

## Research Article

# A Cold Start Context-Aware Recommender System for Tour Planning Using Artificial Neural Network and Case Based Reasoning

Zahra Bahramian,<sup>1</sup> Rahim Ali Abbaspour,<sup>1</sup> and Christophe Claramunt<sup>2</sup>

<sup>1</sup>*School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, North Kargar Ave., After Jalal Al Ahmad Crossing, Tehran 11155-4563, Iran*

<sup>2</sup>*Naval Academy Research Institute, Brest, France*

Correspondence should be addressed to Rahim Ali Abbaspour; [abaspour@ut.ac.ir](mailto:abaspour@ut.ac.ir)

Received 28 April 2017; Revised 19 June 2017; Accepted 28 June 2017; Published 10 September 2017

Academic Editor: Salvatore Carta

Copyright © 2017 Zahra Bahramian 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.

Nowadays, large amounts of tourism information and services are available over the Web. This makes it difficult for the user to search for some specific information such as selecting a tour in a given city as an ordered set of points of interest. Moreover, the user rarely knows all his needs upfront and his preferences may change during a recommendation process. The user may also have a limited number of initial ratings and most often the recommender system is likely to face the well-known cold start problem. The objective of the research presented in this paper is to introduce a hybrid interactive context-aware tourism recommender system that takes into account user's feedbacks and additional contextual information. It offers personalized tours to the user based on his preferences thanks to the combination of a case based reasoning framework and an artificial neural network. The proposed method has been tried in the city of Tehran in Iran. The results show that the proposed method outperforms current artificial neural network methods and combinations of case based reasoning with  $k$ -nearest neighbor methods in terms of user effort, accuracy, and user satisfaction.

## 1. Introduction

Nowadays, huge amounts of tourism information are available over the Web. Such information may be helpful for a given user who plans to visit an unknown region. However, it is often overwhelming and time consuming for the user to look through all the available information in order to find out some relevant places of interest and to organize them in a sort of well-defined tour. Hence, irrelevant information should be filtered and personalized options should be extracted according to the user's preferences.

Recommender systems (RSs) can be roughly defined as information filtering and decision support tools that provide products and services that match user's preferences [1–3]. Such recommender systems limit the information overload, provide personalized content, and increase systems usability as adapted to the problem [4–8]. Most of current

recommender systems assume that user's preferences do not change whatever the context [9, 10]. However, in the case of a recommender system applied to the tourism domain, user's preferences can be affected by many contextual factors that should be particularly considered when providing recommendations. For instance, when considering the user interest for a given point of interest (POI), many additional contextual parameters such as user's own mobility, previous history, and the timing of a recommendation should be considered as important contextual information. Hence, in order to provide satisfactory services to the final users, a context-aware tourism recommender system should be used in order to automatically adapt the recommendations to changing contexts [11–13]. In other words, context-aware tourism recommender systems should incorporate contextual information into the recommendation process to appropriately model and predict the user preferences. Moreover, they should adapt

these recommendations not only to the user's preferences but also to the contextual situation to generate more relevant suggestions.

However, and to the best of our knowledge, previous context-aware tourism recommender systems developed so far have not completely addressed these issues [9, 11–28]. One weakness of these researches is that they often suggest to a given user a list of POIs, which are specific tourist attractions that a user may find interesting such as a historical or cultural buildings or any places worth being visited, rather than a tour as an ordered set of POIs. Moreover, and despite the fact that collaborative recommendations are often provided based on other users, self-learning processes based on the context and assessment of a given user are not taken into account. Also, another common drawback of related context-aware tourism recommender systems is that they generally offer suggestions to a given user in one single session only. Indeed, most of the users rarely know all their needs at first. Therefore, initial recommendations are likely to be not always satisfactory. Moreover, most current context-aware tourism recommender systems often suffer from the cold start problem which occurs when a new user with little prior ratings has been registered to the system, making it difficult to suggest meaningful recommendations for that new user. They also assume that the user's preferences are stable and do not take into account user's preferences variations over time.

In order to overcome these problems, several researchers apply case based reasoning (CBR) [29–31] to develop case based recommender systems that present user's preference as a query item that he likes and each item case as a set of attributes [32–34]. These systems retrieve items that are similar to the user's query case. The main advantages of CBR is that (a) it can be directly applied to learn the user profile at the user modeling level, (b) it has the ability to learn the dynamic behavior of the system over time, and (c) it can provide effective explanation mechanisms when a sufficient number of previous cases are available.

However, an important issue still left for CBR is how to predict tours' ratings in order to recommend the most top rated tours to a user. In other words, some features are more important than the others, and then appropriate feature weighting should be executed before any rating prediction in order to take into account each feature relative importance. In existing case based recommender systems, the  $k$ -nearest neighbor (KNN) method [35] is widely used in rating prediction [32, 33], but it assumes that all selected features are equally important, this being not true in many cases as, for instance, each POI type is likely to have a specific interest value. Another open problem is how to progressively adapt the user model based on the information derived from the difference identified between a selected tour and the other tours in order to suggest a tour in a next cycle.

These problems lead us to proposing a hybrid method that combines an artificial neural network (ANN) [36, 37] with CBR. Our research develops a cold start context-aware recommender system applied to the tourism domain that offers personalized tours to the user based on his preferences and some contextual information. Our objective is to overcome the above-mentioned problems by considering user's

feedback and applying CBR using an ANN. The contributions of this paper include (1) providing a tourism recommender system that considers contextual situations in the recommendation process; (2) proposing an interactive framework that considers user's feedback to (a) suggest tours to a user with limited knowledge about his own needs, (b) overcome the cold start problem which occurs when a new user with little prior ratings uses the system, and (c) take into account user's preferences variations; (3) recommending some tours to the users including both POIs and routes between them over the street network; and (4) proposing tours to the user taking into account only his own preferences and feedbacks without considering the preferences and feedbacks of other users.

The rest of this paper is as follows. Section 2 introduces several concepts from recommender to context-aware recommender systems. It also briefly reviews context-aware recommender systems applied to the tourism domain. Then it presents the principles behind CBR and ANN and a brief introduction to semantic similarity principles. The proposed method and its implementation results are explained in Sections 3 and 4, respectively. Finally, Section 5 provides conclusions and outlines future works.

## 2. Literature Review

This section describes several basic concepts on recommender and context-aware recommender systems with a specific focus on the tourism domain. The main principles behind CBR, ANN, and property-based semantic similarity are briefly discussed.

*2.1. Recommender Systems.* Recommender systems (RSs) select some items proven to better match user's preferences and reduce information overload [4, 5, 38]. Without loss of generality, let us categorize current recommendation approaches towards content-based recommender systems, collaborative recommender systems, knowledge-based recommender systems, and hybrid recommender systems. They are discussed below.

*Content-based recommender systems* derive the similarity of features with respect to user's preferences and select the ones that best match user's preferences [39]. *Collaborative recommender systems* make recommendations based on the preferences of similar users [3, 40]. *Knowledge-based recommender systems* apply reasoning mechanisms to make appropriate suggestions according to user's preferences as related to items properties [41–43]. An interactive recommender system can be also considered as a knowledge-based recommender system. Navigation-by-proposing is one form of interactive recommender systems that are based on preference-based feedback [44–46]. *Hybrid recommender systems* apply a combination of current recommendation techniques. The main idea behind hybrid recommender systems is to exploit the advantages of one technique to compensate for the shortcomings of another and improve the overall performance [47]. A metalevel hybrid recommender system is a class of hybrid recommender system that uses the model learned by one recommender system as an input for another [47].

*2.2. Context-Aware Recommender Systems.* Context may be defined as any additional information that can be used to characterize the location and properties of a given entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves [48]. In context-aware recommender systems (CARSSs), the ratings are as a function of not only items and users, but also of contextual attributes whose role can be roughly formalized as follows [10, 49]:

$$R: \text{user} \times \text{item} \times \text{context} \longrightarrow \text{rating}, \quad (1)$$

where user, item, and context denote the considered modeling dimensions while rating returns the value of the considered item for that user according to the context.

*2.3. Context-Aware Recommender Systems in the Tourism Domain.* Several context-aware recommender systems have been developed so far in the tourism application domain. One of the early mobile examples in the tourism domain is the Cyberguide project [14]. The Cyberguide offers several tour guide prototypes for handheld devices where contextual information on the user surroundings is provided on request. Contextual knowledge of the user's current and past locations are taken into account to generate user daily histories. GeoWhiz [22] proposes a mobile restaurant recommender system that uses a collaborative filtering approach based on user locations. Magitti [16] offers a mobile leisure guide system where a user context is detected. It infers current and possible future leisure activities of interest and recommends suitable venues to the users (e.g., stores, restaurants, parks, and cinemas). It offers a sort of context-aware system that considers time and location and additional context data such as weather and venues opening hours, as well as previous user patterns. It is also activity-aware and filters items not matching the users or explicitly specified activities. Reference [9] develops a location-based service recommendation model (LBSRM) that integrates location and user preferences to recommend the most appropriate hotels to the user. User preferences are taken into account thanks to an adaptive method including long-term and short-term preference adjustments.

Reference [15] proposes a hybrid mobile context-aware system to personalize POIs recommendations. POIs located within a radius around the user's location and based on the user's trajectory and speed are suggested. Contextually filtered POIs are filtered and selected according to the user's preferences. The mobile tourism recommendation system (MTRS) developed by [20] delivers several personalized recommendation services to mobile users, considering contextual information such as the user's location, current time, weather conditions, and POIs already visited by the users. The iTravel system [27] applies a collaborative filtering approach. The similarities between users are evaluated using the user rating lists. Nearby users are detected and rating data are exchanged using mobile peer-to-peer communications. I'm feeling Loco [13] is a ubiquitous location-based sites recommendation. It uses the user's preferences derived from the Foursquare location-based social network, the user's location,

the used transport modes (e.g., walking, bicycle, or car), and the current user feeling in recommendation process. ReRex [11] assesses and models the relationship between contextual factors and item ratings. Using a predictive model based on matrix factorization, ReRex recommends POI and suggests a complete itinerary according to a particular itinerary's context.

SigTur/E-Destination [25] provides personalized recommendations applied to tourism activities. It considers different kinds of data including demographic information and travel motivations and takes into account user actions and ratings to generate recommendations. The approach is based on a tourism domain ontology that favors the classification of user activities. The reasoning process also combines content-based and collaborative recommendations. TRIPBUILDER [17] suggests personalized city sightseeing tours. POIs are derived from Wikipedia and illustrated by Flickr georeferenced photos. Tours are inferred from POI travel times as inferred from Google maps and a Traveling Salesman Problem algorithm that minimizes travel times and maximizes user's interests. POIs are categorized and semantically described by their popularity and average visiting time and further enriched by patterns of tourist movements.

