Research Paper

JIS

HybRecSys: Content-based contextual hybrid venue recommender system

Journal of Information Science
1–15

DOI: 10.1177/0165551518786678
journals.sagepub.com/home/jis
SAGE

Aysun Bozanta
Department of Management Information Systems, Bogazici University, Turkey

Birgul Kutlu
Department of Management Information Systems, Bogazici University, Turkey

Abstract
The popularity of location-based social networks has prompted researchers to study recommendation systems for location-based services. When used separately, each existing venue recommendation system algorithm has its own drawbacks (e.g. cold start, data sparsity, scalability). Another issue is that critical information about context is not commonly used in venue recommendation systems. This article proposes a hybrid recommendation model that combines contextual information, user-based and item-based collaborative filtering and content-based filtering. For this purpose, we collected user visit histories, venue-related information (distance, category, popularity and price) and contextual information (weather, season, date and time of visits) related to individual user visits from Twitter, Foursquare and Weather Underground. Experimental evaluation of the proposed hybrid system (HybRecSys) using a real-world dataset shows better results than baseline approaches.

Keywords
Collaborative filtering; content-based filtering; context; hybrid recommendation; location; venue

1. Introduction

Location-related technologies such as global positioning system (GPS) on mobile devices have created new possibilities for user interaction. These technologies provide location and time information, making it possible for users to share their location. With the increasing use of location-based social networks has come an abundance of information about user locations and preferences. Location recommendation systems have been developed that make personalised location suggestions to users.

Recommendation systems use mainly three algorithm types: collaborative filtering (CF) systems, content-based filtering (CBF) systems and hybrids [1]. Contextual information is thought to increase the performance of recommendation systems [1], but most recommendation engines fail to consider it. Each algorithm has its own drawbacks. For instance, CF algorithms have cold start, scalability and sparsity issues, and they lack content-related information. With CBF, an item and its contents need to be machine recognisable and must contain sufficient information, but because this algorithm considers only venue-related characteristics, it may result in a focus that is too narrow.

This study presents a novel hybrid recommendation algorithm that combines user-based CF, item-based CF, CBF and contextual information, with the goal of eliminating the disadvantages of each individual approach. The proposed hybrid system will reduce the number of the drawbacks of each approach when they are used separately. To the best of our knowledge, this is the first study that combines distance, category, popularity and price in one CBF algorithm. Weather conditions, the season, the date and the time of each visit are used as venue features, and the contextual similarities of venues will be utilised in the algorithm.

Corresponding author:
Aysun Bozanta, Department of Management Information Systems, Bogazici University, 34342, Istanbul, Turkey.
Email: aysun.bozanta@boun.edu.tr

Table 1. Most frequently used types of content information on locations.

Types of content information	Related studies
Distance	[11]–[17]
Category	[8,16,18–22]
Tag	[9,23–26]
Tips/comments	[27–30]
Popularity	[16,31–33]
Price	[14,20,34]

2. Related work

CF is commonly used to provide location recommendations. Memory-based CF and model-based CF are subcategories of CF algorithms. Memory-based CF consists of user-based data, which considers user similarity in making recommendations [2–4], and item-based CF, which considers item similarity [5,6]. Model-based CF, however, generally uses data mining techniques such as artificial neural networks [7], naive Bayesian modelling [8,9], association rule mining [10] and singular value decomposition (SVD) [11].

In CBF, content information on locations is used to counteract data sparsity problems that occur in CF algorithms. The most frequently used types of content information on locations are presented in Table 1.

Contexts represent a set of factors that surround user-item pairs and affect the rating of the user to the item accordingly [35]. Contextual information (e.g. time, weather) is crucial, especially for location recommendation systems, because users consider this information in deciding where to go, unlike recommendations for what to buy. In addition, despite the fact that contextual information is critical for recommending locations, it is not commonly used in existing systems. Adomavicius and Tuzhilin [35] have emphasised that a context-independent representation may lose predictive power if potentially useful information from multiple contexts is ignored. Zheng et al. [36] applied algorithms for pre-filtering, context relaxation and hybrid techniques for contextual variables (trip type, origin city and destination city). Using pre-filtering, Barranco et al. [37] used speed and travel direction as contextual variables. Mode of transportation (biking, walking and driving) is another contextual variable that has been used as a pre-filter [38]. Time is commonly used as context data [39–43], but it is used in different forms. For instance, Waga et al. [39] examined the timestamps of photos taken in touristic places. They also considered the season when the photos were taken. Majid et al. [40] discretized time as morning, afternoon, evening and night and days as weekday and weekend. The recommendations were generated by post-filtering those variables. Yuan et al. [41] examined venue similarity for different time periods during the day. Baral and Li [42] recommended locations according to a location category that is mostly visited in a specific time frame, and Hiesel et al. [43] examined location popularity in different time periods. Weather is yet another important contextual variable for location recommendation [44]. Majid et al. [40] used weather information that is specified as temperature and weather conditions. Hiesel et al. [43] examined location popularity in different weather conditions. Other studies have proposed frameworks for contextual location recommendation systems [12–15]. The most commonly used contextual information types for location recommendation are presented in Table 2.

Even though hybrid systems have been discussed in the literature, there are untouched points that would improve the performance of location recommender systems. Hybrid approaches combine at least two existing approaches to minimise or eliminate the drawbacks of those approaches when they are used on their own. Several studies consider different hybrid algorithms for location recommendation [11,16,40,42,43]. Seven types of hybridisation techniques are mentioned in the literature: weighted, switching, mixed, feature combination, cascade, feature augmentation and meta-level [46]. The advantages and disadvantages are shown in Table 3.

For this study, a weighted hybrid approach was applied in a way that allowed us to include various algorithms with different weights. Venue category, price, popularity and distance were used as content-related variables. However, the tags of the venues were not included because they were generated by users and therefore difficult to categorize. Tips/comments were excluded because Foursquare provides only the last five comments about a particular venue, which is insufficient for sentiment analysis. Contextual variables (e.g. trip type, origin city/destination city, speed, travel direction and mode of transportation) were not included because this study does not deal with path planning. Finally, since temperature usually correlates with weather conditions, only time and weather conditions were used as contextual variables. This study presents a hybrid recommendation system that takes into account user and location similarity, location-related features (distance, category, popularity and price) and contextual information (weather, season, date and time of visit) in a weighted hybridisation method in order to achieve better performance with fewer drawbacks than a single method.

Table 2. Most commonly used contextual information for location recommendation.

Context-related variables	References
Time	[39–45]
Weather conditions	[40,43]
Temperature	[40]
Trip type	[36]
Origin city/destination city	[36]
Speed and travel direction	[37]
Transportation type	[38]

Table 3. Advantages and disadvantages of hybridisation techniques [46].

