequence containing K cluster centroids can be obtained by Algorithm 4. As each travel sequence reflects the traveller's preference, the base number will be great when adding those sequences altogether. Considering the meaning of centroids has great explaining value, so we set the sequence containing number K cluster centroids as the travel preference baseline model.

(2) Performance Evaluation of Sequence Clustering. Updated Sum of Squared Error (SSE) and Silhouette Coefficient (SC) is used in this paper to evaluate the performance of clustering.

Metric 1: SSE

SSE is a technique designed to find the sum of the squared error of sample points to centroids. Theoretically, the lower the SSE, then the better performance of clustering. This paper calculates the travel sequence similarity measure instead of a distance measure as the foundation of clustering. Therefore, the updated SSE is designed to find the sum of the similarity of sample points to centroids. Hence, the higher the updated SSE, theoretically, the better the performance of clustering.

Metric 2: SC

The Silhouette Coefficient is calculated using the mean intracluster distance $a(o)$ and the mean nearest cluster distance $b(o)$ for each sample o in D . To clarify, $b(o)$ is the distance between a sample and the nearest cluster that the sample is not part of. The calculation equation is given below:

$$SC(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}}. \quad (3)$$

The SC value ranges from -1 to 1 , and 1 means the clusters are well apart from each other and clearly distinguished. Just the other way round, when the updated SC value is close to -1 , the performance of clustering is better.

4.4. Travel Route Recommendation. Travel route recommendation requires the user to input his or her city travel historical sequences and a new destination city B . The user's travel preference is measured according to the relative distance between historical sequences and the preferences baseline sequence. We calculate the similarity between the city travel historical sequences and the preferences baseline model, and in the end compute the K -dimensional feature vectors which represent the user's travel preference, in which K represents the number of clustering. Therefore, we define user travel preference as follows.

Definition 7 (User travel preference **Userpre**). Given a user's travel history or sequence L_1, \dots, L_n (n is the number of travel sequences) and the preference referring sequence L_{k1}, \dots, L_{kk} , the travel preference is indicated by a K -dimensional vector as follows:

$$\text{Userpre} = \left(\frac{\sum_{i=1}^n \text{TSSA}(L_i, L_{k1})}{n}, \dots, \frac{\sum_{i=1}^n \text{TSSA}(L_i, L_{kk})}{n} \right). \quad (4)$$

In the same way, every travel history or sequence in city B can be indicated as a K -dimensional feature vector, in which we can find the vector that matches user A 's travel preference with the highest similarity degree. This is to say, that is the travel recommendation presented to user A because the travel sequence represented by the feature vector satisfies the user's travel preference. As Cosine Similarity (equation (5)) is a commonly used approach, we use this metric to measure the similarity of feature vectors:

Input: POI information p_1 and p_2 of travel item r_1 and r_2
Output: POI similarity S_{poi} of r_1 and r_2
(1) $\text{count} \leftarrow \text{Intersection}(p_1.\text{type}, p_2.\text{type});$
//Intersection (a, b) means the number of intersections of label a and label b
(2) $S_{\text{poi}} \leftarrow \text{count} / (p_1.\text{type.size} + p_2.\text{type.size} - \text{count});$
(3) **return** S_{poi}

ALGORITHM 2: Point similarity algorithm (PSA).

Input: Time information t_1 and t_2 of travel item r_1 and r_2 ;
Output: Time similarity S_{time} of r_1 and r_2 ;
(1) Divide the time axis by half an hour, and number from 1, then t_1 and t_2 can be represented by digital sequence l_1 and l_2
(2) $l = \text{LCS}(l_1, l_2)$ //Calculate the longest common subsequence of sequence l_1 and l_2
(3) $S_{\text{time}} \leftarrow |l| / (|l_1| + |l_2| - |l|)$
(4) **return** S_{time}

ALGORITHM 3: Time similarity algorithm (TSA).

$$c(x, y) = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}. \quad (5)$$

The travel sequence recommendation algorithm is given in Algorithm 5.