In [26], tourist locations of interest are derived from their proximity to Flickr geotagged photos, semantically contextualized according to their related season and weather context, and finally rated in order to recommend some POIs. Travel histories are then derived and annotated in a sort of topic-based model, this favoring the search for user similarities. Reference [21] develops a probabilistic topic model to recommend POIs, based on user's interests as well as the ones of socially related users.

POST-VIA 360 [19] assists tourists in previsit, during-visit, and postvisit stages of their travels by means of a customer relationship management (CRM) approach. Geolocated POIs are classified and recommended thanks to an artificial immune system (AIS) that successively rates the POIs according to user rating. PlanTour [18] generates personalized oriented multiple-day tourist plans. The POIs are identified from a Minutube traveling social network and clustered over a daily basis. Tourist plans maximize the user utility and POIs travel distance. Reference [28] mines multiple periods in the time sequence to calculate the user's period of arriving at a certain area. Then they use item-based collaborative filtering for recommending locations based on users' preference and the periodic behaviors.

Table 1 summarizes such context-aware recommender systems as applied to the tourism domain according to a series of preselected criteria. The previous context-aware recommender systems in tourism domain apply single-shot recommendation strategy and return a single set of suggestions to a given user in one session only. They do not consider the user feedback in an interactive recommender system in tourism domain. Moreover, they are unable to propose recommendations to a user that has limited knowledge about his needs, or little prior ratings, or changing preferences during the time. In addition, none of the previous context-aware RSs in tourism domain recommend tours as sequences of POIs to the user.

TABLE 1: Context-aware recommender systems applied to the tourism domain.

Recommender system	Context-aware properties										Recommender System properties					User feedback			
	User	Environment	Location	Time	Contextual prefiltering	Contextual recommendation postfiltering	Contextual modeling	POI	Hotel	Restaurant	Leisure	Route	Tour	Content-based	Collaborative	Knowledge-based	Hybrid	Yes	No
Cyberguide [14]	✓		✓	✓		✓		✓						✓				✓	
GeoWhiz [22]	✓	✓	✓	✓	✓				✓						✓			✓	✓
Magitti [16]	✓	✓	✓	✓	✓		✓			✓							✓	✓	✓
LBSRM [9]	✓	✓	✓	✓	✓			✓						✓				✓	✓
MTRS [20]	✓	✓	✓	✓	✓		✓							✓				✓	✓
REJA [15]	✓	✓	✓	✓	✓				✓					✓				✓	✓
I'm feeling Loco [13]	✓		✓	✓	✓			✓						✓				✓	✓
ReRex [11]	✓	✓	✓	✓			✓	✓						✓				✓	✓
iTravel [27]	✓	✓	✓	✓		✓		✓						✓	✓			✓	✓
SigTur/E-Destination [25]	✓	✓	✓	✓		✓		✓									✓	✓	✓
TRIPBUILDER [17]	✓	✓	✓	✓			✓					✓						✓	✓
[26]	✓	✓	✓	✓		✓		✓							✓			✓	✓
SocoTraveler [21]	✓	✓	✓	✓			✓	✓							✓			✓	✓
POST-VIA 360 [19]	✓	✓	✓	✓			✓	✓							✓			✓	✓
PlanTour [18]	✓	✓	✓	✓	✓			✓							✓			✓	✓
[28]	✓	✓	✓	✓		✓		✓							✓			✓	✓

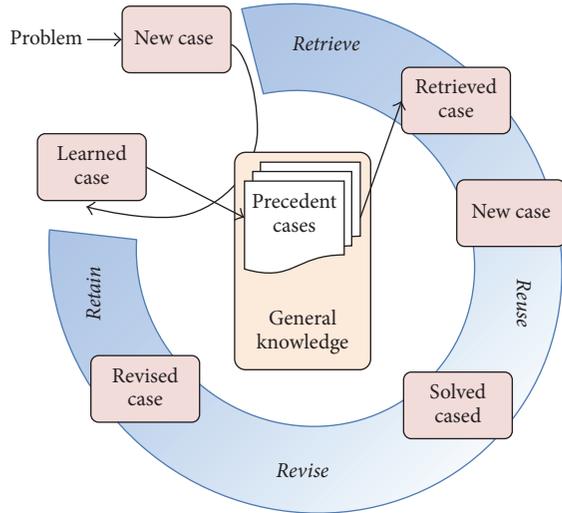


FIGURE 1: Case based reasoning cycle.

**2.4. Case Based Reasoning.** Case Based Reasoning (CBR) is a machine learning method that models human reasoning processes for learning and problem solving. A CBR stores past cases as a historical repository. Each case consists of two parts including the problem description and the solutions [29, 50, 51]. The problem description represents the context, that is, when the case takes place, while the solution part shows the methods applied to resolve the case. The objective of CBR is to solve new problems using solutions that were used to address similar problems [30, 31]. The structural CBR approach is one type of case representation that represents cases according to a common structured vocabulary using predefined attributes and values [52]. CBR generally involves four steps [53]: (1) retrieving similar previously experienced cases, (2) reusing cases by copying or integrating the emerging solutions from the retrieved cases, (3) revising or adapting the retrieved solutions to solve a new problem, and (4) retaining a new confirmed or validated solution (Figure 1).

**2.5. Artificial Neural Network.** Artificial Neural Network (ANN) [36, 37] is a branch of artificial intelligence inspired by the biological brain. It is a structure of connected nodes (i.e., artificial neurons) that are arranged in layers. The links between nodes have weights associated with them depending on the amount of influence one node has on another. A feedforward ANN is an ANN without any cycles in the network and propagates incoming data in a forward direction only. A multilayer perceptron (MLP) is one type of feedforward ANNs with full connection of each layer to the next one. The MLP consists of one input layer, one or more hidden layers, and one output layer. Nodes in the input layer respond to data that is fed into the network, while output nodes produce network output values. Hidden nodes receive the weighted output from the previous layer's nodes. Figure 2 shows the  $n_{\text{input}} - n_{\text{hidden}} - n_{\text{output}}$  ( $n_{\text{input}}$  denotes input neurons,  $n_{\text{hidden}}$  denotes hidden neurons, and  $n_{\text{output}}$  denotes output neurons) architecture of a feedforward multilayer perceptron ANN.

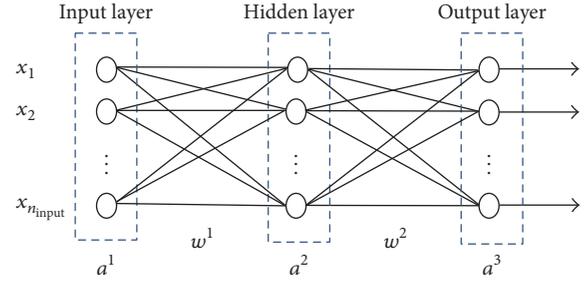


FIGURE 2: Structure of a feedforward neural network.

The output of unit  $i$  in layer  $l + 1$  is

$$a^{l+1}(i) = f^l \left( \sum_{j=1}^{n_l} w^l(i, j) a^l(j) + b^l(i) \right). \quad (2)$$

The most common concrete algorithm for learning MLP is the backpropagation algorithm. In the backpropagation algorithm, the computed output values are compared with the correct output values to compute the value of error function. Then, the error is fed back through the network. The derivatives of the error function with respect to the network weights are calculated. The weights of each connection are adjusted using gradient descent method such that the value of the error function decreases. This process is repeated until the network converges to a state with small error.

Different training algorithms have been so far applied to train a backpropagation MLP. Each algorithm adjusts the ANN as follows:

$$w_k = w_{k-1} + \Delta w_k, \quad (3)$$

where  $k$  is the index of iterations,  $w_k$  is the vector of weights in  $k$  iteration of training algorithm, and  $\Delta w_k$  is the vector of weights changes that is computed by each training algorithm including gradient descent with momentum and adaptive learning rate backpropagation (GD), resilient backpropagation (RP), conjugate gradient backpropagation with Fletcher-Reeves updates (CGF), conjugate gradient backpropagation with Polak-Ribiere updates (CGP), conjugate gradient backpropagation with Powell-Beale restarts (CGB), scaled conjugate gradient backpropagation (SCG), BFGS quasi-Newton backpropagation (BFG), one-step secant backpropagation (OSS), and Levenberg-Marquardt backpropagation (LM) (Table 2).

**2.6. Property-Based Semantic Similarity.** The semantic similarity of two objects can be evaluated based on their common features or the lack of distinctive features of each object according to a domain ontology. A feature-based semantic similarity measure is one of the main approaches that evaluate the semantic similarity among concepts [54]. It considers the features of the concepts considered according to the ontology and evaluates the common and distinct features to derive a similarity measure.

Consider objects  $O_1$  and  $O_2$  with  $n_p$  properties ( $p, q = 1, 2, \dots, n_p$ ). Using Tversky's formulation, the similarity

TABLE 2: Algorithms for training backpropagation multilayer perceptron neural network.