Technique	Advantages	Disadvantages
Weighted	Easy to add all of the system's capabilities	Weight of different techniques does not change over different subjects
Switching	The system can be sensitive to the strengths and weaknesses of its constituent recommenders	Introduces additional complexity into the recommendation process since the switching criteria must be determined
Mixed	Practical to make a large number of recommendations simultaneously	Does not avoid the 'new user' start-up problem
Feature combination	Reduces the sensitivity of the system to the number of users who have rated an item	Gives the system information about the inherent similarity of items that are otherwise opaque
Cascade	The system avoids employing a second, lower priority technique on items that are already well differentiated	Imprecise recommendations resulting from using insufficient numbers of techniques
Feature augmentation	Offers a way to improve core system performance	Can work only when the data include extra features
Meta level	The learned model is a compressed representation of a user's interest, especially for a content/collaborative hybrid	Imprecise recommendations resulting from using insufficient numbers of techniques

3. Material and methods

3.1. Data collection

Data were collected in three stages. First, Twitter was used because it allows direct crawling of its users' check-in history (unlike Foursquare, etc.). Twitter also allows programmers to utilise REST APIs, which are frequently used for designing web APIs to use a pull strategy for data retrieval, and streaming APIs, which are used for continuous streaming of public data with a push strategy. A Twitter dataset was connected via PHP APIs (for Twitter API version 1.1), and MySQL database was used for storing the retrieved data.

Some Foursquare users link their accounts with Twitter and their check-in information can be crawled from Twitter. In order to access these tweets, the REST API 'GET search/tweets' was used. This API returned a collection of tweets that matched a specified query. When a user whose Foursquare account is linked to Twitter makes a check-in using Foursquare, related tweets, including all check-in information, appear on his or her Twitter timeline. The API returned the users who checked in and shared this check-in on their Twitter accounts. Location information shared by users via Foursquare was collected over a period of 2 months. All tweets of collected users were recorded to reach their check-in history.

Venue information on Foursquare was accessed using URLs on check-in tweets. The Foursquare API (https://developer.foursquare.com/docs/api/venues/details) which gives details about a venue was used, and the following attributes were collected:

Venue Name
Category

Table 4. Terms used in this article.

Terms	Definition
$\text{freq}(u_n,v_n)$	Number of visits from *n*th user to *n*th venue
$\min\text{freq}(u_n)$	Minimum number of visits of *n*th user
$\max\text{freq}(u_n)$	Maximum number of visits of *n*th user
$\text{user\_sim}\ (\overrightarrow{u_n}, \overrightarrow{u_m})$	Similarity between *n*th and *m*th users
$\overrightarrow{u_n}$	Vector consisting of ratings of *n*th user
$\overrightarrow{u_m}$	Vector consisting of ratings of *m*th user
$\text{rating}\ (u_n, v_n)$	Rating of *n*th user to *n*th venue
$\text{venue\_sim}(\overrightarrow{v_n},\overrightarrow{v_m})$	Similarity between *n*th and *m*th venues
$\overrightarrow{v_n}$	Vector consisting of ratings of *n*th venue
$\overrightarrow{v_m}$	Vector consisting of ratings of *m*th venue
Check-in count	Total number of check-ins in a specific venue
Like count	Total number of people who liked the specific venue
User count	Total number of unique people that checked in at the specific venue
Tip count	Total number of comments for a specific venue

Latitude
Longitude
Check-in count
Visitor count
Tip count
Price classification

In order to identify the weather conditions at the time of check-ins, the weather history was collected from the Weather Underground website. Each check-in date was matched to the date in the weather history, and the weather information (e.g. sunny, rainy or snowy) was added to the check-in data. Weather Underground provides comma-separated value (CSV) files that include dates and weather conditions. The visit check-in dates were compared with the dates in the weather files, and the coding automatically added the related weather conditions to the check-in data.

3.2. Data preprocessing

Data collected from Twitter and Foursquare were filtered according to the city of the check-in (check-ins were limited to those from Istanbul so as to increase visit frequencies) and the venue category (limited to restaurants because these provided intensive check-in frequency). This preprocess yielded 2882 users, 1992 venues and 8019 visits.

Users who visited only one venue and venues that were visited by only one user were excluded from the dataset, leaving a total of 1101 users, 711 venues and 4694 visits.

The preprocessing of rating, distance, venue and context variables is explained in detail in the following subsections. The terms used in this study are shown in Table 4.

3.2.1. The rating preprocess. First, the visit frequency of user-venue pairs was normalised linearly, and frequency values were converted to ratings ranging from 1 to 5. A rating of 1 was assigned when the maximum and minimum frequencies of an individual's visits were equal

$$\text{rating}(u_n, v_n) = \left(\left(\frac{\text{freq}(u_n, v_n) - \min\text{freq}(u_n)}{\max\text{freq}(u_n) - \min\text{freq}(u_n)}\right) * 4\right) + 1 \tag{1}$$

3.2.2. The distance preprocess. The latitude and longitude values of check-ins were converted to *x, y* and *z* coordinates for each visit. User centres were calculated by taking the average of all visits of a specific user. For each visit of a specific user, the Euclidean distance from the venue to the user centre was determined.

3.2.3. The venue property preprocess. The popularity variable was created from four venue properties that were collected using the Foursquare API, namely, the check-in count, the like count, the user count and the tip count. A principal

Table 5. KMO and Bartlett's test results.

KMO measure of sampling adequacy		0.821
Bartlett's test of sphericity	Approximate chi square	13,951.963
	Degrees of freedom	6
	Significance	0.000

KMO: Kaiser–Meyer–Olkin.

Table 6. Total variance explained.

Component	Variance (%)	Cumulative variance (%)
1	93.69	93.69
2	4.46	98.15
3	1.08	99.19
4	0.80	100

Table 7. Component matrix for popularity.

Check-in count	0.965
Like count	0.985
User count	0.981
Tip count	0.941

component analysis (PCA), a multivariate analysis used for dimension reduction [47], was performed on 711 venues using the four venue properties. The value of the Kaiser–Meyer–Olkin (KMO) measure was 0.821 where the acceptable level is generally 0.6. Bartlett's test of sphericity was significant at 1% alpha level (Table 5), indicating that the sample was adequate for PCA.

The results show that 93.69% of total variance is explained by a single component (Table 6), which we refer to as popularity. It consists of four variables.

High values on check-in count, like count, user count and tip count mean that these variables are overwhelmingly explained by a single component (Table 7).

The popularity value of each venue was calculated from the results of the PCA, whereupon those values were normalised on a scale from 1 to 5.

A total of 34 restaurant categories were collected with the Foursquare API and included in the dataset. A user-category matrix that shows how many venues each user visited in each category is prepared.

Foursquare has four price categories: cheap, average, expensive and very expensive. The price category of each venue was collected from the Foursquare API, and a user-price matrix was prepared to show how many venues each user visited in each price category.