In the first line of the algorithm, we use the TSCA for travel history clustering analysis of all users. The array newMedoids stores the sequence containing K cluster centroids (K as the number of clusters). In the second line to the seventh line in the algorithm, we calculate the user's travel preference, and the K -dimensional feature vector is stored in the one-dimensional array userpre. From the eighth to the twelfth line, every travel history or sequence in the user's destination city can be indicated as a K -dimensional feature vector, which is stored in the size $m * k$ 2-dimensional array cityseq (m as the total number of all historical sequences in the destination city). In the fourteenth to the nineteenth line, we use the Cosine Similarity function CosSim to find the feature vector in cityseq that match the user's travel preference vector with the highest similarity degree. The result is the travel recommendation presented to the user.

5. Experiment and Evaluation

There are various views on social network data based recommender systems by considering the usage of various recommendation algorithms. In our experiment, there are six steps to generate the dataset, as indicated in Figure 2. Web crawler collects travel spatiotemporal data from social media, travel agent websites, and navigation apps. We select 10 cities (Chongqing, Chengdu, Beijing, Shanghai, Xian, Hangzhou, Nanjing, Tianjin, Guangzhou, and Wuhan) and scenic spots in these cities to analyze the sample travellers' touring history sequences, as indicated in Figure 3. We

further compare the scenic labels with those in the Tourist Attraction Knowledge Base (denoted as TAKB) using Natural Semantic Matching technology and manual filtering. In every city, 20 attractions are selected to form the city travel knowledge base. Finally, we split the travel sequence dataset as 70% of the data for training and 30% for testing the CTRR-SES algorithm. In the following experimental evaluation, we randomly select different users for testing.

To validate the CTRR-SES, the experiment was designed based on the collected touring data.

5.1. Accuracy and Validation of Travel Route Recommendation Algorithm

5.1.1. Impact of the Value of K on the Travel Preference Baseline Model. The value of K to perform the K -means clustering algorithm has a great impact on the experimental results. Thus, we run the fixed K value multiple times and use the updated SSE and the mean of SC to determine the optimal value. As indicated in Figure 4, when the K value is greater than 4, then the growth rate of SSE decreases. The increase of value K leads to the increase of the value of SC (o). Next, we set the degree of similarity as the recommendation accuracy rate. Feature vectors of the recommended route and that of the corresponding route in the testing dataset are computed using the similarity function when $K = 2, 3, 4, 5, 6$ (experimental results are shown in Figure 4). As the bars show, the recommendation accuracy rate is the highest when $K = 4$. Hence, in the following experiments, we set the value of $K = 4$ in this paper.

5.1.2. Length Comparison of Recommendation Sequence and Original Sequence. The sequence length of the original route in the testing dataset and that of the recommendation route

Input: travel sequences set $TS = \{L_1, L_2, \dots, L_n\}$ and the number of clusters k

Output: travel sequence cluster set $TC = \{TC_1, TC_2, \dots, TC_k\}$ and k center sequences set $newMedoids = \{L_1, L_2, \dots, L_K\}$

Initialization: $oldMedoids \leftarrow \text{null}$, $newMedoids \leftarrow \text{null}$;

- (1) Select k sequences L_1, L_2, \dots, L_k from TS randomly as initial center sequences to $oldMedoids$;
- (2) $TC_i \leftarrow L_i$ //Each center sequence corresponds to a cluster
- (3) **while** ($!isEqual(oldMedoids, newMedoids)$)
- (4) Calculate the similarity of each sample sequence from TS to each center sequence from $newMedoids$ and place the sample sequence in the cluster with the highest similarity to the center sequence;
- (5) $oldMedoids \leftarrow newMedoids$;
- (6) Recalculate the center sequence of each cluster TC_i , sequences with the highest similarity from each sample sequence in the cluster, as $newMedoids$;
- (7) **return** TC and $newMedoids$;

ALGORITHM 4: Travel sequence clustering algorithm (TSCA).

