ISSN 1846-6168 (Print), ISSN 1848-5588 (Online)
https://doi.org/10.31803//tg-20210706082307

Original scientific paper
Received: 2021-07-06, Accepted: 2021-09-25

Developing a Location-Based Recommender System Using Collaborative Filtering Technique in the Tourism Industry

Iman Kianinezhad*, Mehdi Bayati, Ali Harounabadi, Donya Akbari

Abstract: The rapid growth of new information and products in the virtual environment has made it time consuming to acquire relevant information and knowledge amidst a vast amount of information. Therefore, an intelligent system that can offer the most appropriate and desirable among the large amount of information and products by following the conditions and features selected by each user should be essentially efficient. Systems that perform this task are called recommendation systems. Given the volume of social network data, challenges such as short-term processing and increased accuracy of recommendations are discussed in this type of system. Hence, it can perform processes faster with less error and can be effective in improving the performance of social recommending systems in improving the classification and clustering of information with the help of collaboration filtering methods. This study first develops an innovative conceptual model of a social network-based tourism recommendation system using Flicker network data. This model is based on 9 key components. The comparison show that the proposed method has an accuracy of 0.3% and a lower error rate.

Keywords: Collaborative Filtering Algorithm; DB Scan Clustering; Haversine; Multi-Factor Systems; Recommender Systems; Similarity Criteria; Tourism

1 INTRODUCTION

The recommender system is a necessity and a popular technology that, by collecting data from activities and inclinations, gains the interest of its customer from a set of data such as movies, hardware, clothing, etc. and makes offers to various customers. Currently, huge growth of content and the number of users in the internet world is undeniable. For this reason, mechanisms were created to filter the information available on the Internet. Collaboration filtering is one of the techniques used in this regard. Collaboration filtering came from the idea that people often get the best recommendations from people with the same taste. Collaboration filtering includes techniques for matching people with similar tastes and interests and providing recommendations based on that [1]. Recommendation systems have incorporated themselves into people's daily online activities and have been used successfully in a wide range of different fields. However, there is still work to be done to achieve great success in complex fields such as tourism. One reason is that properties of items in this field are not easily measurable and depend on a large number of factors. When you plan to travel, you ask friends or tour guides about attractive places or attractions, and then select the attractions you want. Recommending systems suggest the best and most accurate location based on your taste and characteristics to have a pleasant and satisfying experience [1]. Recently, recommending systems are increasingly used in the field of e-tourism. In this field, services such as travel advice, a list of points of interest that match the user's taste, and recommendations for tourist packages, etc. are provided. Recommendation systems available in the tourism industry offer the best to the user based on tourist destination, time limit and certain budget. The user typically expresses her needs, interests, and limitations based on selected parameters. Once the user makes her selection, the system links the list of specified destinations using the same parameter vector [2].

This study tends to develop a recommendation system using the pre-filtering approach by DBScan clustering and Haversine criterion and to improve accuracy of the proposed model using asymmetric similarity criterion in collaboration filtering approach. Using demographic information, this study tends to solve the problem of cold start. A hybrid approach used to tourism recommendation system based on background information, collaboration filtering and demographic information is another measure addressed in this study.

2 LITERATURE REVIEW

Currently, web users are faced with many options when surfing the web. Hence, recommendation systems (RS) and many web personalization tools provide web users with customized items. These systems are available on many websites that cover social networks, e-commerce, e-business, e-tourism, etc. [3]. In essence, RS compares users based on an appropriate similarity criterion, which is an important rule for success of the whole system. However, different similarity criteria often lead to different sets of neighbors for a particular active user. A good similarity criterion produces a close set of neighbors for a particular active user. In fact, many of the similarity criteria for collaboration recommender systems rely on overlap between users. However, the size of this overlap is not examined in detail, and most previous work has studied similarity criteria based on a predefined number of common items [3].

There are currently four types of recommendation systems:

- Content-based recommenders
- Recommenders based on collaboration filtering
- Hybrid systems
- Knowledge-based filtering.

These examples have different input data and each suggests different algorithms. Their input mainly includes user information (demographics), item (service content, description), content (location, time, activity) and feedback.