3.2.4. The contextual variable preprocess. Season, day, time frame and weather conditions are contextual variables used in this recommender system. The Unix time format provided by Twitter was converted to date and time stamp. Season was determined by the check-in date. Some periods have a similar number of check-ins; for example, some users have the same pattern of check-in behaviour for every day of a given week. Discretization of contextual variables performs better, so it was applied to all contextual variables. From the check-in date, days were determined and discretized as 'weekday' and 'weekend' [43,45]. Spring and summer were discretized as 'hot season', and autumn and winter were discretized as 'cold season' [45]. Majid et al. [40] discretized time as morning, afternoon, evening and night, whereas they discretized a day as morning and evening. However, examining check-in behaviours in the dataset, it is found that discretizing data as morning, noon and evening was more suitable for our purposes. The time range 07:00 to 11:59 was specified as morning, 12:00 to 16:59 as noon and 17:00 to 06:59 as evening.

More than 10 different weather conditions (e.g. sunny, rainy, snowy, rainy and stormy snowy and stormy) were included in the dataset. These conditions were discretized in three broad categories: sunny, rainy and snowy. A dataset

Table 8. Sample contextual characteristics of venues.

Venue	H	C	Wday	Wend	M	N	E	S	R	SW
1	0.75	0.25	0.68	0.32	0.08	0.17	0.75	0.67	0.25	0.08
2	0.5	0.5	0.5	0.5	0	1	0	1	0	0
3	0.83	0.17	0.96	0.04	0.13	0.52	0.35	0.65	0.35	0
4	0.33	0.67	1	0	0	0.33	0.67	0.67	0.33	0

H: hot; C: cold; Wday: weekday; Wend: weekend; M: morning; N: noon; E: evening; S: sunny; R: rainy; SW: snowy.

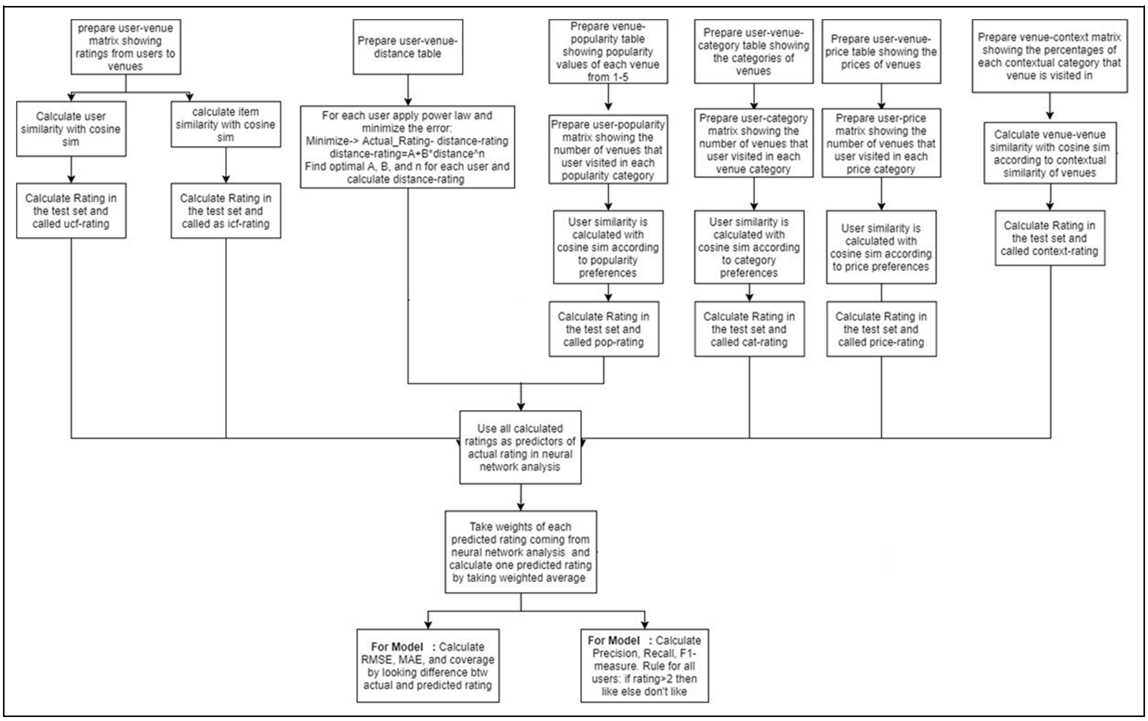

Figure 1. Flowchart of HybRecSys.

that included contextual information for each venue was prepared, and the context percentages of each category were calculated (Table 8).

3.2.5. Development of the HybRecSys recommender system. Figure 1 presents the flowchart of the HybRecSys location recommendation system by explaining each step in the algorithm. Moreover, Figure 2 presents the framework of the HybRecSys location recommendation system, which consists of three main components: user similarities (based on check-in history, category, popularity and user price preferences), venue similarities (based on check-in history and venue context characteristics) and distance from the venue to the user centre.

3.2.6. Calculating user similarities. User similarities can be measured using Jaccard similarity, Cosine distance, Euclidean distance or correlation distance. Cosine distance, used to measure the degree of similarity between two vectors of an inner product space that measures the cosine of the angle between them [48], was employed in this study. First, we constructed the user-venue matrix, which keeps track of the user ratings to venues. User–user similarity was calculated with equation (2), using the user-venue matrix

$$\mathrm{user\_sim}\left(\overrightarrow{\mathrm{u}_n},\ \overrightarrow{\mathrm{u}_m}\right) = \cos\left(\overrightarrow{\mathrm{u}_n},\ \overrightarrow{\mathrm{u}_m}\right) = \frac{\overrightarrow{\mathrm{u}_n} \cdot \overrightarrow{\mathrm{u}_m}}{||\overrightarrow{\mathrm{u}_n}|| * ||\overrightarrow{\mathrm{u}_m}||} \tag{2}$$