Training algorithm	Description
GDX	<p>It renovates weight and bias values conceding to the gradient descent and current trends in the error surface (<math>\Delta E_k</math>) with adaptive training rate.</p> $\Delta w_k = \rho \Delta w_{k-1} + \alpha p \frac{\Delta E_k}{\Delta w_k}.$
RP	<p>It considers only the sign of the partial derivative to determine the direction of the weight update and multiplies it with the step size (<math>\Delta_k</math>).</p> $\Delta w_k = -\text{sign}\left(\frac{\Delta E_k}{\Delta w_k}\right) \Delta_k.$
CGF	<p>The vector of weights changes is computed as</p> $\Delta w_k = \alpha_k p_k,$ <p>where <math>p_k</math> is the search direction that is computed as</p> $p_k = \begin{cases} -g_0, & k = 0, \\ -g_k + \beta_k p_{k-1}, & k \neq 0, \end{cases}$ <p>where the parameter <math>\beta_k</math> is computed as</p> $\beta_k = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}}.$
CGP	<p>The vector of weights changes is computed as</p> $\Delta w_k = \alpha_k p_k,$ <p>where <math>p_k</math> is the search direction that is computed as</p> $p_k = \begin{cases} -g_0, & k = 0, \\ -g_k + \beta_k p_{k-1}, & k \neq 0, \end{cases}$ <p>where the parameter <math>\beta_k</math> is computed as</p> $\beta_k = \frac{g_{k-1}^T g_k}{g_{k-1}^T g_{k-1}}.$
CGB	<p>The search direction is reset to the negative of the gradient only if there is very little orthogonality between the present gradient and the past gradient as explained in this condition.</p> $ g_{k-1}^T g_k  \geq 0.2 \ g_k\ ^2.$
SCG	<p>It denotes the quadratic approximation to the error <math>E</math> in a neighborhood of a point <math>w</math> by</p> $E_{qw}(y) = E(w) + E'(w)^T y + \frac{1}{2} y^T E''(w) y.$ <p>To minimize <math>E_{qw}(y)</math>, the critical points for <math>E_{qw}(y)</math> are found as the solution to the following linear system:</p> $E'_{qw}(y) = E''(w) y + E'(w) = 0.$
BFG	<p>The vector of weights changes is computed as</p> $\Delta w_k = -H_k^T g_k,$ <p>where <math>H_k</math> is the Hessian (second derivatives) matrix approximated by <math>B_k</math> as</p> $B_k = B_{k-1} + \frac{y_{k-1} y_{k-1}^T}{y_{k-1}^T \Delta x_{k-1}} - \frac{B_{k-1} \Delta x_{k-1} (B_{k-1} \Delta x_{k-1})^T}{\Delta x_{k-1}^T B_{k-1} \Delta x_{k-1}}.$
OSS	<p>It generates a sequence of matrices <math>G^{(k)}</math> that represents increasingly accurate approximations to the inverse Hessian (<math>H^{-1}</math>). The updated expression uses only the first derivative information of <math>E</math> as follows:</p> $G^{(k+1)} = G^{(k)} + \frac{pp^T}{p^T v} - \frac{(G^{(k)} v) v^T G^{(k)}}{v^T G^{(k)} v} + (v^T G^{(k)} v) uu^T,$ <p>where</p> $p = w^{(k+1)} - w^{(k)},$ $v = g^{(k+1)} - g^{(k)},$ $u = \frac{p}{p^T v} - \frac{G^{(k)} v}{v^T G^{(k)} v},$ <p>and <math>g</math> is the transfer function. It does not store the complete Hessian matrix and assumes that the previous Hessian was the identity matrix at each iteration.</p>
LM	<p>The vector of weights changes is computed as</p> $\Delta w_k = -\left(J_k^T J_k + \mu I\right)^{-1} J_k e_k,$ <p>where combination coefficient <math>\mu</math> is always positive, <math>I</math> is the identity matrix, <math>J_k</math> is Jacobian matrix (first derivatives), and <math>e_k</math> is network error.</p>

between two objects is given by a feature-based semantic similarity calculation method as follows [54, 55]:

$$\text{SIM}(O_1, O_2) = \sum_{q=1}^{n_p} w_q \text{SIM}_{p_q}, \quad (4)$$

$$\text{SIM}_{p_q} = \frac{\alpha \text{CF}_{p_q}(O_1, O_2)}{\beta \text{CF}_{p_q}(O_1) + \gamma \text{CF}_{p_q}(O_2) + \alpha \text{CF}_{p_q}(O_1, O_2)},$$

where  $\text{CF}_{p_q}(O_1)$ ,  $\text{CF}_{p_q}(O_2)$ , and  $\text{CF}_{p_q}(O_1, O_2)$  denote the common features of  $O_1$  and  $O_2$ , distinctive features of  $O_1$ , and distinctive features of  $O_2$  with respect to property  $p_q$ , respectively.  $w_q$  assigns the relevance amount of the property  $p_q$  and  $\alpha, \beta, \gamma \in \mathbb{R}$  are constants that value the respective importance of the features considered.

### 3. Proposed Methodology

The objective of this paper is to consider the user feedbacks and develop a hybrid interactive context-aware recommender system applied to the tourism domain that offers personalized tours to a user based on his preferences. The proposed method combines an interactive tourism recommender system designed as a knowledge-based recommender system with a content-based tourism recommender system thanks to applying CBR and ANN.

The proposed method takes into account contextual information in the modeling process as a contextual modeling approach whose objective is to recommend the most appropriate tours to a user. It takes into account contextual information as follows:

- (i) POIs locations in the user's environment
- (ii) POIs opening and closing times
- (iii) the street network
- (iv) the tours already visited by the user denoted as his mobility history
- (v) the distance traveled by the user on the street network.

Moreover, the rating assigned to the visited tours by the user and user's feedbacks are considered as user preferences and feedbacks, respectively. The following subsections develop the problem formulation and the proposed context-aware tourism recommender system.

**3.1. Problem Formulation.** In this paper, each tour is modeled as a sequence of POIs. A POI is a specific tourist attraction or place of interest that a user may find of value.  $\text{Pois}$  denotes the set of all POIs located in the case area as

$$\text{Pois} = \{\text{poi}_1, \text{poi}_2, \dots, \text{poi}_{n_{\text{poi}}}\}, \quad (5)$$

where  $n_{\text{poi}}$  is the total number of POIs that are located in the case area.

The proposed method considers  $n_{\text{type}}$  POI types such as restaurant and cultural place types. Each POI belongs to one

of these types. Let  $\text{Types}$  denote the set of all POI types available in the case area as

$$\text{Types} = \{\text{tp}_1, \text{tp}_2, \dots, \text{tp}_{n_{\text{type}}}\}. \quad (6)$$

The function  $\text{FuncType}$  returns the type of a given POI as

$$\text{FuncType: Poi} \rightarrow \text{Type}. \quad (7)$$

A  $\text{Tour}_i$  is a nonempty nonrepetitive sequence of POIs that a user visits:

$$\text{Tour}_i = \{\text{tr}_{i1}, \text{tr}_{i2}, \dots, \text{tr}_{in}\}, \quad (8)$$

where  $n$  is the number of POIs in  $\text{Tour}_i$  and  $\text{tr}_{ij}$  is a POI ( $\text{tr}_{ij} \in \text{Pois}$ ,  $j = 1, 2, \dots, n$ ) in such a way that  $\forall l \neq m: \text{tr}_{il} \neq \text{tr}_{im}$ .

$\text{tm}_{ij}$  ( $j = 1, 2, \dots, n$ ) denotes the time that a user starts visiting  $\text{tr}_{ij}$ . Without loss of generality, let us introduce a constraint to illustrate the potential of the approach. The proposed method also assumes that the dining time is between 12 am and 2 pm. Therefore, if the start time of visiting a POI in a tour is between 12 am and 2 pm, the mentioned POI must be located in "Restaurant" type.

$$\text{if } 12 \text{ am} \leq \text{tm}_{ij} \leq 2 \text{ pm}, \quad (9)$$

$$\text{then } \text{FuncType}(\text{tr}_{ij}) = \text{"Restaurant"}.$$

Then  $\text{Tours}$  denotes the set of all possible tours considering the mentioned constraints that is given as

$$\text{Tours} = \left\{ \bigcup_{\text{possible tours}} \text{Tour}_i \right\}. \quad (10)$$

The user can assign a rating to each visited tour which is then stored in his history. The rating values are in the unit interval  $[0, 1]$  and reflect the user's need. The tour with the highest rating will be the most interesting one. The  $\text{FuncRank}$  function returns  $\text{Rating}_i$  as the rating of  $\text{Tour}_i$ .

$$\text{FuncRate: Tours} \rightarrow [0, 1] \quad (11)$$

$$\text{Rating}_i = \text{FuncRate}(\text{Tour}_i).$$

For each  $\text{Tour}_i = \{\text{tr}_{i1}, \text{tr}_{i2}, \dots, \text{tr}_{in}\}$ ,  $P_i$  is computed as

$$P_i = \left( p_{i1}, p_{i2}, \dots, p_{in_{\text{type}}}, \text{TourDistance}_i \right), \quad (12)$$

where  $p_{ij}$  ( $j = 1, 2, \dots, n_{\text{type}}$ ) denotes the number of POIs in  $\text{Tour}_i$  that are located in  $\text{tp}_j$  POI type;  $\text{TourDistance}_i$  denotes the distance traveled in  $\text{Tour}_i$  that is the sum of distances between each two consecutive POIs in  $\text{Tour}_i$  on the street network and computed as

$$\text{TourDistance}_i = \sum_{j=1}^{n-1} \text{Dist}(\text{tr}_{ij}, \text{tr}_{i(j+1)}), \quad (13)$$

where  $\text{Dist}(\text{tr}_{ij}, \text{tr}_{i(j+1)})$  is the shortest path between  $\text{tr}_{ij}$  and  $\text{tr}_{i(j+1)}$  on the street network.

*3.2. Proposed Context-Aware Tourism Recommender System.* Our objective is to develop a context-aware tourism recommender system that suggests the highest rated tours to a user based on his preferences. We assume that the user does not know a priori the types of POIs he would like to visit, and that he has little prior knowledge of possible recommended tours. The system asks the user to assign the starting time and the ending time of his requested tour. The proposed method first considers the user's mobility history and his interactions with the system to capture user's preferences. The proposed interactive tour recommender system is developed in three successive steps.

*Step 1.* The system learns the current user model and recommends  $n_{\text{show}}$  new tours to the user based on the current user model and some contextual information.

*Step 2.* The user selects (or rejects) a tour among the recommended tours and assigns a rating to it as a feedback. The feedback on the selected tour elicits the user's needs. This feedback is case based rather than feature-based.

*Step 3.* The system revises the user model for the next cycle using information about the selected tour and  $n_{\text{similar}}$  similar tours.

The recommendation process finishes either after the presentation of a suitable tour to the user or after some predefined iterations.

In order to implement this interactive tour recommender system, the system applies the CBR method as a knowledge-based filter. It develops a structural case representation that each case consists of a tour and its related rating. Each tour is considered as a problem description while its rating is the problem solution. CBR stores a history of the visited tours and their related ratings as previous cases. When applying the retrieval and reusing stages of a CBR cycle, the system uses the previously visited tours and recommends the  $n_{\text{show}}$  highest rated tours to the user. At the revising stage of a CBR cycle, the user selects or rejects one of the recommended tours and revises its rating by assigning a new rating to it as a feedback. In the retaining stage of CBR cycle, the system adapts the user model to use it in the next cycle.