The collaborative filtering recommender system suffers from data shortages due to its reliance on numerical rankings to provide recommendations to users. This problem also makes the exact calculation of similar neighbors difficult for the system and reduces the quality of the offers. Existing methods are not able to process the missing new item ranking and cold start forecast for active users, which ultimately leads to poor quality recommendations. Therefore, quality of the results is prevented from declining by using the double-walled clustering algorithm to cluster the rating matrix of users and items, and the missing ranks in dual clusters are estimated and filled by using the Bi-Mean algorithm [4]. Chaudhary and Anupama developed a popularity model based on review and collaborative filtering by presenting an in-depth study and analyzing different algorithms. The proposed model is a straight forward model based on popularity that ranks the goods that the user rates. This model has recently become popular and has been used by companies such as Netflix, Amazon and Facebook [5]. Jin et al. [6] developed a new method of measuring similarity that effectively uses user textual information. Their method uses a unique factor to formulate a nonlinear equation and takes into account the user rating habits. This method can improve accuracy of predictions and is tested on a set of data and compared with other algorithms. The results indicate that their method can improve quality of the recommendation. Tohidi and Dadkhah [7] proposed a mobile recommender system called BomApettite to recommend a restaurant to a group based on tastes of all members of the group. They combined restaurant information from a well-known platform. By evaluating the system, it was found that reliability of the system was increased. Advantages of the systems are the use of collaborative filtering, which is one of the most widely used methods of the recommender system, to recommend a new item to the user.

3 METHODOLOGY

This study tends to predict the best attraction according to geographical location of the target user. For this purpose, a recommender system is proposed which suggests the best location by pre-filtering approach using DBScan clustering and the Haversine criterion, as well as improving the accuracy of the proposed model using the asymmetric similarity criterion in collaborative filtering approach. Currently, there is a wide range of information on the Internet; thus, the use of recommender systems will help users to find the most suitable location according to basic needs of the user. This can be done by analyzing the user's own profiles, previous searches, preferences, comments and user interactions with other services and users.

Tourism recommender systems mainly contain information such as user profile (tourist), item (tourist destination), time, user activities on social networks and weather conditions. The information used by the recommender system can be collected through feedback, ranking, social media or implicitly through data collection programs, user browsing history and agents. According to explanations provided in the field of the proposed algorithm and how DB Scan clustering works, different steps of this algorithm is shown in Fig. 1.

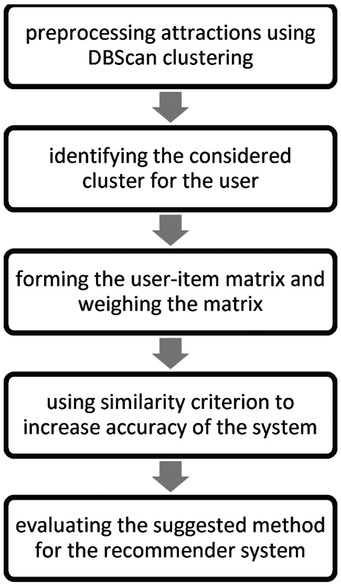
preprocessing attractions using DBScan clustering
identifying the considered cluster for the user
forming the user-item matrix and weighing the matrix
using similarity criterion to increase accuracy of the system
evaluating the suggested method for the recommender system

Figure 1 The suggested algorithm flowchart

User (Tourist)

There are two types of target users: domestic and foreigner. It includes individuals or groups with social and demographic information related to the tourist destination. This information is mainly used to model user ratings by date and preferences. The recommender system uses this user model to provide sufficient recommendations about the tourist destination. User information is defined by the following variables (as shown in Fig. 2):