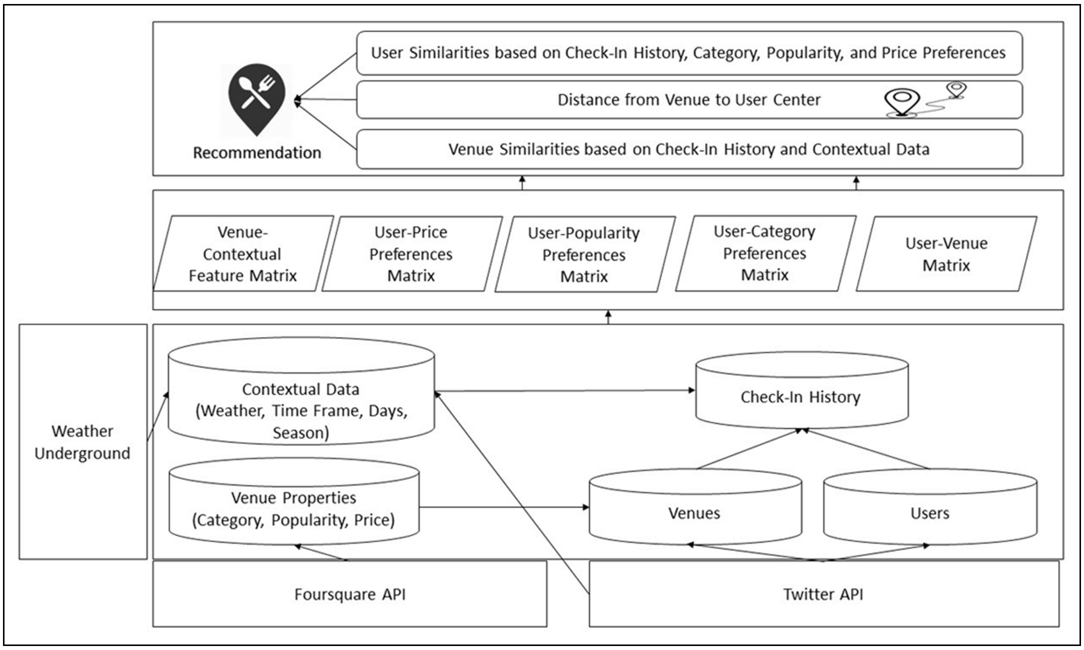

Figure 2. Framework of HybRecSys.

Formula (3) was used to calculate the user ratings of the venues

$$\mathrm{rating}(\mathrm{u}_n, \mathrm{v}_n) = \frac{\sum \mathrm{user\_sim}(\overrightarrow{\mathrm{u}_n}, \overrightarrow{\mathrm{u}_m}) \times \mathrm{rating}(\mathrm{u}_m, \mathrm{v}_n)}{\sum \mathrm{user\_sim}(\overrightarrow{\mathrm{u}_n}, \overrightarrow{\mathrm{u}_m})} \tag{3}$$

Second, we constructed a user-category preference matrix that would keep track of the number of venues a specific user visits in each category. Popularity values were discretized as high, medium or low according to the normalised popularity values obtained from the PCA. Next, we generated a user-popularity preference matrix to keep track of the number of venues that a specific user visited in each popularity category. We also constructed a user-price preference matrix to keep track of the number of venues that a specific user visited in each price category. User–user similarities according to category, popularity and price preferences were also calculated, using a technique similar to the one used in equation (2). Predicted ratings using user similarity depend on the category, popularity and price preferences of users. These ratings were also calculated in a similar manner, using equation (3).

3.2.7. Calculating venue similarities. As with user similarities, we calculated venue similarity by utilising the cosine distance from the user-venue matrix. Venue–venue similarity was calculated using equation (4)

$$\mathrm{venue\_sim}\left(\overrightarrow{\mathrm{v}_n}, \overrightarrow{\mathrm{v}_m}\right) = \cos\left(\overrightarrow{\mathrm{v}_n}, \overrightarrow{\mathrm{v}_m}\right) = \frac{\overrightarrow{\mathrm{v}_n} \cdot \overrightarrow{\mathrm{v}_m}}{||\overrightarrow{\mathrm{v}_n}|| * ||\overrightarrow{\mathrm{v}_m}||} \tag{4}$$

The predicted user ratings of venues were calculated using the following formula

$$\mathrm{rating}(\mathrm{u}_n, \mathrm{v}_n) = \frac{\sum \mathrm{venue\_sim}(\overrightarrow{\mathrm{v}_n}, \overrightarrow{\mathrm{v}_m}) \times \mathrm{rating}(\mathrm{u}_n, \mathrm{v}_m)}{\sum \mathrm{venue\_sim}(\overrightarrow{\mathrm{v}_n}, \overrightarrow{\mathrm{v}_m})} \tag{5}$$

Next, we prepared a venue–context matrix (Table 8) to keep track of venue contextual characteristics. This matrix presents the percentage of venue preferences in different contextual circumstances. With this matrix, venue similarities were calculated using equation (4). Predicted ratings that depended on the contextual similarity of venues were calculated using equation (5).

3.2.8. Constructing the distance equation. The calculation of ratings according to distance between a venue and a user assumed that if the distance between a venue and the user centre was short, the user would visit that venue more frequently [48–50]. Users are typically more willing to check in at venues near their centres, but each user's perception of distance is different. For this reason, a power law distribution [41,49,51] was fitted to each user's visits, and optimal coefficients (*A, B* and *n*) were found for each user to model the willingness of a user to check-in at a place to minimise the difference between actual rating and predicted rating

$$\text{rating}_{\text{distance}} = A + B * \text{distance}^n \tag{6}$$

3.2.9. Developing the hybrid approach. In this study, we applied a weighted hybrid recommender that computed the score of recommended items from the results of all available recommendation techniques. Instead of using equal weights for each algorithm, we applied an artificial neural network analysis in order to find the optimal weights for each technique. A generalised feed forward artificial neural network model was applied using Weka[1] software. The following parameters were used

Learning rule: Momentum (momentum factor = 0.5);
Stopping criteria: Mean square error (minimum MSE = 0.005);
Learning rate: 0.1;
Activation function: Linear sigmoid;
Initial weight: Randomised;

The results of all available recommendation techniques were used as inputs, and actual ratings were used as output to be predicted. Final ratings were calculated by multiplying the weights coming from the results of the artificial neural networks by the ratings from various algorithms, as shown in equation (7)

$$\begin{aligned}\text{rating} = w_1 * \text{rating}_{\text{ucf}} + w_2 * \text{rating}_{\text{icf}} + w_3 * \text{rating}_{\text{distance}} + w_4 \\ * \text{rating}_{\text{popularity}} + w_5 * \text{rating}_{\text{category}} + w_6 * \text{rating}_{\text{price}} + w_7 * \text{rating}_{\text{context}}\end{aligned} \tag{7}$$

Pseudocode of the HybRecSys algorithm can be found in Table 9.

4. Experiments and evaluation

In this section, baseline approaches and evaluation methods are described and major findings are presented.

4.1. Baseline approaches

The HybRecSys was compared with a user-based *K*-nearest neighbourhood (KNN) algorithm, an item-based KNN algorithm, a biased matrix factorisation and a SVD++. These algorithms are used for rating prediction that is suitable for our dataset and are ready to use in the LibRec,[2] a Java library for recommender systems. The default settings of LibRec was used for each algorithm. Setting parameters are presented in Table 10.