To overcome the mentioned problems in an interactive case based recommender system, the proposed approach combines an ANN with CBR and actually takes into account the specific properties and values of each tour. We introduce a hybrid recommender system that applies ANN (as a content-based recommender) and CBR (as a knowledge-based recommender) in a homogeneous approach denoted as a metalevel hybrid recommender system. A feedforward multilayer perceptron ANN is used in the retrieval, reusing, and retaining phases of a CBR cycle as follows.

*3.2.1. Learning Current User Model and Recommending New Tours.* In this step, the proposed ANN is applied to the retrieval and reusing phases of a CBR cycle. The proposed multilayer perceptron neural network is made of 3 layers: an input layer, a hidden layer, and an output layer. For

each visited  $\text{Tour}_i$ , the inputs of the proposed ANN are the elements of  $P_i$ . Therefore, the input layer has  $(n_{\text{type}} + 1)$  neurons. The proposed ANN output is the related rating of the tour ( $\text{Rating}_i$ ) and therefore the output layer has one neuron. The system uses the history of the visited tours and their related ratings to train the ANN.

The ANN computes the ratings of all possible tours (Tours) and recommends the  $n_{\text{show}}$  best computed rated tours to the user, where  $n_{\text{show}}$  is a system specified constant.

*3.2.2. Assigning the User Feedback.* The user selects (or rejects) a tour from  $n_{\text{show}}$  recommended tours and assigns its rating as a feedback. This feedback is used to revise the user model.

*3.2.3. Revising the User Model according to His Feedback.* In this step, the proposed ANN is still used in the retaining phase of a CBR cycle due to its strong adaptive learning ability. The ANN adapts the user model based on the user's feedback. The similarity of the selected tours with all other tours is computed. Consider two tours  $\text{Tour}_i = \{\text{tr}_{i1}, \text{tr}_{i2}, \dots, \text{tr}_{in}\}$  and  $\text{Tour}_j = \{\text{tr}_{j1}, \text{tr}_{j2}, \dots, \text{tr}_{jn}\}$ . The system computes  $P_i = (p_{i1}, p_{i2}, \dots, p_{in_{\text{type}}}, \text{TourDistance}_i)$  and  $P_j = (p_{j1}, p_{j2}, \dots, p_{jn_{\text{type}}}, \text{TourDistance}_j)$ . In order to compute the similarity between  $\text{Tour}_i$  and  $\text{Tour}_j$ ,  $\text{SIM}(\text{Tour}_i, \text{Tour}_j)$ , the similarity between  $P_i$  and  $P_j$  is computed using Tversky's feature-based semantic similarity method described in Section 2.6.

$$\begin{aligned} \text{SIM}(\text{Tour}_i, \text{Tour}_j) &= \sum_{q=1}^{n_{\text{type}}+1} w_q \text{SIM}_{P_q} \\ \text{SIM}_{P_q} &= \frac{\alpha \text{CF}_{P_q}(\text{Tour}_i, \text{Tour}_j)}{\beta \text{CF}_{P_q}(\text{Tour}_i) + \gamma \text{CF}_{P_q}(\text{Tour}_j) + \alpha \text{CF}_{P_q}(\text{Tour}_i, \text{Tour}_j)} \\ &\sim \begin{cases} \min(p_{iq}, p_{jq}), & q = 1, 2, \dots, n_{\text{type}}, \\ \min(\text{TourDistance}_i, \text{TourDistance}_j), & q = n_{\text{type}} + 1. \end{cases} \end{aligned} \quad (14)$$

In this paper, we assume  $\alpha = \beta = \gamma = 1$ .

Therefore, the system determines the  $n_{\text{similar}}$  most similar tours to the selected tour, where  $n_{\text{similar}}$  is a system specified constant. It assigns their ratings the same as the selected tour rating. Using new ratings of the selected tour and its similar tours, the system adapts the proposed ANN and updates the weights of the proposed ANN. Therefore, the user model is updated for the next cycle. Accordingly, the system stores the user's feedbacks as new experiences in the memory.

This process is iterated until either a suitable tour is presented to the user or a predefined number of iterations is reached, where the number of iterations is a system specified constant. In fact, too few interaction stages might lead to inaccurate user modeling and recommendations, while on the other hand interacting in too many iteration stages is likely to be a cumbersome user task. In order to provide a good balance, we consider 20 iteration stages. This choice is relatively similar to the ones made by related works and that considered between 10 and 20 iteration stages [56–59].

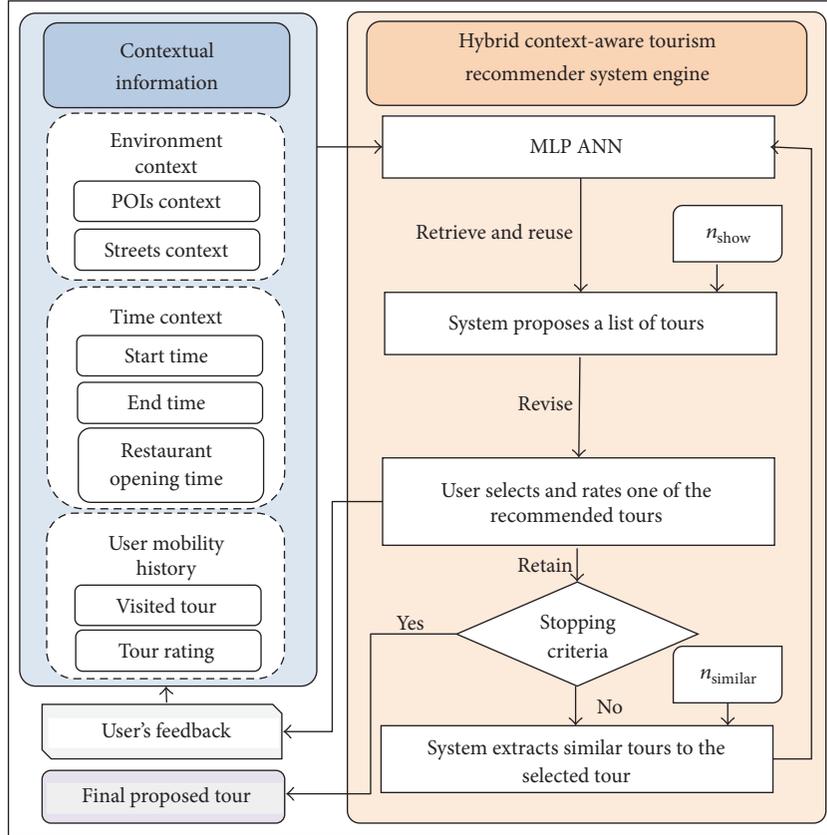


FIGURE 3: Architecture of the proposed hybrid context-aware recommender system applied to the tourism domain.

The proposed hybrid CBR-ANN approach is used as an interactional tour recommender system by combining content-based recommender system (using ANN) and knowledge-based recommender system (using CBR) (Figure 3). As shown in Figure 3, a supporting database represents the location and type of each POI, streets information, and restaurants' opening times. The system specifies the number of tours that the system recommends to the user at each cycle ( $n_{show}$ ), the number of the most similar tours to the selected tour that is used in the retaining phase ( $n_{similar}$ ), and the number of iterations. The user defines the starting and ending times of the required tour to the system and assigns a rating to the selected tour. Algorithm 1 shows the pseudocode of the proposed method.

#### 4. Experimental Results and Discussion

The proposed hybrid context-aware tourism recommender system is evaluated according to the experimental context of a user that wants to plan a tour. The algorithm has been implemented by an Android-based prototype.

**4.1. Data Set.** The proposed hybrid context-aware tourism recommender system suggests tours to each user who wants to plan a tour in regions 11 and 12 of Tehran, the capital city of Iran. In order to develop the experimental setup, the role of the users has been taken by 20 students in the Surveying and

Geospatial Engineering School of the University of Tehran. In the two regions considered, there are 55 POIs. Each POI belongs to one of the eleven POI types identified ( $n_{type} = 11$ ) including hotel, cultural place, museum, shopping center, mosque, park, theater, cinema, cafe, restaurant, and stadium (Figure 4).

Without loss of generality, a series of assumptions have been considered as follows:

- (i) Each tour is a sequence of two to five POIs. It is similar to the one made by several previous works in tourism recommender system [25, 60, 61] that recommend two to five POIs per tour. However, the proposed method can be extended and applied to situations with bigger number of POIs per tour with some minor adaptations.
- (ii) The time that a user starts visiting a POI is either 8 am, 10 am, 12 am, 2 pm, or 4 pm ( $tm_{ij} \in \{8 \text{ am}, 10 \text{ am}, 12 \text{ am}, 2 \text{ pm}, 4 \text{ pm}\}$ ).
- (iii) The user starts to visit a POI ( $tr_{i(j+1)}$ ) 2 hours after he starts to visit a previous POI ( $tr_{ij}$ ):

$$tm_{i(j+1)} = tm_{ij} + 2 \text{ hours.} \quad (15)$$

Initially, each user visits 6 tours in the case area and assigns a rating to each of them by the unit interval. The system saves these ratings in the user's history. To evaluate the

```

(1) Input: CB_initial is a set of visited tours and their ratings
(2) Output: CB, net, recommendation
(3) CB = CB_initial
(4) net = Train_ANN(CB) //net is a trained ANN using CB
(5) while ~termination criteria
(6)     R = Reuse(CB, net, n_show) //System recommends the first n_show highest
        computed rated tours
(7)     S = User_Select(R) //User selects and rates one of the recommended tours
(8)     E = Extract(S, n_similar) //System extracts the first n_similar prior tours which
        have maximum similarities with tour S
(9)     net, CB = Retain(net, CB, S, E) //System adapts CB and ANN
(10) end

```

ALGORITHM 1: Pseudocode of the proposed hybrid context-aware tourism recommender system.