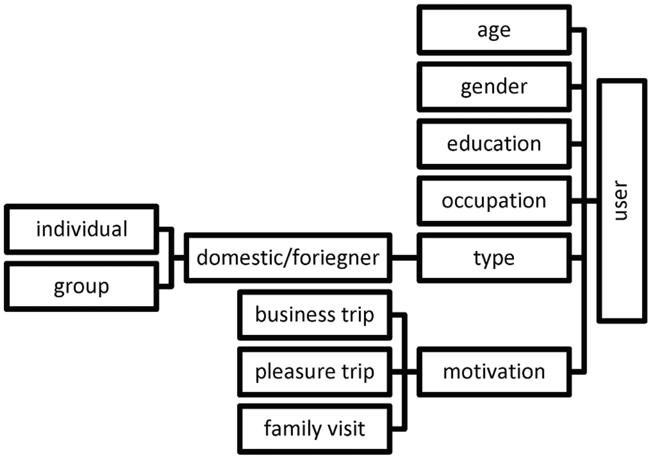

Figure 2 User variables

Item (Destination)

The location can be explicitly checked by visiting an account (such as Foursquare, Facebook) or geotags (Flickr), or implicitly, data can be collected from browsing history, agents, location-based services, location data collection programs or through a sensor in the form of mobile data from GPS, telecommunication towers, Wi-Fi, or telecom operators. Locations have many characteristics such as:

- Specific locations (user or item): any valuable information (geography, address)

- Geography may have value (continent, country, province, and city) or values such as tourist address, location, coverage and social network control.

Time

Time information is defined as calendar, day, period, and hour. Course attributes include period (morning, noon, evening, night), as well as other features such as calendar, day, and hour. This data is explicitly obtained by time tags (Facebook) or implicitly by browser history or collector program.

3.1 Inserting a Set of User Comments for Different Items in a Specific Field

At this stage, rates will be taken from the user in a specific field, such as time, geographical location, time, etc. In fact, at this stage, the user will assign a point to items from a specific field, which can indicate the importance of the item. This set of ratings will form an item matrix:

$$\boldsymbol{R}(User-Item)=\begin{bmatrix} d_{m_1 n_1} & \cdots & d_{m_1 n_n} \\ \vdots & \ddots & \vdots \\ d_{m_n n_1} & \cdots & d_{m_n n_n} \end{bmatrix} \tag{1}$$

Where, $\boldsymbol{R}$ is known as item-user matrix, m refers to the number of rows, which is the same number of users, and n refers to the number of columns, which are the same as the specific items in a particular field. $d_{i,j}$ also refers to the rate given by the i^{th} user to the j^{th} item.

3.2 Clustering User-Item Matrix Using the Proposed Algorithm

Once the matrix of ratings of different users in different items is formed, the users will be clustered using this matrix. It is very important in clustering algorithms to properly determine the number of clusters. In fact, the proper number of clusters will optimize the clustering algorithm process. In fact, to determine the number of clusters, we first enter the maximum number of C and then consider the maximum number of Cs from 1 to C. Suppose the data of a set is clustered by a method. For each data i, $a(i)$ is the average dissimilarity of data i to other data in the same cluster. Then we get the average of dissimilarity of data i with the data of other clusters. Of these numbers, we find the lowest number and call it $b(i)$.

To explain the algorithm, it is necessary to be familiar with the parameters ε and μ, which are explained:

- Each point in the data is apart from other points. Any point whose distance from a given point is less than ε (EPS) is considered as the neighbor of that point.
- Any assumed point (MinPoints) that has μ neighbors is a central point.

The relationship of points is divided into three categories based on their position (being central or not) in each cluster:

Connected points: A point is connected to a cluster that is first the central point, second adjacent to one of the points within the cluster.

Accessible points: A point is accessible to a cluster that is not the central point but is adjacent to one of the points within the cluster.

If the point does not have any of the above positions, noise is considered for that cluster. Moreover, if the point is noise relative to all clusters, it is placed in the noise cluster.

In the clustering algorithm, an initial population will be created. This population will be produced by Eq. (2):

$$X=Uniform\_Random\_Number\,(Lower, Upper) \tag{2}$$

Where, the *Uniform_Random_Number* function will generate a random number with a uniform distribution, X represents the population, and *Lower* and *Upper* will indicate the lower and upper limits, respectively, to generate normal random numbers.