Because it was difficult to find the coding of other hybrid algorithms in previous studies, the HybRecSys was created step by step. In each step, the performance improvement was observed with the additional variables. Earlier versions of the HybRecSys (referred to here as Hybrid 1, 2 and 3) are explained in the following:

Hybrid 1 was created from a combination of only $\text{rating}_{\text{ucf}}$ and $\text{rating}_{\text{icf}}$.
Hybrid 2 was created by the combination of $\text{rating}_{\text{ucf}}$, $\text{rating}_{\text{icf}}$ and $\text{rating}_{\text{distance}}$.
Hybrid 3 was created by the combination of $\text{rating}_{\text{ucf}}$, $\text{rating}_{\text{icf}}$, $\text{rating}_{\text{distance}}$, $\text{rating}_{\text{popularity}}$, $\text{rating}_{\text{category}}$ and $\text{rating}_{\text{price}}$.

The HybRecSys was created from the combination of $\text{rating}_{\text{ucf}}$, $\text{rating}_{\text{icf}}$, $\text{rating}_{\text{distance}}$, $\text{rating}_{\text{popularity}}$, $\text{rating}_{\text{category}}$, $\text{rating}_{\text{price}}$ and $\text{rating}_{\text{context}}$. It was also compared with its earlier versions.

Table 9. Pseudocode of HybRecSys algorithm.

Algorithm: HybRecSys

1. User-based collaborative filtering
   Input: user-venue rating matrix
   Output: rating_{ucf}
   - Calculate the user similarities with Cosine similarity
   - Using the user similarities calculate ratings in the test set
2. Item-based collaborative filtering
   Input: user-venue rating matrix
   Output: rating_{icf}
   - Calculate the venue similarities with Cosine similarity
   - Using the venue similarities calculate ratings in the test set
3. Category
   Input: user-category preferences matrix
   Output: $\text{rating}_{category}$
   - Calculate the user similarities with Cosine similarity according to the category preferences
   - Using the user similarities, calculated ratings in the test set
4. Popularity
   Input: user-popularity preferences matrix
   Output: $\text{rating}_{popularity}$
   - Calculate the user similarities with Cosine similarity according to the popularity preferences
   - Using the user similarities, calculated ratings in the test set
5. Price
   Input: user-price preferences matrix
   Output: rating_{price}
   - Calculate the user similarities with Cosine similarity according to the price preferences
   - Using the user similarities, calculated ratings in the test set
6. Contextual information
   Input: venue-context matrix
   Output: $\text{rating}_{context}$
   - Calculate the venue similarities with Cosine similarity according to the contextual characteristics of the venues
   - Using the user similarities, calculated ratings in the test set
7. Distance
   Input: user-venue distance list
   Output: $\text{rating}_{distance}$
   - For each user, fit power law distribution
   - For each user, solve GRG nonlinear programming equation
8. Finding final predicted rating
   Input: rating_{ucf}, rating_{icf}, $\text{rating}_{category}$, $\text{rating}_{popularity}$, rating_{price}, $\text{rating}_{context}$, $\text{rating}_{distance}$
   Output: final rating
   - Apply artificial neural network analysis by using given inputs to find out final ratings

Table 10. Setting parameters of algorithms in LibRec.

User-based KNN	Item-based KNN	Biased matrix factorisation	SVD++
K = 5	*K* = 5	num.factors = 10	num.factors = 10
Similarity = PCC	Similarity = PCC	num.max.iter = 200	num.max.iter = 200
Num.shrinkage = 30	Num.shrinkage = 30	learn.rate = 0.01 -max -1 -bold-driver	learn.rate = 0.01 -max -1 -bold-driver
Num.neighbours = 50	Num.neighbours = 50	reg.lambda = 0.1 -u 0.1 -i 0.1 -b 0.1 -s 0.001	reg.lambda = 0.1 -u 0.1 -i 0.1 -b 0.1 -s 0.001

KNN: *K*-nearest neighbourhood; SVD: singular value decomposition.

4.2. Evaluation

Generally, three types of experimental settings are used for evaluating recommender systems: (1) offline experiments that use a pre-collected dataset of users' choices and rated items, (2) user studies where a set of test subjects is recruited and asked to perform several tasks requiring an interaction with the recommendation system and (3) an online evaluation

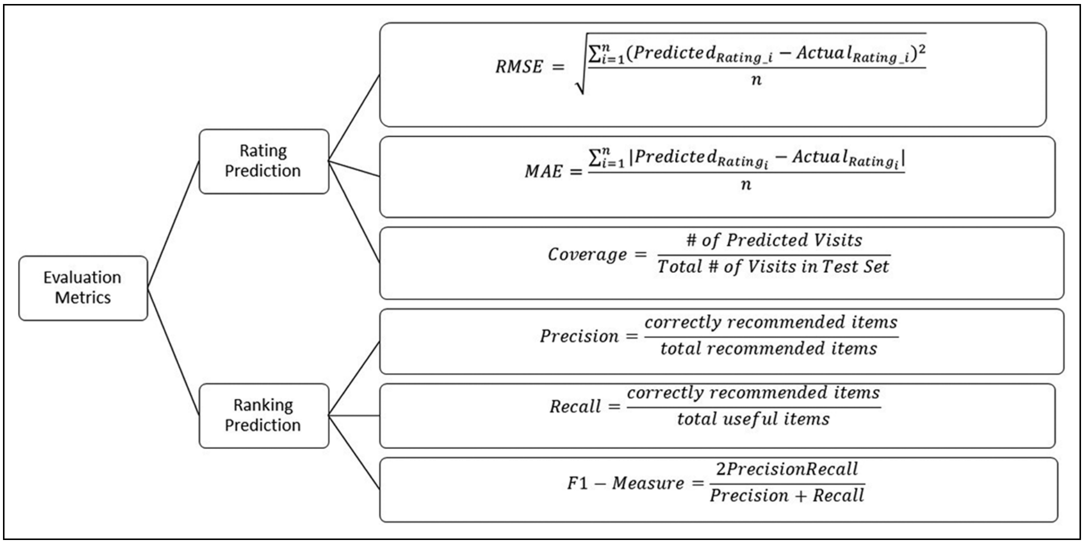

Figure 3. Evaluation metrics.

Table 11. RMSE, MAE and coverage values based on different algorithms.

Approaches	RMSE	MAE	Coverage (%)
User-based KNN [53]	1.296077	0.866262	38
Item-based KNN [54]	1.308751	0.869726	38
Biased matrix factorisation [55]	1.433791	0.893339	100
SVD++ [56]	1.426615	0.886675	100
Hybrid 1	1.295172	0.865241	49
Hybrid 2	1.281491	0.855446	100
Hybrid 3	1.254821	0.835233	100
HybRecSys	1.214821	0.8058951	100

RMSE: root-mean-squared error; MAE: mean absolute error; KNN: *K*-nearest neighbourhood; SVD: singular value decomposition.

that redirects a small percentage of the traffic to different alternative recommendation engines and records user interactions with those different systems [52]. Offline experiments were used in this study to validate the algorithm. To evaluate recommender systems, user ratings can be predicted and compared with actual user ratings (rating prediction); alternatively, a recommended set of items can be compared with the actual items in the user sets (ranking prediction). Different metrics were used for rating and ranking prediction. Metrics used for the evaluation of the HybRecSys are presented in Figure 3.