Input: Historical Travel sequences set $HS = \{L_{11}, L_{12}, \dots, L_{1n}\}$ of user A, city a, historical travel sequences $HSA = \{L_{21}, L_{22}, \dots, L_{2m}\}$ of city a, historical travel sequences $HSAU$ of all users

Output: Travel recommendation sequences of city a for user A

- (1) $newMedoids = \text{TSCA}(HSAU)$ //Cluster historical travel sequences of all users
- (2) **for** $i \leftarrow 1$ to $newMedoids.size$ **do**
- (3) **for** $j \leftarrow 1$ to n **do**
- (4) $userpre[i] \leftarrow userpre[i] + \text{TSCA}(newMedoids[i], L_{1j})$;
- (5) $userpre[i] = userpre[i]/n$;
- (6) **end for**
- (7) **end for**
- (8) **for** $t \leftarrow 1$ to m **do**
- (9) **for** $r \leftarrow 1$ to $newMedoids.size$ **do**
- (10) $cityseq[t][r] = \text{TSSA}(newMedoids[r], L_{2t})$;
- (11) **end for**
- (12) **end for**
- (13) $sim \leftarrow 0$;
- (14) **for** $t \leftarrow 1$ to m **do**
- (15) **if** $sim < \text{CosSim}(userpre, cityseq[t])$ **then**
- (16) $sim \leftarrow \text{CosSim}(userpre, cityseq[t])$;
- (17) $outputseq \leftarrow L_{2t}$;
- (18) **end if**
- (19) **end for**
- (20) **return** $outputseq$;

ALGORITHM 5: Travel sequence recommendation algorithm (TSRA).

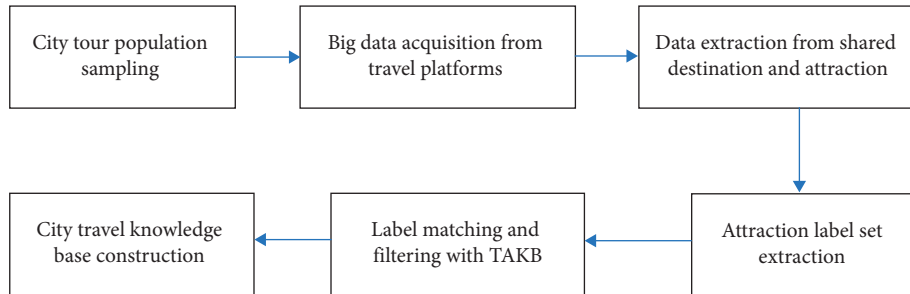


FIGURE 2: City travel knowledge data generating process.

are counted and compared, as shown in Figure 5. Compared with the sequence length of the real route, the experimental result of a small error proves that our algorithm is validated in its accuracy.

5.1.3. *Hit Rate.* The formula of hit rate is given:

$$\text{Hit rate} = \frac{2 * (|P_r \cap P_o| / P_r * (|P_r \cap P_o| / P_o))}{(|P_r \cap P_o| / P_r) + (|P_r \cap P_o| / P_o)}. \quad (6)$$