Once the initial population is generated, as noted, the fitness function of the population will be calculated using DBScan algorithm. The distance of the target user's location from the centers of the cluster is calculated using the Haversine equation to identify the nearest cluster. The Haversine equation is used in the study of spherical geometry, especially in calculating the distance between two points distributed outside the sphere. Eq. (2) calculates the distance based on latitude and longitude. We considered points P1 and P2 as two points in space. φ_1 is latitude and λ_1 is longitude of point P1 and φ_2 is latitude and λ_2 is longitude of point P2. Radius of the earth is denoted by r. The function input is obtained from Eq. (3). Using the Haversine equation, the distance of the target user is identified based on geographical location with the center of the clusters and the nearest cluster.

$$d=2r\cdot sin^{-1}\left(\sqrt{\begin{matrix} haversin\,(\varphi_2-\varphi_1)+ \\ +cos\,(\varphi_1)\,cos\,(\varphi_2)\,haversin\,(\lambda_2-\lambda_1) \end{matrix}}\right) \tag{3}$$

$$haversin\,(\theta)=sin^2\left(\frac{\theta}{2}\right) \tag{4}$$

Once the matrix of distance between the two vectors is calculated, the smallest distance will be selected as the best solution for this population. From the solutions obtained for different cases of algorithm iterations, the least distance will be selected as the best solution and the vector obtained from this least solution will be considered as the result of clustering.

3.3 Rating New Users

After clustering the users in the considered clusters, the new user rating will be estimated. To do this, it is first determined which cluster the user is in. Once the user cluster is determined, the number of k users with the highest similarity to the user will be identified, and then the new user rating will be estimated based on average of ratings. Similarity criterion of two users will be calculated by Eq. (5).

$$D_r(a, b) = \sqrt{\sum_{i=1}^{r}(a_i - b_i)^2} \tag{5}$$

Where, variables a_i and b_i refer to similar properties for two users, d indicates the dimension space of the problem, and r refers to the number of dimensions involved in calculating similarity.

3.4 Recommending Item to New User

Once the new user is placed in the considered cluster and the neighbors with high similarity to this user are found, first items of users with high similarity to the new user are sorted in descending order based on their rate and a new item is suggested to users. Then, based on the number of items recommended to the user, it is recommended to the new user from the beginning of the set of items. The similarity criterion is used to introduce users with high similarity through the Pearson criterion. Pearson correlation coefficient for two numerical variables is in the range of 1 to –1, 1 means complete agreement and –1 indicates complete disagreement. If the users are x and the new user is y, the Pearson criterion is defined as:

$$r = \sum_{i=0}^{n}(x_i - x)(- y) \Big/ \sqrt{\sum (x_i - x)}\sqrt{\sum (y_i - y)} \tag{6}$$

4 RESULTS

4.1 Computer Specifications

Implementations of this study were done with a computer with following specifications: 3.8 GHz processor, 8 GB internal memory and Windows 7 operating system. To run the proposed algorithm, the programming language and MATLAB version 2018b were used. The Flickr website dataset was also used for the case study.

4.2 Dataset Descriptions

All tourist destinations in the UK in images of 10 thousand Flickr database, including 914 points from 24 different cities, were grouped in 7 clusters. The value of ε, which represents the radius of the neighborhood, was 999 meters, and value of the MinPts parameter was 30. The dataset contained 943 users who visited and rated at least one of the 914 points. In fact, considering the rates given to the locations by different users, an R matrix consisting of n users and m locations is formed that each cell of this matrix is the rate of the image given to it by the user. In fact, this matrix is formed using the rates that users gave to different places on the Flickr tourism social network.

4.3 Implementing the Proposed Algorithm

We must first run the proposed algorithm on the Flickr dataset and evaluate the results. Tab. 1 shows the input values for the proposed algorithm:

Table 1 Input values of the proposed algorithm

Parameter	Value
Number of clusters	7
Number of users	943
Number of locations	914
Neighborhood radius	999
Number of recommendations	20
Number of iterations	100
Population size	10

In fact, the data set is initially divided into 7 clusters. The number of users in this data set is 943 and the number of locations is 914. When a new user logs in and is categorized, the closest similarity for this new user is determined using the Haversine similarity criterion. The number of recommendations to the new user logged in is 20. The population size in the classification algorithm is equal to 10 and the number of iterations of this algorithm is equal to 100.