The root-mean-squared error (RMSE) is one of the most popular metrics used for evaluating the accuracy of predicted ratings. The mean absolute error (MAE) is an alternative to RMSE. *Coverage* indicates what percentage of visits can be predicted in all test sets. *Precision* indicates the percentage of correctly recommended items over total recommended items, while *Recall* shows the percentage of recommended items over the total number of liked items by the user. The F1-Measure, calculated using both precision and recall, measures the accuracy of the system.

The K-fold ($K = 10$) cross-validation technique was used to split the data into training and test sets. The dataset was split into 10 disjoint sets, making sure that each set contained about 10% of each user's visits. For each fold, one set was used as a test set, and nine were used as training sets.

The RMSE, MAE and coverage values of the algorithms available in LibRec, earlier versions of HybRecSys (Hybrid 1, Hybrid 2 and Hybrid3) and the latest version of HybRecSys (HybRecSys) are presented in Table 11.

According to the RMSE and MAE values, the HybRecSys outperforms other algorithms. User-based KNN, item-based KNN, SVD++ and biased matrix factorisation follow HybRecSys according to RMSE and MAE values. The coverage is 100% for the HybRecSys. Although the RMSE and MAE values of user-based and item-based KNN are slightly higher than those of the HybRecSys, their coverage value is only 38%. The coverage value of SVD++ and biased matrix factorisation is also 100%, but their RMSE and MAE values are higher than those of HybRecSys.

Table 12. Precision, recall and F1 measures of the algorithms.

Approaches	Precision	Recall	F1 measure
User-based KNN [53]	0.1220	0.0823	0.0983
Item-based KNN [54]	0.1154	0.1107	0.1130
Biased matrix factorisation [55]	0.0893	0.1200	0.1031
SVD++ [56]	0.0702	0.0976	0.0816
Hybrid 1	0.1203	0.1097	0.1147
Hybrid 2	0.1299	0.1113	0.1199
Hybrid 3	0.1417	0.1301	0.1356
HybRecSys	0.1667	0.1460	0.1493

KNN: *K*-nearest neighbourhood; SVD: singular value decomposition.

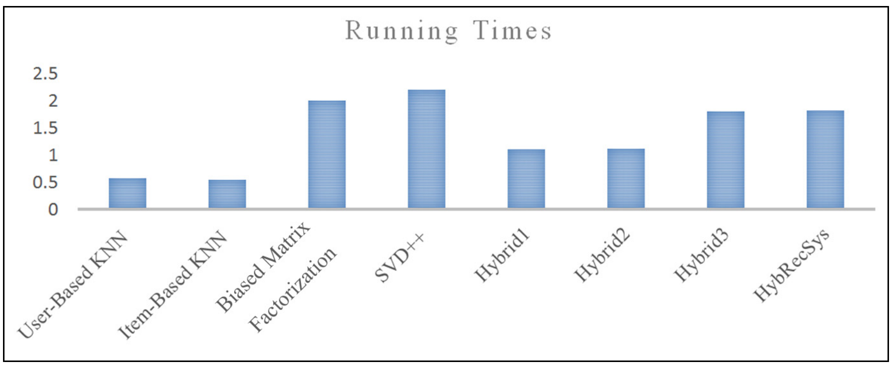

Figure 4. Running times of the algorithms.

When HybRecSys is compared with its earlier versions, it is clear that the addition of each variable increases the performance of the algorithm. The RMSE and MAE values decrease most dramatically after the addition of venue-related features (price, popularity and category). The HybRecSys algorithm performs better than its earlier versions and also outperforms other algorithms.

The precision, recall, and F1 values of the algorithms available in LibRec, earlier versions of the HybRecSys and the current HybRecSys are presented in Table 12.

According to precision, recall and F1 values, the HybRecSys outperforms all other algorithms. User-based KNN, item-based KNN, biased matrix factorisation and SVD++ follow HybRecSys according to precision metrics. According to recall values, biased matrix factorisation, item-based KNN, SVD++ and user-based KNN follow the HybRecSys. Finally, according to the F1 measure, item-based KNN, biased matrix factorisation, user-based KNN and SVD++ follow HybRecSys.

When the precision, recall and F1 values of the HybRecSys are compared with its earlier versions, it clearly outperforms the earlier versions.

HybrecSys is also compared with the other algorithms according to the running times. Running times in seconds are presented in Figure 4. It is obvious that running times of item-based KNN and user-based KNN is lower than other algorithms. However, SVD++ and biased matrix factorisation have the highest values of running times. Hybrid 1, Hybrid 2, Hybrid 3 and HybRecSys follow user-based KNN accordingly. They also have lower running times than SVD++ and biased matrix factorisation. Although HybRecSys does not have the lowest running time, since its other metrics are better compared with other algorithms, it can be still preferred.

5. Conclusion

This study proposed a hybrid recommendation model HybRecSys that combines user-based CF, item-based CF, CBF and contextual information in order to eliminate the disadvantages of each approach. For this purpose, users' visit history, venue-related information (distance, category, popularity and price classification) and contextual information (weather, season, date and time of visits) related to each visit were collected from different sources (Twitter, Foursquare and Weather Underground). For CBF, the variables of distance, category, popularity and price, which had not previously been combined in a single algorithm, were used to determine content-based similarity of users by embedding more venue-related features. Weather conditions, the season, date and the time of each visit were used as venue features. Contextual similarities of venues were also utilised in the system. A weighted hybrid recommender that computed the score of recommended items from the results of all available recommendation techniques was applied. Instead of using equal weights for each technique, artificial neural network analysis was performed in order to find their optimal weights. The results of all available recommendation techniques were used as inputs, and actual ratings were the outputs that are predicted. Final ratings were calculated by multiplying the weights coming from the results of neural networks by the ratings from different algorithms. The hybrid system mitigated the drawbacks of each approach that occur when they are used separately. The use of multiple data sources provided variables that improved the performance of the recommendation system.

The HybRecSys was compared with four existing algorithms, namely, user-based and item-based KNN, biased matrix factorisation and SVD++, and with its previous versions. The four algorithms were evaluated using two types of metrics, rating prediction (RMSE, MAE and coverage) and ranking prediction (precision, recall and F1 measure). The K-fold cross-validation ($K = 10$) technique was used for data splitting and training, and test datasets were generated. We conducted extensive experiments to evaluate the effectiveness of the HybRecSys and to validate its advantages over other algorithms. The results of each metric show that the HybRecSys outperforms its earlier versions as well as the other four existing algorithms. The HybRecSys effectively overcomes the challenges arising from data sparsity by modelling user preferences from the venue category, popularity and price.