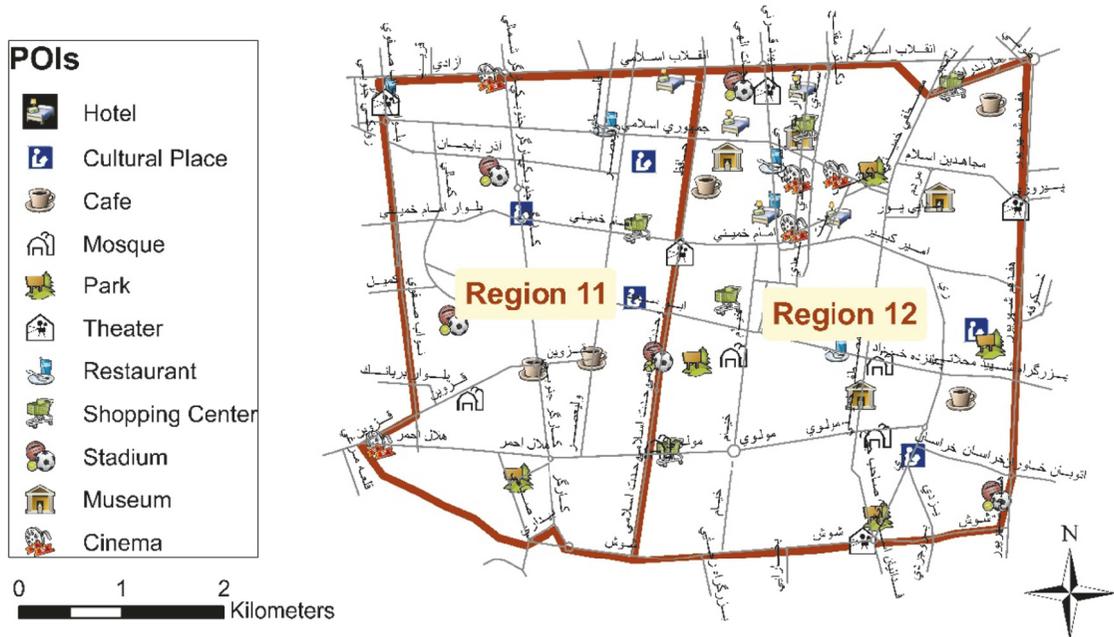


FIGURE 4: Synthetic view of the case study.

proposed context-aware tourism recommender system, some evaluation metrics and methods should be determined.

**4.2. Evaluation Metrics and Methods.** In order to evaluate the proposed method, the proposed system (CBR-ANN based recommender system) is compared to two other recommender systems including ANN based recommender system and CBR-KNN based recommender system:

- (i) The ANN based recommender system is a single-shot content-based recommender system that uses an ANN to learn user profile.
- (ii) The CBR-KNN based recommender system is an interactive recommender system that uses CBR method as a knowledge-based filter combined with KNN in the retrieving, reusing, and retaining phases of CBR.

On the other hand, three evaluation metrics are considered to compare CBR-ANN with ANN and CBR-KNN methods including user effort, accuracy\_error, and user\_satisfaction

- (i) user effort: The system needs enough user feedbacks to generate valid recommendations, while without gathering enough user feedbacks this may lead to a poor user model and inaccurate recommendations. The number of user interactions with the system before reaching the final recommendation determines the user effort metric.
- (ii) accuracy\_error: The accuracy\_error denotes the degree to which the recommender system matches the user's preferences. It is evaluated by the difference between the rating given by the system to the final selected tour denoted as  $\hat{s}_{\text{final}}$  and the one of the user denoted as  $s_{\text{final}}$ . The lower the error of accuracy is,

the better the method is performing; it is given as follows:

$$\text{accuracy\_error} = |\hat{s}_{\text{final}} - s_{\text{final}}|. \quad (16)$$

- (iii) *user\_satisfaction*: The *user\_satisfaction* is evaluated by determining to which degree the final recommended tour is considered as valid by the user or not. The rating assigned by the user to the final recommended tour ( $s_{\text{final}}$ ) is used as an indicator of *user\_satisfaction*. It is valued by an interval  $[0, 1]$ , that is, 0 as not satisfied to 1 as satisfied.

$$\text{user\_satisfaction} = s_{\text{final}}. \quad (17)$$

Moreover, to evaluate the proposed method, two different types of evaluation methods are applied including per user and overall evaluation methods:

- (i) The per user evaluation method only takes into account the recommendation process of a specific user and computes evaluation metrics for a specific user.
- (ii) The overall evaluation method computes average, minimum, and maximum quality over all users. Let  $UE_i$ ,  $AE_i$ , and  $US_i$  denote the user effort, accuracy\_error, and *user\_satisfaction* of a specific user, respectively. Let  $n_{\text{user}}$  denote the number of users considered (i.e., 20 in this paper). The mean, minimum, and maximum values of the user effort over all users are computed as follows:

$$\begin{aligned} \text{MeanUE} &= \frac{1}{n_{\text{user}}} \sum_{i=1}^{n_{\text{user}}} UE_i, \\ \text{MinUE} &= \min \left( \bigcup_{i=1}^{n_{\text{user}}} UE_i \right), \\ \text{MaxUE} &= \max \left( \bigcup_{i=1}^{n_{\text{user}}} UE_i \right). \end{aligned} \quad (18)$$

The mean, minimum, and maximum values of the *accuracy\_error* over all users are computed as follows:

$$\begin{aligned} \text{MeanAE} &= \frac{1}{n_{\text{user}}} \sum_{i=1}^{n_{\text{user}}} AE_i, \\ \text{MinAE} &= \min \left( \bigcup_{i=1}^{n_{\text{user}}} AE_i \right), \\ \text{MaxAE} &= \max \left( \bigcup_{i=1}^{n_{\text{user}}} AE_i \right). \end{aligned} \quad (19)$$

The mean, minimum, and maximum values of the *user\_satisfaction* over all users are computed as follows:

$$\begin{aligned} \text{MeanUS} &= \frac{1}{n_{\text{user}}} \sum_{i=1}^{n_{\text{user}}} US_i, \\ \text{MinUS} &= \min \left( \bigcup_{i=1}^{n_{\text{user}}} US_i \right), \\ \text{MaxUS} &= \max \left( \bigcup_{i=1}^{n_{\text{user}}} US_i \right). \end{aligned} \quad (20)$$

After determining the evaluation metrics, an overall evaluation of the figures that emerge from a given user taken as an example is described in the next subsections.

*4.3. Per User Evaluation of the Results.* For each user, the system learns the current user model and recommends new tours accordingly (Section 4.3.1). Then the user selects one of the recommended tours and assigns a rating to it. The system revises the user model accordingly (Section 4.3.2). After 20 iterations, the final recommendation is presented to the user.

*4.3.1. Learning Current User Model and Recommending New Tours.* For each user, during the retrieval and reusing stage of a CBR cycle, the proposed ANN takes the history of a given user as an input to determine the relationship between the visited tours and their related ratings. This gives the rationale behind the learning process. The proposed feedforward multilayer perceptron neural network consists of three layers including one input layer, one hidden layer, and one output layer. Herein, eleven POI types are considered. Since the inputs of the proposed neural network are the number of POIs in the tour that are located in each POI type and the distance traveled in the tour, the input layer has twelve neurons. The output layer has one neuron that evaluates the rating of the considered tour.

The optimum number of hidden neurons depends on the problem and is related to the complexity of the input and output mapping, the amount of noise in the data, and the amount of training data available. Too few neurons lead to underfitting, while too many neurons contribute to overfitting. For each user, to obtain the optimal number of neurons in the hidden layer, different ANNs with different hidden neurons number between one and twelve are trained.

In order to train each of the ANNs, the data set is divided into three subsets including training, validation, and test sets. We do consider four tours and their related ratings as training data set, one tour and its related rating as validation data set, and one tour and its related rating as test data set. Then Mean Squared Error (MSE) is computed for training, validation, and test sets. MSE is given by the average squared difference between tour ratings computed by the ANN and assigned by the user. The training set is used to adjust the ANN's weights and biases while the validation set is used to prevent the overfitting problem. At execution, it appears that the MSE on the validation set decreases during the initial phase of

TABLE 3: Mean Square Error of different neural networks structures for a given user.

Number of neurons in hidden layer	MSE
1	0.018
2	0.022
3	0.067
4	0.029
5	<b>0.010</b>
6	0.018
7	0.014
8	0.011
9	0.044
10	0.032
11	0.045
12	0.038

the training. However, when the ANN begins to overfit the training data, the validation MSE begins to rise. The training stops at this point and the weights and biases related to the minimum validation MSE are returned. The MSE on the test data is not used during the ANN training, but it is used to compare the ANNs during and after training.

For ANNs with hidden neurons of a number between one and twelve, the network is trained and the Mean Square Error (MSE) value is computed. Table 3 shows MSE values related to ANNs with different number of hidden neurons for user 1. According to this experiment, it appears that the best multilayer perceptron neural network should have five hidden neurons as reflected by the MSE values for this user.

For each user, once the optimal number of hidden neurons is determined, the ANN is trained using several training algorithms that are used for training the backpropagation ANN. Table 4 shows Mean Square Error of different training algorithms used to train the ANN related to user 1. The results show that the Levenberg-Marquardt (LM) training method is the best training method for user 1 because it has the least MSE value. The trained ANN determines the relationship between the visited tours and their respective recommendation ratings and thus models the user’s preferences.

Similar approaches are commonly used to determine the best number of hidden neurons and the best training algorithm for the other users. Table 5 shows the best number of hidden neurons, and the best training algorithm of ANN to learn the current user model. The results show that GDX and LM are most often the best training functions.

*4.3.2. Revising the User Model Based on His Feedback.* For each user, after training the ANN, the new ratings of the tours are computed using the trained ANN. Therefore, according to the current user model, new  $n_{show}$  best rated tours are recommended to the user.

In the revision stage of CBR cycle, the user reviews the recommendations and selects and rates a tour as a feedback. In the retaining stage of CBR cycle, the system revises the user model based on the user’s feedback. The system computes the property-based semantic similarity between the selected

tour and all other tours, determines the  $n_{similar}$  most similar tours to the selected tour, and assigns their ratings according to the selected tour rating. These values are used to adapt the user model for the next cycle by updating the weights of the proposed ANN. Therefore, the system learns from the current user’s feedback and uses it to revise the user model. This process is iterated for 20 cycles in order to determine the final recommendation.

*4.3.3. Experimental Results for One User.* Figures 5 and 6 show the result of the proposed method for user 1 considering  $n_{show} = 4$  and  $n_{similar} = 2$ . One can remark that the final selected tour is different from the tours which the user has previously rated. Similar approaches are used for the other users.

Figure 7 shows the rating that user 1 assigns to the selected tour at each cycle. In the first cycle, the proposed ANN recommends four tours and the user selects one of them. The user’s rating of the first selected tour is 0.50. In the next cycles, the user selects one of the recommended tours and the system revises the recommendations according to the user feedbacks. The user’s rating of the selected tour reaches 0.80 after eleven cycles.

To determine the best recommendation, the parameters of the proposed method should be set.