4.4 Evaluating the Proposed Algorithm

The following parameters are used to evaluate the proposed algorithm:

Mean Absolute Error (***MAE***): The mean of the measured error is equal to the rate that the system predicts that the user will give to the item minus the actual rate that the user has given to the item and is shown as Eq. (7).

$$MAE = \frac{\sum_{i=1}^{n}\left|value_{actual} - value_{predicted}\right|}{n} \tag{7}$$

where, $value_{actual}$ shows the actual value of the user's rating for the item and $value_{predicted}$ represents the value suggested by the recommender system.

Accuracy: The number of favorable predictions divided by total predictions that will be calculated as Eq. (8).

$$Accuracy = 1 - \left[\frac{\sum_{i=1}^{n}\left|value_{actual} - value_{predicted}\right|}{n}\right] \tag{8}$$

First, the mean error of the recommendations made by the proposed recommender system was calculated and then this error value was subtracted from one and accuracy was calculated.

Root Mean Square Error (RMSE): In Eq. (9), $R_{u,i}$, the value of r indicates whether item i has been rated by user u in the available data. The lower the $RMSE$ value, the higher the accuracy of the recommender system results.

$$RMSE = \sqrt{\frac{\sum_{(u,i)|R_{u,i}} (\hat{r}_{u,i} - r_{u,i})^2}{N}} \tag{9}$$

***Coverage* percentage**: Another evaluation criterion of the recommender system is "coverage percentage", the value of which represents the percentage of <item, user> pairs in the evaluated data for which the recommender system can predict a rating to total number of <item, user> pairs in the dataset evaluated.

$$Caverage = \\ = \frac{number\ of\ (u,i)\ pairs\ with\ solution}{total\ number\ of\ (u,i)\ pairs\ existing\ in\ the\ evaluated\ set} * 100 \tag{10}$$

Precision: This criterion has values in the range [0, 1] and indicates the degree of error. A value of zero indicates an absolute error and a value of one indicates the absence of an error. Assuming that the output of the recommender system is an integer in the range [1, 5], the value of precision can be defined in the form of Eq. (11).

$$Precision = 1 - \frac{RMSE}{4} \tag{11}$$

The number 4 in denominator of the fraction indicates maximum amount of error (4 = 5 – 1).

F-Measure: *F-Measure* is calculated to combine the coverage percentage and *RMSE* in the form of a criterion. To do this, *RMSE* must be converted to precision introduced above and displayed as a number in the range [0, 1]. Therefore, *F-Measure* equation is defined as Eq. (12):

$$F\text{-} Measure = \frac{2 * Coverage * Pr ecision}{Coverage + Pr ecision} \tag{12}$$

4.5 Output Results

Once the classification program, the specifications of which are given in Tab. 2, is run, the classification operation is performed and the value of Best for each iteration of this algorithm is obtained as follows:

According to Fig. 3, the value of Best is shown in each iteration. In fact, this value indicates the within-class distance that decreases with each iteration, which will have optimal results for the proposed algorithm. The Mean vector also represents the median distances within the classes in each iteration of the proposed algorithm. Once the proposed algorithm is run, we need to calculate fit indices such as *Precision*, *Error*, *MAE*, and *RMSE* and evaluate the results. Tab. 2 evaluates the results.

Table 2 Output results from running the program

Parameter	Value
Precision	0.93414
Error	0.30512
MAE	1.1754
RMSE	1.5685

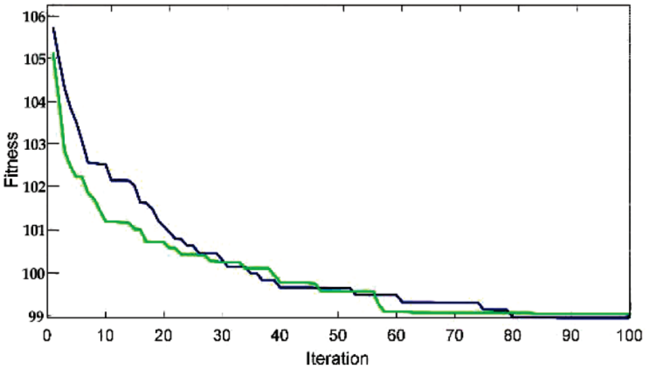

Figure 3 Value of Best per iteration of the proposed algorithm

To further explain the program outputs, assume the test data set in batches of 50, the results will be specified in the form of diagrams shown in Fig. 4.