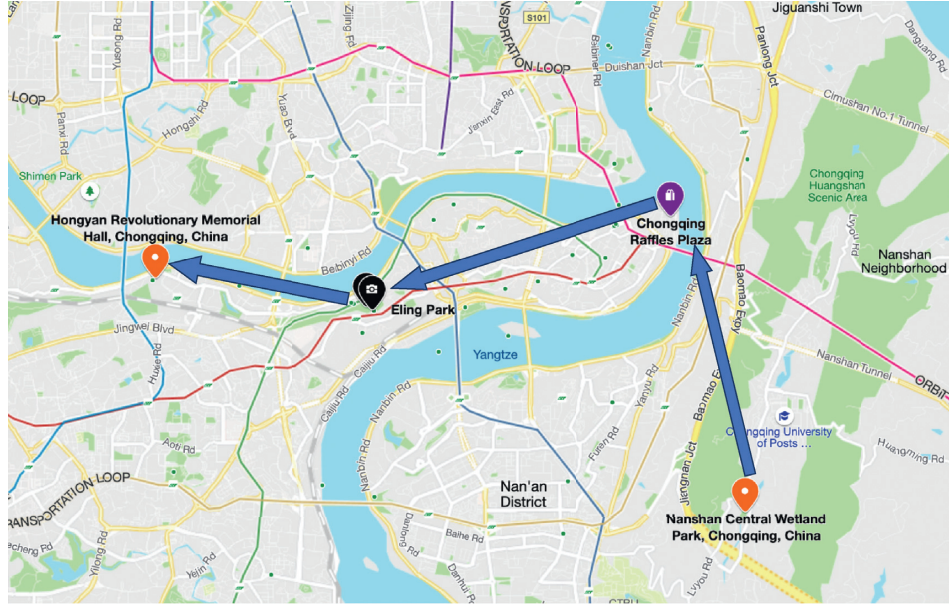


FIGURE 3: Example of city travel event sequence.

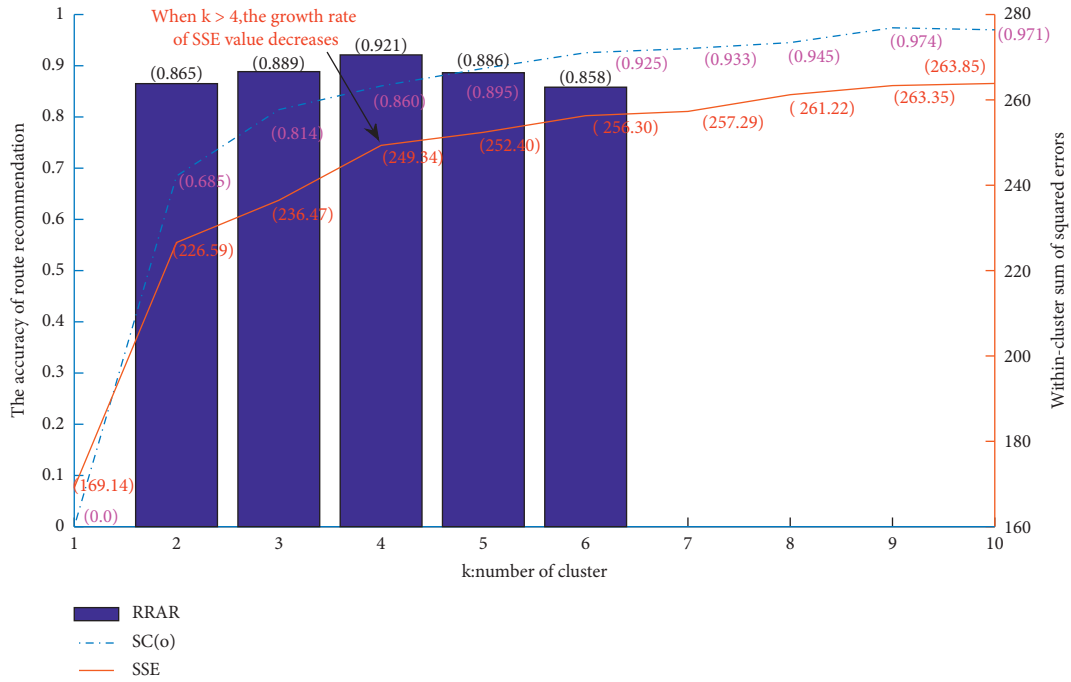


FIGURE 4: Evaluation index of clustering quality and accuracy with different K values.

In equation (6), P_r is the set of attractions in the recommended route, and P_o is the set of attractions in the user's travel historical sequence. The higher Hit Rate indicates better performance of recommendation by our algorithm. Then we calculate the accuracy of the route recommendation. The experimental hit rate result is 0.70, which further validates the CTRR-SES, proving that this algorithm will provide city travel route recommendation that effectively matches the user's preference.