*4.3.4. Setting the Proposed Method Parameters.* To evaluate the impact of the proposed method parameters, the proposed system is tried using different  $n_{show}$  and  $n_{similar}$  values. Table 6 shows the number of iteration steps to reach a stable state of final selected tour for different  $n_{show}$  values. The results show that increasing  $n_{show}$  reduces the number of required iterations to reach the final selected tour.

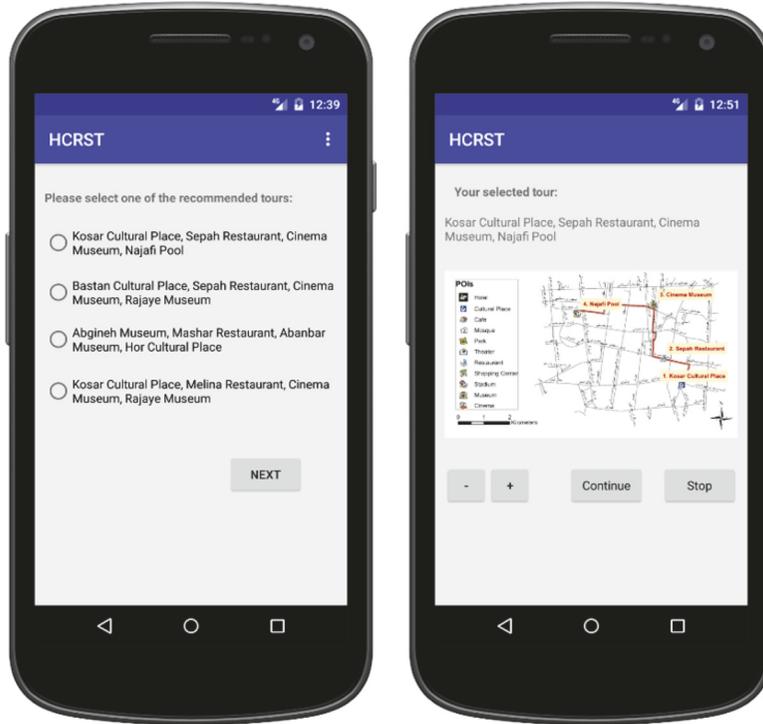
Table 7 shows the number of iteration steps to reach a stable state of the final selected tour for different  $n_{similar}$  values. The results show that increasing  $n_{similar}$  reduces the rating of the final selected tour.

*4.3.5. One User Result Evaluation.* Table 8 shows the evaluation results of the proposed method for user 1. The experiments show that the user’s rating of the selected tour reaches a good value of 0.80 after 11 cycles, 0.5 in one cycle, and 0.70 after 12 cycles in the CBR\_ANN, ANN, and CBR\_KNN methods, respectively. This indicates that CBR\_ANN needs relatively less user effort compared to CBR\_KNN. The experimental results show accuracy\_error values of the CBR\_ANN, ANN, and CBR\_KNN methods are 0.01, 0.06, and 0.04, respectively; this outlines a better performance of the CBR\_ANN method regarding the accuracy\_error metric. Applying the CBR-ANN method increases the user’s rating of the selected tour through the cycles by 60% and 14% relative to the ANN and CBR-ANN methods, respectively; this denotes an increasing user\_satisfaction.

*4.4. Overall Evaluation of the Results.* Table 9 shows the overall evaluation results of the proposed method. The results show that the CBR\_ANN method needs less user effort than the CBR-KNN method and decreases the mean,

TABLE 4: Mean Square Error of different training algorithms for a given user.

Training algorithm	GDX	RP	CGF	CGP	CGH	SCG	BFG	OSS	LM
MSE	0.033	0.033	0.033	0.023	0.013	0.013	0.012	0.012	0.010

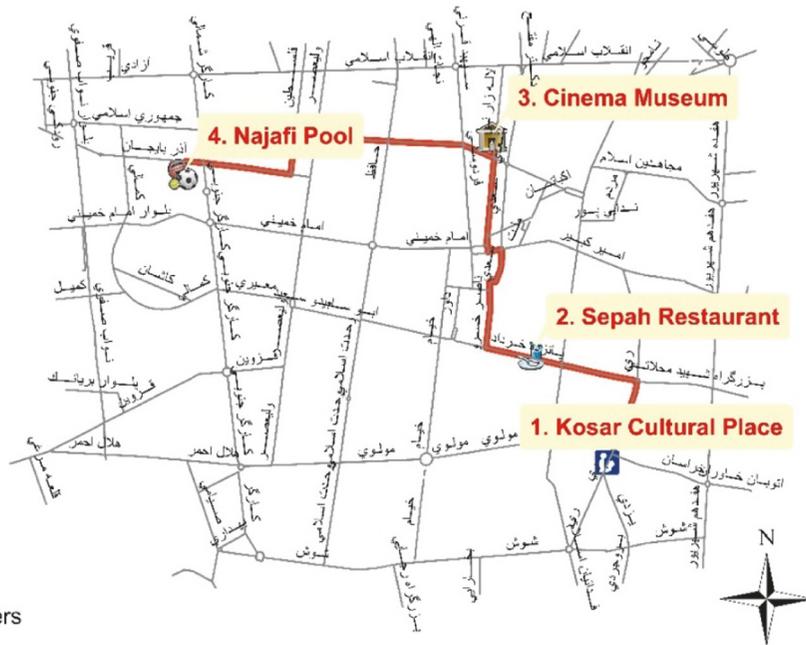


(a)

(b)

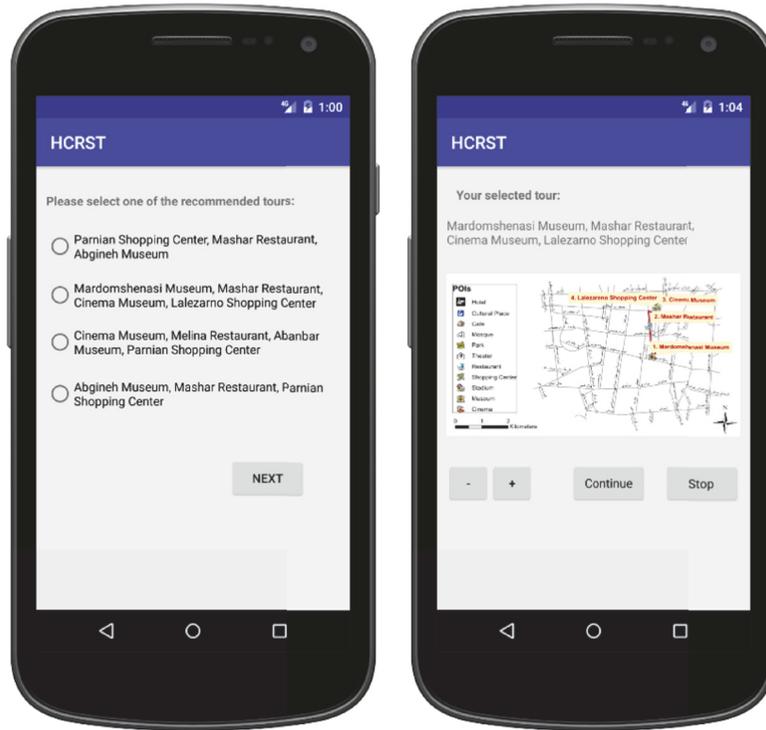
- POIs**
- Hotel
  - Cultural Place
  - Cafe
  - Mosque
  - Park
  - Theater
  - Restaurant
  - Shopping Center
  - Stadium
  - Museum
  - Cinema

0 1 2 Kilometers



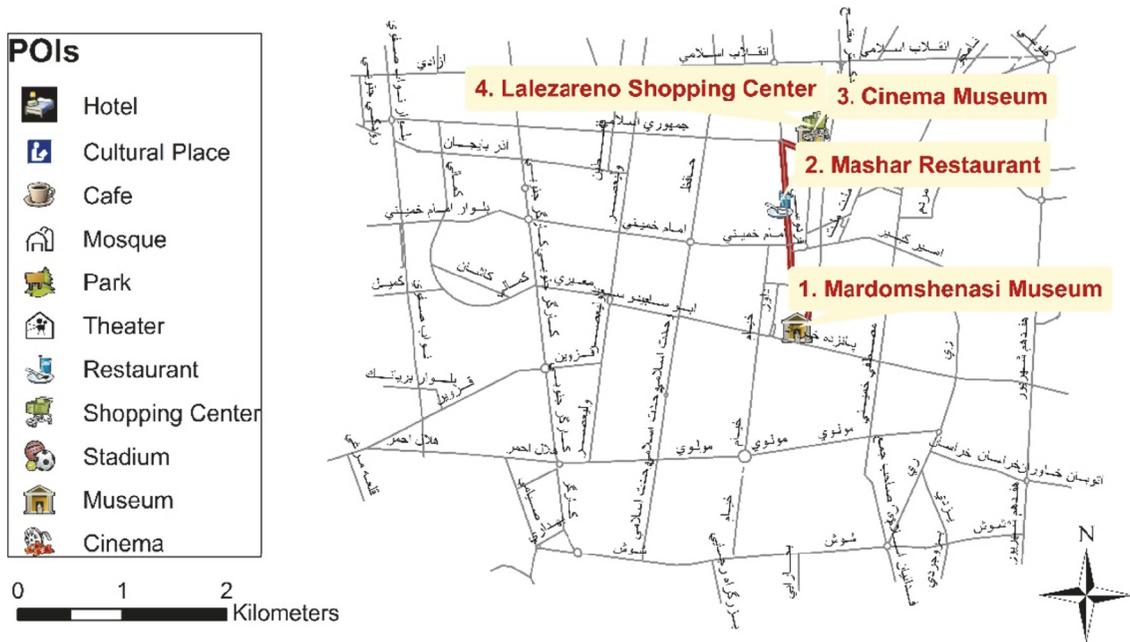
(c)

FIGURE 5: Proposed tour recommendation, (a) the first recommended tours, and (b, c) users' selected tour from the first recommended tours for a given user.



(a)

(b)



(c)

FIGURE 6: Proposed tour recommendation, (a) final recommended tours, and (b, c) users' selected tour from final recommended tours for a given user.

minimum, and maximum values of the users' efforts (i.e., 17%, 33%, and 13%, resp.). Moreover, the results show that the CBR-ANN method outperforms the ANN and CBR-ANN regarding the overall accuracy\_error and user\_satisfaction

metrics. Applying the CBR-ANN method increases the mean, minimum, and maximum values of users' satisfaction of the selected tour by 75%, 140%, and 23% compared to the ANN method, respectively. It also increases the mean,

TABLE 5: Best number of hidden neurons and training algorithm of ANN to learn the user model.