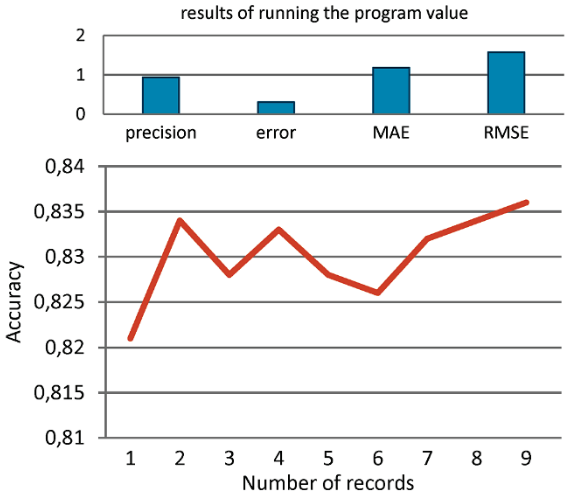

Figure 4 Diagram of precision for different number of test records

As shown in Fig. 4, precision ascends from the batch 6 and increases. In batches 1 and 3, the accuracy of recommendations to the user increases, and in batches 2, 4 and 5, this precision decreases. The reason for this decrease and increase can be attributed to similar neighbors to the new user, as well as location that these neighbors visited and suggested to the new user. Note that the horizontal axis shows the number of records in batches of 50.

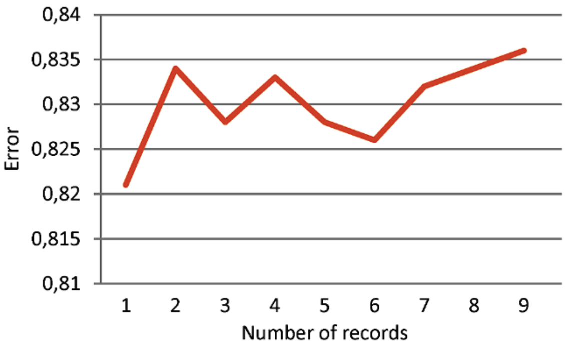

Figure 5 MAE for different number of records of the test dataset

As shown in Fig. 5, MAE of the proposed program is first increased and then decreased with the increase in the number of test data set records. This means that the location recommended to users in the first batches has an increasing error rate, and then the locations recommended to users in the last batches are more correct, which reduces the error. Fig. 6 properly shows RMSE.

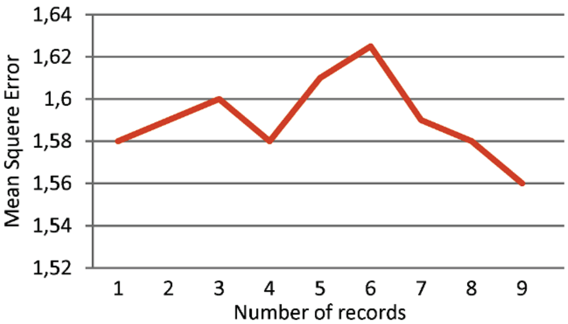

Figure 6 RMSE for different number of records of the test dataset

As shown in Fig. 6, RMSE first increases and then decreases as the number of test dataset records increases. The reason for this is that the locations recommended to the users in the first batches have a higher error rate and the locations recommended to the users in the last batches have a lower error rate.