Data sparsity, which resulted from an insufficient number of user-rated venues, was compensated for by calculating different similarity measures (user, item, price, category, popularity-based and contextual similarity values). The problem of over-specialization caused by the recommendation of highly similar venues was circumvented by taking into account the preferences of users from numerous different aspects, not venue characteristics. The quality of recommendation was thus improved.

The contribution of this study can be summarised as follows:

Four different approaches (user-based CF, item-based CF, CBF and contextual recommendation) were combined to develop a hybrid recommender system.

To the best of our knowledge, this is the first study that combines the variables of distance, category, popularity and price classification in a CBF algorithm with the aim of determining content-based similarity of users by embedding a venue-related feature.

Weather conditions, the season, the date and the time of each visit were combined as venue features. Contextual similarities of venues were also utilised in the system.

Our study is not without limitations. Although the size of the collected data is large, after the filtration it became relatively small. However, this small dataset generated meaningful results and it is possible that it may produce better results with larger datasets. In addition, HybRecSys algorithm was unable to generate recommendations for new users. Cold start problem will be addressed in a future study. Second, contextual variables were not used in a personalised way, which means a user may like a particular venue in one contextual circumstance, where another may like the same venue in a different context. For all similarity calculations, Cosine similarity was used. The performance might improve if other similarity measures (adjusted cosine similarity, Jaccard similarity, etc.) are implemented. When it came to venue opening and closing hours, these were not checked because it is not possible to know whether the venue was open the entire day. Finally, we aim to improve the performance of the HybRecSys by using real user reaction via user surveys.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Funding

This study was supported by the Bogazici University Scientific Research Projects Fund (project no. 11463).

Notes

1. https://www.cs.waikato.ac.nz/ml/weka/
2. http://www.librec.net. Version 1.3 of LibRec

References

1. Adomavicius G and Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE T Knowl Data En* 2005; 17(6): 734–749.
2. Li Q, Zheng Y, Xie X et al. Mining user similarity based on location history. In: *Proceedings of the 16th ACM SIGSPATIAL international conference on advances in geographic information systems*, Irvine, CA, 5–7 November 2008, Article No. 34. New York: ACM.
3. Ye M, Yin P and Lee WC.Location recommendation for location-based social networks. In: *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, San Jose, CA, 2–5 November 2010, pp. 458–461. New York: ACM.
4. Hasegawa T and Hayashi T.Collaborative filtering based spot recommendation seamlessly available in home and away areas. In: *2013 IEEE/ACIS 12th International Conference on Computer and Information Science (ICIS)*, Niigata, Japan, 16–20 June 2013, pp. 547–548. New York: IEEE.
5. Levandoski JJ, Sarwat M, Eldawy A et al.LARS: a location-aware recommender system. In: *2012 IEEE 28th international conference on data engineering (ICDE)*, Washington, DC, 1–5 April 2012, pp. 450–461. New York: IEEE.
6. Takeuchi Y and Sugimoto M.CityVoyager: an outdoor recommendation system based on user location history. In: Ma J, Jin H, Yang LT et al. (eds) *Ubiquitous intelligence and computing*. Berlin: Springer, 2006, pp. 625–636.
7. Knoch S, Chapko A, Emrich A et al. Context-aware running route recommender learning from user histories using artificial neural networks. In: *2012 23rd international workshop on Database and Expert Systems Applications (DEXA)*, Vienna, 3–7 September 2012, pp. 106–110. New York: IEEE.
8. Gupta A and Singh K.Location based personalized restaurant recommendation system for mobile environments. In: *2013 international conference on advances in computing, communications and informatics (ICACCI)*, Mysore, India, 22–25 August 2013, pp. 507–511. New York: IEEE.
9. Subramaniyaswamy V, Vijayakumar V, Logesh R et al. Intelligent travel recommendation system by mining attributes from community contributed photos. *Procedia Comput Sci* 2015; 50: 447–455.
10. Saraee M, Khan S and Yamaner S.Data mining approach to implement a recommendation system for electronic tour guides. In: *Proceedings of the 2005 international conference on e-business, enterprise information systems, e-government, and outsourcing, EEE 2005*, Las Vegas, NV, 20–23 June 2005, pp. 215–218. Las Vegas, NV: CSREA Press.
11. Sattari M, Toroslu IH, Karagoz P et al. Extended feature combination model for recommendations in location-based mobile services. *Knowl Inf Syst* 2015; 44(3): 629–661.
12. Wang H, Li G and Feng J.Group-based personalized location recommendation on social networks. In: Chen L, Jia Y, Sellis T et al. (eds) *Web technologies and applications*. Cham: Springer, 2014, pp. 68–80.
13. Yin H, Cui B, Sun Y et al. LCARS: a spatial item recommender system. *ACM T Inform Syst* 2014; 32(3): 11.
14. Park MH, Hong JH and Cho SB.Location-based recommendation system using Bayesian user's preference model in mobile devices. In: Indulska J, Ma J, Yang LT et al. (eds) *Ubiquitous intelligence and computing*. Berlin: Springer, 2007, pp. 1130–1139.
15. Aihara K, Koshiba H and Takeda H.Behavioral cost-based recommendation model for wanderers in town. In: *Proceedings of the 14th international conference on human-computer interaction. Towards mobile and intelligent interaction environments*, Orlando, FL, 9–14 July 2011, pp. 271–279. Berlin: Springer.
16. Yu Z, Feng Y, Xu H et al. Recommending travel packages based on mobile crowdsourced data. *IEEE Commun Mag* 2014; 52(8): 56–62.
17. Yin H, Cui B, Chen L et al. Modeling location-based user rating profiles for personalized recommendation. *ACM T Knowl Discov D* 2015; 9(3): 19.
18. Bao J, Zheng Y and Mokbel MF.Location-based and preference-aware recommendation using sparse geo-social networking data. In: *Proceedings of the 20th international conference on advances in geographic information systems*, Redondo Beach, CA, 6–9 November 2012, pp. 199–208. New York: ACM.
19. Yin H, Sun Y, Cui B et al.LCARS: a location-content-aware recommender system. In: *Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining*, Chicago, IL, 11–14 August 2013, pp. 221–229. New York: ACM.
20. Kuo MH, Chen LC and Liang CW. Building and evaluating a location-based service recommendation system with a preference adjustment mechanism. *Expert Syst Appl* 2009; 36(2): 3543–3554.
21. Shimada K, Uehara H and Endo T.A comparative study of potential-of-interest days on a sightseeing spot recommender. In: *2014 IIAI 3rd international conference on advanced applied informatics (IIAIAAI)*, Kitakyushu, Japan, 31 August–4 September 2014, pp. 555–560. New York: IEEE.
22. Ahmedi L, Rrmoku K, Sylejmani K et al. A bimodal social network analysis to recommend points of interest to tourists. *Soc Netw Anal Min* 2017; 7(1): 14.
23. Cao L, Luo J, Gallagher AC et al. A worldwide tourism recommendation system based on geotagged web photos. In: *2010 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, Dallas, TX, 14–19 March 2010, pp. 2274–2277. New York: IEEE.