User number	Best number of hidden neurons	Best training function
1	5	LM
2	8	GDX
3	3	LM
4	5	CGB
5	2	CGB
6	4	BFG
7	5	GDX
8	9	RP
9	8	GDX
10	10	CGF
11	5	CGP
12	7	GDX
13	7	CGF
14	4	SCG
15	8	RP
16	3	LM
17	6	SCG
18	4	LM
19	11	OSS
20	10	RP

TABLE 6: Rating of the final selected tour and the number of iteration steps to reach the final selected tour considering different  $n_{show}$  values.

$n_{show}$	2	4	10
Final selected tour rating	0.80	0.80	0.80
Number of iterations to reach the final selected tour	19	11	5

TABLE 7: Rating of final selected tour and the number of iteration steps to reach the final selected tour considering different  $n_{similar}$  values.

$n_{similar}$	1	2	5
Final selected tour rating	0.80	0.80	0.75
Number of iterations to reach final selected tour	17	11	8

TABLE 8: Per user evaluation of the proposed method for a given user.

Method	UE	AE	US
CBR_ANN	11	0.01	0.80
ANN	1	0.06	0.50
CBR_KNN	12	0.04	0.70

minimum, and maximum values of users' satisfaction of the selected tour by 14%, 33%, and 14% compared to the ANN-KNN method, respectively. Moreover, the CBR-ANN method

TABLE 9: Overall evaluation of the proposed method.

Method	CBR_ANN	ANN	CBR_KNN
Overall UE			
MeanUE	10	1	12
MinUE	6	1	9
MaxUE	13	1	15
Overall AE			
MeanAE	0.02	0.10	0.08
MinAE	0.01	0.01	0.01
MaxAE	0.07	0.13	0.09
Overall US			
MeanUS	0.70	0.40	0.60
MinUS	0.60	0.25	0.45
MaxUS	0.80	0.65	0.70

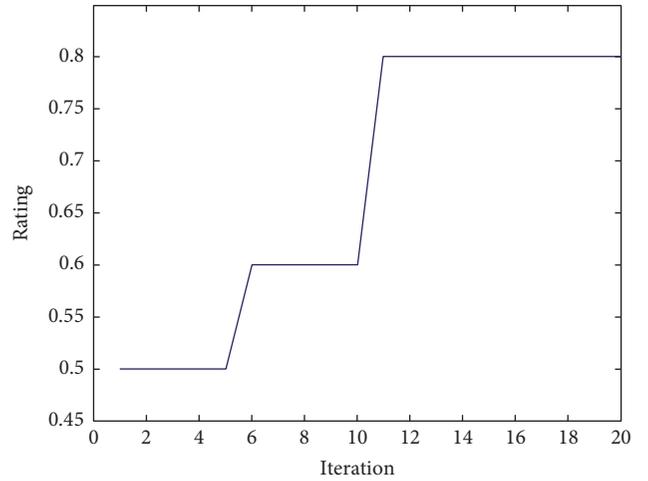


FIGURE 7: The rating that a given user assigns to the selected tour at each cycle.

has smaller mean and minimum values of accuracy\_error relative to the ANN and CBR\_KNN methods. This outlines a better performance of the CBR\_ANN method regarding the accuracy\_error metric.

4.5. Discussion. The proposed tourism recommender system has the following abilities:

- (i) It considers contextual situations in the recommendation process including POIs locations, POIs opening and closing times, the tours already visited by the user, and the distance traveled by the user. Moreover, it considers the rating assigned to the visited tours by the user and user's feedbacks as user preferences and feedbacks.
- (ii) It takes into account both POIs selection and routing between them simultaneously to recommend the most appropriate tours.
- (iii) It proposes tours to a user who has some specific interests. It takes into account only his own preferences

and does not need any information about preferences and feedbacks of the other users.

- (iv) It is not limited to recommending tours that have been previously rated by users. It computes the ratings of unseen tours that the user has not expressed any opinion about.
- (v) The proposed method takes into account the user's feedbacks in an interactive framework. It needs user's feedbacks to apply an interactive learning algorithm and recommend a better tour gradually. Therefore, it overcomes the mentioned cold start problem. The proposed method only needs a small number of ratings to produce sufficiently good recommendations. Moreover, it proposes some tours to a user with limited knowledge about his needs. Thanks to the available knowledge of the historical cases, less knowledge input from the user is required to come up with a solution. Also, it takes into account the changing of the user's preferences during the recommendation process.
- (vi) The proposed method uses an implicit strategy to model the user by asking him to assign a rating to the selected tour. Such process requires less user effort relative to when a user assigns a preference value to each tour selection criteria.
- (vii) The proposed method recalculates user predicted ratings after he has rated a single tour in a sequential manner. In comparison with approaches in which a user rates several tours before readjusting the model, the user sees the system output immediately. Also, it is possible to react to the user provided data and adjust them immediately.
- (viii) The proposed method combines CBR and ANN and exploits the advantages of each technique to compensate for their respective drawbacks. CBR and ANN can be directly applied to the user modeling as a regression or classification problem without more transformation to learn the user profile. Moreover, a noteworthy property of the CBR is that it provides a solution for new cases by taking into account past cases. Also, the trained ANN calculates the set of feature weights, those playing a core role in connecting both learning strategies. The proposed method has the advantages of being adaptable with respect to dynamic situations. As the ANN revises the user model, it has an online learning property and continuously updates the case base with new data.
- (ix) The results show that the proposed CBR-ANN method outperforms ANN based single-shot and CBR\_KNN based interactive recommender systems in terms of user effort, accuracy\_error, and user\_satisfaction metrics.

## 5. Conclusion and Future Works

The research developed in this paper introduces a hybrid interactive context-aware recommender system applied to

the tourism domain. A peculiarity of the approach is that it combines CBR (as a knowledge-based recommender system) with ANN (as a content-based recommender system) to overcome the mentioned cold start problem for a new user with little prior ratings. The proposed method can suggest a tour to a user with limited knowledge about his preferences and takes into account the user's preference changing during recommendation process. Moreover, it considers only the given user's preferences and feedbacks to select POIs and the route between them on the street network simultaneously.

Regarding further works, additional contextual information such as budget, weather, user mood, crowdedness, and transportation mode can be considered. Currently, the user has to explicitly rate the tours thus providing feedbacks which can be considered as a cumbersome task. One direction remaining to explore is a one where the system can obtain the user feedbacks by analyzing the user activity, such as saving or bookmarking recommended tour. Moreover, other techniques in content-based filtering (i.e., fuzzy logic) and knowledge-based filtering (i.e., critiquing) can be used. Also, collaborative filtering can be combined to the proposed method.

## Conflicts of Interest

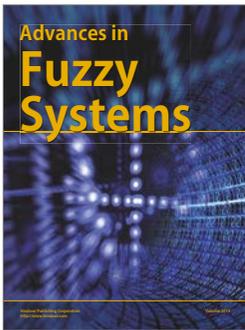
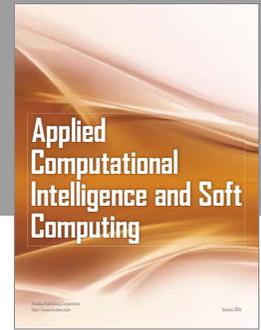
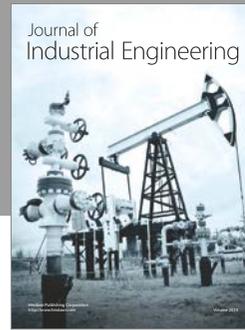
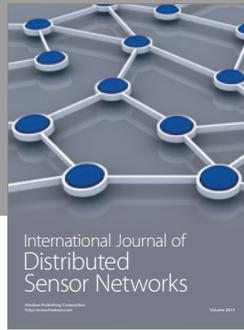
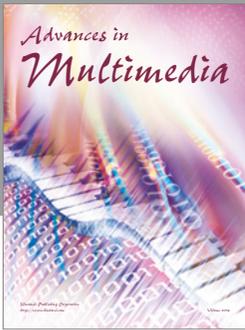
The authors declare that there are no conflicts of interest regarding the publication of this article.

## References

- [1] C. C. Aggarwal, *Recommender Systems: The Textbook*, Springer, 2016.
- [2] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, pp. 109–132, 2013.
- [3] F. Ricci, B. Shapira, and L. Rokach, *Recommender Systems Handbook*, 2nd edition, 2015.
- [4] J. Borràs, A. Moreno, and A. Valls, "Intelligent tourism recommender systems: a survey," *Expert Systems with Applications*, vol. 41, no. 16, pp. 7370–7389, 2014.
- [5] D. Gavalas, V. Kasapakis, C. Konstantopoulos, K. Mastakas, and G. Pantziou, "A survey on mobile tourism Recommender Systems," in *Proceedings of the 3rd International Conference on Communications and Information Technology, ICCIT 2013*, pp. 131–135, June 2013.
- [6] W.-J. Li, Q. Dong, and Y. Fu, "Investigating the temporal effect of user preferences with application in movie recommendation," *Mobile Information Systems*, vol. 2017, Article ID 8940709, 10 pages, 2017.
- [7] K. Zhu, X. He, B. Xiang, L. Zhang, and A. Pattavina, "How dangerous are your Smartphones? App usage recommendation with privacy preserving," *Mobile Information Systems*, vol. 2016, Article ID 6804379, 10 pages, 2016.
- [8] K. Zhu, Z. Liu, L. Zhang, and X. Gu, "A mobile application recommendation framework by exploiting personal preference with constraints," *Mobile Information Systems*, vol. 2017, Article ID 4542326, 9 pages, 2017.
- [9] M.-H. Kuo, L.-C. Chen, and C.-W. Liang, "Building and evaluating a location-based service recommendation system