5.2. Robustness of Travel Route Recommendation Algorithm. To test the robustness of the CTRR-SES, we design the following experiments, as shown in Table 1. Randomly change one or multiple sequences in the user's historical city touring sequence, and the experimental results are much like the original results detailed in Figure 6. Thus our algorithm has good performance in its robustness and stability.

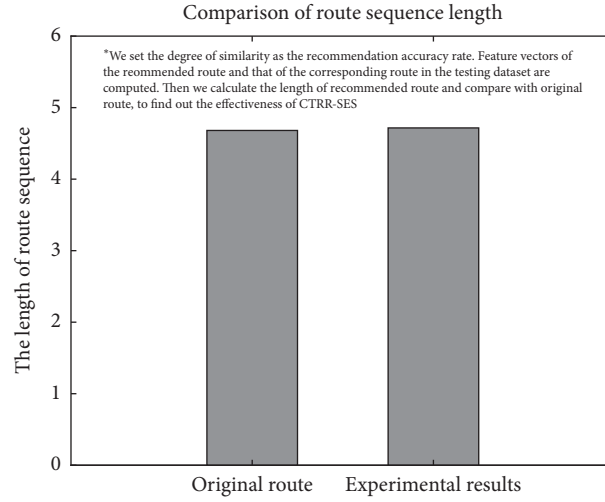


FIGURE 5: The length comparison of the original route with the recommendation route.

TABLE 1: Robustness tests.

Experiment no.	Result	Experimental design	Figure
1	The recommendation route accuracy rate is 92.1% after alteration and 99.8% of similarity degree with the original	Randomly change one item of one sequence in the user's historical city touring sequences	Figure 6
2	The recommendation route accuracy rate is 91.8% after alteration and 99.5% of similarity degree with the original recommendation route	Randomly change 50% items in one sequence in the user's historical city touring sequences	Figure 6
3	The recommendation route accuracy rate is 91.7% after alteration and 99.1% of similarity degree with the original recommendation route	Randomly change one item in each sequence of 50% sequences in the user's historical city touring sequences	Figure 6
4	The recommendation route accuracy rate is 91.5% after alteration and 98.8% of similarity degree with the original recommendation route	Randomly change 50% items in each sequence of 50% sequences in the user's historical city touring sequences	Figure 6

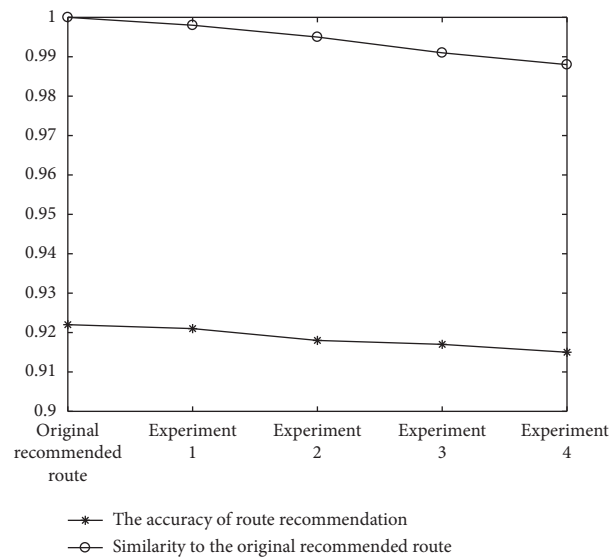


FIGURE 6: Comparative experiment of the original route with partial sequence alteration.