24. Guo L, Shao J, Tan KL et al. WhereToGo: personalized travel recommendation for individuals and groups. In: *IEEE 15th international conference on mobile data management*, Brisbane, QLD, Australia, 14–18 July 2014, pp. 49–58. New York: IEEE.
25. Memon I, Chen L, Majid A et al. Travel recommendation using geo-tagged photos in social media for tourist. *Wireless Pers Commun* 2015; 80(4): 1347–1362.
26. Xu Z, Chen L and Chen G. Topic based context-aware travel recommendation method exploiting geotagged photos. *Neurocomputing* 2015; 155: 99–107.
27. Dhake B, Lomte SS, Auti RA et al. LARS: an efficient and scalable location-aware. *Int J Sci Res Educ* 2014; 2(11): 2371–2378.
28. Sarwat M, Levandoski JJ, Eldawy A et al. LARS*: an efficient and scalable location-aware recommender system. *IEEE T Knowl Data En* 2014; 26(6): 1384–1399.
29. Krishna PV, Misra S, Joshi D et al. Learning automata based sentiment analysis for recommender system on cloud. In: *International conference on computer, information and telecommunication systems (CITS)*, Athens, 7–8 May 2013, pp. 1–5. New York: IEEE.
30. Mordacchini M, Passarella A, Conti M et al. Crowdsourcing through cognitive opportunistic networks. *ACM T Auton Adap Sys* 2015; 10(2): 13.
31. Zheng Y, Zhang L, Ma Z et al. Recommending friends and locations based on individual location history. *ACM T Web* 2011; 5(1): 5.
32. Zheng VW, Zheng Y, Xie X et al. Towards mobile intelligence: learning from GPS history data for collaborative recommendation. *Artif Intell* 2012; 184: 17–37.
33. Korakakis M, Spyrou E, Mylonas P et al. Exploiting social media information toward a context-aware recommendation system. *Soc Netw Anal Min* 2017; 7(1): 42.
34. Yu CC and Chang HP.Personalized location-based recommendation services for tour planning in mobile tourism applications. In: *International conference on electronic commerce and web technologies*, Linz, 1–4 September 2009, pp. 38–49. Berlin: Springer.
35. Adomavicius G and Tuzhilin A.Context-aware recommender systems. In:Ricci F, Rokach L, Shapira B et al. (eds) *Recommender systems handbook*. Boston, MA: Springer, 2015, pp. 191–226.
36. Zheng Y, Burke R and Mobasher B.Differential context relaxation for context-aware travel recommendation. In: *International conference on electronic commerce and web technologies*, Vienna, 4–5 September 2012, pp. 88–99. Berlin: Springer.
37. Barranco MJ, Noguera JM, Castro J et al. A context-aware mobile recommender system based on location and trajectory. In: Casillas J, Martínez-López F and Corchado Rodríguez J (eds) *Management intelligent systems*. Berlin: Springer, 2012, pp. 153–162.
38. Savage NS, Baranski M, Chavez NE et al. I'm feeling loCo: a location based context aware recommendation system. In: Gartner G and Ortag F (eds) *Advances in location-based services*. Berlin: Springer, 2012, pp. 37–54.
39. Waga K, Tabarcea A and Franti P.Context aware recommendation of location-based data. In: *2011 15th international conference on system theory, control, and computing (ICSTCC)*, Sinaia, 14–16 October 2011, pp. 1–6. New York: IEEE.
40. Majid A, Chen L, Chen G et al. A context-aware personalized travel recommendation system based on geotagged social media data mining. *Int J Geogr Inf Sci* 2013; 27(4): 662–684.
41. Yuan Q, Cong G, Ma Z et al.Time-aware point-of-interest recommendation. In: *Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval*, Dublin, 28 July–1 August 2013, pp. 363–372. New York: ACM.
42. Baral R and Li T.MAPS: a multi aspect personalized POI recommender system. In: *Proceedings of the 10th ACM conference on recommender systems*, Boston, MA, 15–19 September 2016, pp. 281–284. New York: ACM.
43. Hiesel P, Braunhofer M and Worndl W.Learning the popularity of items for mobile tourist guides. In: *Proceedings of RecTour*, Boston, MA, 15 September 2016.
44. Trattner C, Oberegger A, Eberhard L et al. Understanding the impact of weather for POI recommendations. In: *Proceedings of RecTour*, Boston, MA, 15 September 2016.
45. Baltrunas L and Amatriain X.Towards time-dependent recommendation based on implicit feedback. In: *Workshop on context-aware recommender systems (CARS '09)*, New York, 25 October 2009.
46. Burke R. Hybrid recommender systems: survey and experiments. *User Model User-Adap* 2002; 12(4): 331–370.
47. Wold S, Esbensen K and Geladi P. Principal component analysis. *Chemometr Intell Lab* 1987; 2(1–3): 37–52.
48. Yuan Q, Cong G and Sun A. Graph-based point-of-interest recommendation with geographical and temporal influences. In: *Proceedings of the 23rd ACM international conference on information and knowledge management*, Shanghai, China, 3–7 November 2014, pp. 659–668. New York: ACM.
49. Mao Y, Yin P, Lee W et al. Exploiting geographical influence for collaborative point-of-interest recommendation. In: *Proceedings of the 34th ACM SIGIR international conference on research and development in information retrieval*, Beijing, China, 24–28 July 2011, pp. 325–334. New York: ACM.
50. Zheng Y, Zhang L, Xie X et al.Mining interesting locations and travel sequences from GPS trajectories. In: *Proceedings of the 18th international conference on World Wide Web*, Madrid, 20–24 April 2009, pp. 791–800. New York: ACM.
51. Gorakala SK and Usuelli M. *Building a recommendation system with R*. Birmingham: Packt Publishing Ltd., 2015.

52. Shani G and Gunawardana A.Evaluating recommendation systems. In:Ricci F, Rokach L, Shapira B et al. (eds) *Recommender systems handbook*. Boston, MA: Springer, 2011, pp. 257–297.
53. Konstan JA, Miller BN, Maltz D et al. GroupLens: applying collaborative filtering to Usenet news. *Commun ACM* 1997; 40(3): 77–87.
54. Deshpande M and Karypis G. Item-based top-n recommendation algorithms. *ACM T Inform Syst* 2004; 22(1): 143–177.
55. Koren Y, Bell R and Volinsky C. Matrix factorization techniques for recommender systems. *Computer* 2009; 42(8): 42–49.
56. Koren Y.Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining*, Las Vegas, NV, 24–27 August 2008, pp. 426–434. New York: ACM.