- with a preference adjustment mechanism,” *Expert Systems with Applications*, vol. 36, no. 2, pp. 3543–3554, 2009.
- [10] G. Adomavicius and A. Tuzhilin, “Context-aware recommender systems,” in *Recommender Systems Handbook*, pp. 217–253, Springer, 2011.
- [11] L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci, “Context relevance assessment and exploitation in mobile recommender systems,” *Personal and Ubiquitous Computing*, vol. 16, no. 5, pp. 507–526, 2012.
- [12] H. Huang and G. Gartner, “Using context-aware collaborative filtering for POI recommendations in mobile guides,” in *Advances in Location-Based Services*, Lecture Notes in Geoinformation and Cartography, pp. 131–147, 2012.
- [13] N. S. Savage, M. Baranski, N. E. Chavez, and T. Höllerer, “I’m feeling LoCo: a location based context aware recommendation system,” in *Advances in Location-Based Services*, Lecture Notes in Geoinformation and Cartography, pp. 37–54, Springer, Berlin, Germany, 2012.
- [14] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton, “Cyberguide: a mobile context-aware tour guide,” *Wireless Networks*, vol. 3, no. 5, pp. 421–433, 1997.
- [15] M. J. Barranco, J. M. Noguera, J. Castro, and L. Martínez, “A context-aware mobile recommender system based on location and trajectory,” in *Management Intelligent Systems*, vol. 171 of *Advances in Intelligent Systems and Computing*, pp. 153–162, Springer, Berlin, Germany, 2012.
- [16] V. Bellotti, B. Begole, E. H. Chi et al., “Activity-based serendipitous recommendations with the magitti mobile leisure guide,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI ’08)*, pp. 1157–1166, ACM, Florence, Italy, April 2008.
- [17] I. R. Brilhante, J. A. Macedo, F. M. Nardini, R. Perego, and C. Renso, “On planning sightseeing tours with TripBuilder,” *Information Processing and Management*, vol. 51, no. 2, pp. 1–15, 2015.
- [18] I. Cenamor, T. de la Rosa, S. Núñez, and D. Borrajo, “Planning for tourism routes using social networks,” *Expert Systems with Applications*, vol. 69, pp. 1–9, 2017.
- [19] R. Colomo-Palacios, F. J. García-Peñalvo, V. Stantchev, and S. Misra, “Towards a social and context-aware mobile recommendation system for tourism,” *Pervasive and Mobile Computing*, vol. 38, pp. 505–515, 2017.
- [20] D. Gavalas and M. Kenteris, “A web-based pervasive recommendation system for mobile tourist guides,” *Personal and Ubiquitous Computing*, vol. 15, no. 7, pp. 759–770, 2011.
- [21] J. He, H. Liu, and H. Xiong, “SocoTraveler: Travel-package recommendations leveraging social influence of different relationship types,” *Information and Management*, vol. 53, no. 8, pp. 934–950, 2016.
- [22] T. Horozov, N. Narasimhan, and V. Vasudevan, “Using location for personalized POI recommendations in mobile environments,” in *Proceedings of the International Symposium on Applications and the Internet (SAINT ’06)*, pp. 124–129, IEEE, Phoenix, Ariz, USA, January 2006.
- [23] M. Kenteris, D. Gavalas, and A. Mpitziopoulos, “A mobile tourism recommender system,” in *Proceedings of the 15th IEEE Symposium on Computers and Communications (ISCC ’10)*, pp. 840–845, IEEE, Riccione, Italy, June 2010.
- [24] J. A. Mocholi, J. Jaen, K. Krynicki, A. Catala, A. Picón, and A. Cadenas, “Learning semantically-annotated routes for context-aware recommendations on map navigation systems,” *Applied Soft Computing Journal*, vol. 12, no. 9, pp. 3088–3098, 2012.
- [25] A. Moreno, A. Valls, D. Isern, L. Marin, and J. Borràs, “SigTur/E-Destination: ontology-based personalized recommendation of Tourism and Leisure Activities,” *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 633–651, 2013.
- [26] Z. Xu, L. Chen, and G. Chen, “Topic based context-aware travel recommendation method exploiting geotagged photos,” *Neurocomputing*, vol. 155, pp. 99–107, 2015.
- [27] W.-S. Yang and S.-Y. Hwang, “ITravel: a recommender system in mobile peer-to-peer environment,” *Journal of Systems and Software*, vol. 86, no. 1, pp. 12–20, 2013.
- [28] B. Xu, Z. Ding, and H. Chen, “Recommending locations based on users’ periodic behaviors,” *Mobile Information Systems*, vol. 2017, Article ID 7871502, 9 pages, 2017.
- [29] Y. Guo, J. Hu, and Y. Peng, “Research of new strategies for improving CBR system,” *Artificial Intelligence Review*, vol. 42, no. 1, pp. 1–20, 2014.
- [30] M. M. Richter, “The search for knowledge, contexts, and Case-Based Reasoning,” *Engineering Applications of Artificial Intelligence*, vol. 22, no. 1, pp. 3–9, 2009.
- [31] C. K. Riesbeck and R. C. Schank, *Inside Case-Based Reasoning*, Psychology Press, 2013.
- [32] L. Chen, G. Chen, and F. Wang, “Recommender systems based on user reviews: the state of the art,” *User Modeling and User-Adapted Interaction*, vol. 25, no. 2, pp. 99–154, 2015.
- [33] F. Ricci, D. Cavada, N. Mirzadeh, and A. Venturini, “Case-based travel recommendations,” *Destination Recommendation Systems: Behavioural Foundations and Applications*, pp. 67–93, 2006.
- [34] C.-S. Wang and H.-L. Yang, “A recommender mechanism based on case-based reasoning,” *Expert Systems with Applications*, vol. 39, no. 4, pp. 4335–4343, 2012.
- [35] D. T. Larose, *Discovering Knowledge in Data: An Introduction to Data Mining*, John Wiley & Sons, 2014.
- [36] I. N. da Silva, D. H. Spatti, R. A. Flauzino, L. H. B. Liboni, and S. F. dos Reis Alves, *Artificial Neural Networks: A Practical Course*, Springer, 2016.
- [37] J. Heaton, *Introduction to the Math of Neural Networks*, Heaton Research Inc, 2012.
- [38] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou, “Mobile recommender systems in tourism,” *Journal of Network and Computer Applications*, vol. 39, no. 1, pp. 319–333, 2014.
- [39] R. P. Biuk-Aghai, S. Fong, and Y.-W. Si, “Design of a recommender system for mobile tourism multimedia selection,” in *Proceedings of the 2nd International Conference on Internet Multimedia Services Architecture and Applications (IMSAA ’08)*, pp. 1–6, IEEE, Bangalore, India, December 2008.
- [40] F. Martinez-Pabon, J. C. Ospina-Quintero, G. Ramirez-Gonzalez, and M. Munoz-Organero, “Recommending ads from trustworthy relationships in pervasive environments,” *Mobile Information Systems*, vol. 2016, Article ID 8593173, 18 pages, 2016.
- [41] T. Berka and M. Plöbñig, “Designing recommender systems for tourism,” in *Proceedings of the 11th International Conference on Information Technology in Travel & Tourism (ENTER ’04)*, 2004.
- [42] A. Hinze and S. Junmanee, “Travel recommendations in a mobile tourist information system,” in *ISTA’ 2005, 4th International Conference*, Lecture Notes in Informatics, pp. 89–100, GI edition, 2005.
- [43] W. Husain and L. Y. Dih, “A framework of a Personalized Location-based traveler recommendation system in mobile application,” *International Journal of Multimedia and Ubiquitous Engineering*, vol. 7, no. 3, pp. 11–18, 2012.

- [44] L. McGinty and B. Smyth, "Comparison-based recommendation," in *ECCBR 2002: Advances in Case-Based Reasoning*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp. 575–589, 2002.
- [45] H. Shimazu, "ExpertClerk: Navigating shoppers' buying process with the combination of asking and proposing," in *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI '01)*, pp. 1443–1448, August 2001.
- [46] H. Shimazu, "ExpertClerk: A conversational case-based reasoning tool for developing salesclerk agents in e-commerce webshops," *Artificial Intelligence Review*, vol. 18, no. 3-4, pp. 223–244, 2002.
- [47] R. Burke, "Hybrid recommender systems: survey and experiments," *User Modelling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.
- [48] A. K. Dey, "Understanding and using context," *Personal and Ubiquitous Computing*, vol. 5, no. 1, pp. 4–7, 2001.
- [49] P. M. P. Rosa, J. J. P. C. Rodrigues, and F. Basso, "A weight-aware recommendation algorithm for mobile multimedia systems," *Mobile Information Systems*, vol. 9, no. 2, pp. 139–155, 2013.
- [50] H. Ince, "Short term stock selection with case-based reasoning technique," *Applied Soft Computing Journal*, vol. 22, pp. 205–212, 2014.
- [51] I. Pereira and A. Madureira, "Self-Optimization module for Scheduling using Case-based Reasoning," *Applied Soft Computing Journal*, vol. 13, no. 3, pp. 1419–1432, 2013.
- [52] R. Bergmann, K.-D. Althoff, S. Breen et al., *Developing Industrial Case-Based Reasoning Applications: The INRECA Methodology*, Springer, 2003.
- [53] A. Aamodt and E. Plaza, "Case-based reasoning: foundational issues, methodological variations, and system approaches," *AI Communications*, vol. 7, no. 1, pp. 39–59, 1994.
- [54] S. Likavec and F. Cena, "Property-based semantic similarity: what counts?" in *CEUR Workshop Proceedings*, pp. 116–124, 2015.
- [55] A. Tversky, "Features of similarity," *Psychological Review*, vol. 84, no. 4, pp. 327–352, 1977.
- [56] K. Christakopoulou, F. Radlinski, and K. Hofmann, "Towards conversational recommender systems," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016*, pp. 815–824, San Francisco, Calif, USA, August 2016.
- [57] B. Kveton and S. Berkovsky, "Minimal interaction search in recommender systems," in *Proceedings of the 20th ACM International Conference on Intelligent User Interfaces (IUI '15)*, pp. 236–246, ACM, Atlanta, Ga, USA, April 2015.
- [58] T. Mahmood, G. Mujtaba, and A. Venturini, "Dynamic personalization in conversational recommender systems," *Information Systems and e-Business Management*, vol. 12, no. 2, pp. 213–238, 2014.
- [59] T. Mahmood and F. Ricci, "Improving recommender systems with adaptive conversational strategies," in *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia (HT '09)*, pp. 73–82, ACM, Turin, Italy, July 2009.
- [60] E. R. Ciurana Simó, *Development of a Tourism Recommender System*, 2012.
- [61] C.-C. Yu and H.-P. Chang, "Personalized location-based recommendation services for tour planning in mobile tourism applications," in *Proceedings of the 10th International Conference on E-Commerce and Web Technologies (EC-Web '09)*, pp. 38–49, Springer, Linz, Austria, 2009.



# Hindawi

Submit your manuscripts at  
<https://www.hindawi.com>