6. Conclusion

Existing travel recommendation studies seldom analyze user behavior with different granularities to calculate spatiotemporal sequence similarity. As a lack of full understanding of behavior events from multigranularity and multiperspective, those studies are not suitable for the growing need for in-depth city travel route recommendations. We adopt the coevolving spreading dynamics to the relevance of the traveling preferences and the events in the city tour networks and explore its application on the city tour recommendation system. Based on defining the user's touring sequence model, firstly, this paper presents the Event Sequence Similarity Measurement Method, which calculates the weighted mean of time, space, and activity similarity in certain granularity to measure spatiotemporal sequence similarity. Next, we design the CTRR-SES by applying the User Travel Preference Baseline Learning Model to study user's city travel historical data and compute personalized travel preferences. Finally, our algorithm is validated by a series of experiments of its effectiveness and feasibility, and CTRR-SES shows better performance in predicting the user's new city travel sequence fitting the user's individual preference. Our work provides reference and guidance to research the multigranularity spatiotemporal sequence similarity problem for city travel route recommendation. However, only 54 real cases are selected to evaluate the performance of the CTRR-SES algorithm, and we will include more experiments and datasets to validate the work in future research.

Data Availability

The travel historical data used to support the findings of this study have been deposited in the Mafengwo.com repository.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant no. 61672115 and Project No. 2020CDCGJSJ040 supported by the Fundamental Research Funds for the Central Universities.

References

- [1] Y. Zhu, Y. G. Zou, and L. Chen, "Analysis of spatial characteristics of urban tourism flow based on UGC data-taking Shanghai as an example," *Tourism Forum*, vol. 12, no. 4, pp. 33–41, 2019.
- [2] L. Li and Y. Jing, *Android city tour guide system based on web service*, 2012.
- [3] Y. Guo and H. Liu, "The embedding convergence of smart cities and tourism internet of things in China: an advance perspective," *Advances in Hospitality and Tourism Research*, vol. 2, no. 1, pp. 54–69, 2014.
- [4] M. D. Aswale, "Survey on recommendation of personalized travel sequence," *Ijarccce*, vol. 6, no. 1, pp. 108–113, 2017.
- [5] J. Huang, Y. Liu, Y. Chen, and C. Jia, "Dynamic recommendation of POI sequence responding to historical trajectory," *ISPRS International Journal of Geo-Information*, vol. 8, no. 10, p. 433, 2019.
- [6] S. Jiang, X. Qian, T. Mei, and Y. Fu, "Personalized travel sequence recommendation on multi-source big social media," *IEEE Transactions on Big Data*, vol. 2, no. 1, pp. 43–56, 2016.
- [7] W. Wörndl, A. Hefe, and D. Herzog, "Recommending a sequence of interesting places for tourist trips," *Information Technology & Tourism*, vol. 17, no. 1, pp. 31–54, 2017.
- [8] L. Zheng, F. Z. Wei, C. T. Lu et al., "Gated spectral units: modelling co-evolving patterns for sequential recommendation," *SIGIR*, vol. 2019, pp. 1077–1080, 2019.
- [9] W. Wang, Q. H. Liu, J. H. Liang et al., "Co-evolution spreading in complex networks," *Physics Report*, vol. 820, pp. 1–52, 2019.
- [10] K. H. Lim, "Recommending tours and places-of-interest based on user interests from geo-tagged photos," 2015.
- [11] T. Kurashima and T. Iwata, "Travel route recommendation using geotags in photo sharing sites," *CIKM*, vol. 10, pp. 579–588, 2010.
- [12] H. Yin and C. Wang, "Trip mining and recommendation from Geo-tagged photos," 2012.
- [13] M. Thenmozhi and S. Harshitha, "A framework for tourist recommendation system exploiting geo-tagged photos," 2016.
- [14] W. P. Sun, L. Chang, C. Z. Bin et al., "Travel route recommendation based on knowledge graph and frequent sequence mining," *Computer Science*, vol. 46, no. 2, pp. 56–61, 2019.
- [15] Y. Li, J. H. Xie, and L. C. Yang, "The preference analysis for tourist choice of scenic spot: a stated preference approach," *Tourism Tribune*, vol. 33, no. 12, pp. 88–98, 2018.