Table 3 Results of running the proposed algorithm and other methods

Algorithm Parameter	Number of neighbours	DBCACF	Proposed
MAE (Neighbour)	10	0.71	0.716
	20	0.6402	0.6413
	30	0.7331	0.7352
	40	0.7544	0.7422
	50	0.7636	0.7683
	60	0.7836	0.7758
	70	0.7973	0.7945

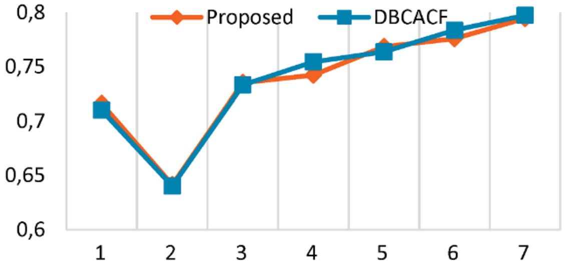

Figure 7 Results of running the proposed algorithm and the other method

Once the proposed algorithm is run on the dataset and the results are reviewed, in this step we must compare and analyze the results of running the proposed algorithm with the algorithm. Note that DBCACF algorithm is run on the dataset and the results are measured. Tab. 3 shows the results properly.

As shown in Tab. 3 and Fig. 7, MAE of the proposed method for different number of neighbors is less than the method presented in Tab. 3 and thus it is more optimal.

Tab. 4 and Fig. 8 compare RMSE of the proposed method and other methods, and shows that RMSE of the proposed method is less than the method in Tab. 4 and is more optimal.

Table 4 Comparison of RMSE

Algorithm parameter	Number of neighbours	DBCACF	Proposed
RMSE (Neighbour)	10	1.1839	1.1822
	20	1.13	1.1354
	30	1.1207	1.1197
	40	1.0998	1.0899
	50	1.0868	1.0749
	60	1.0752	1.0697
	70	1.0683	1.0623

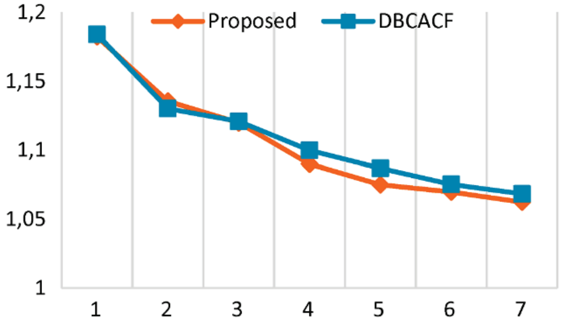

Figure 8 comparison of RMSE

As shown in Tab. 5 and Fig. 9, precision of the proposed method for different number of neighbors is slightly higher and more optimal than the compared method.

Table 5 Comparison of precision for the proposed methods

Algorithm parameter	Number of neighbours	DBCACF	Proposed
Precision (Neighbour)	10	0.2012	0.2141
	20	0.24945	0.2601
	30	0.31355	0.31628
	40	0.35235	0.3664
	50	0.35935	0.36425
	60	0.36378	0.36949
	70	0.36895	0.37345

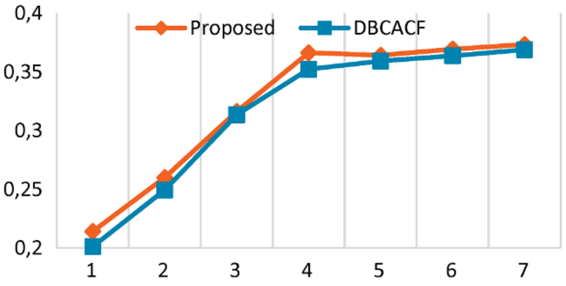

Figure 9 Comparison of precision for different methods

5 CONCLUSION

This study tended to develop and implement a tourism recommender system in a completely dynamic and flexible manner. Because of that, it receives the interests and priorities of each tourism very personally, fix the cold start problem seen in these tourism systems by properly assessing the behavior of new users, and perform tourism planning according to the information obtained from each person. In addition to its ability to recommend places of interest to users, this system allows tourists to manage their time and personal planning. One of the main goals of this study is to solve the problem of cold start of new users, of whom there is little information in the system. Collaborative filtering algorithm is run by forming a database of priorities of users for goods and services. This technology has been very successful in both research and implementation, as well as in information filtering applications. However, important research questions remain in the face of two fundamental challenges of collaborative filtering systems. As explained, user dataset and its items were first clustered by clustering with DBScan algorithm. Next, the degree of similarity of the appropriate items for the new user was determined using the Haversine similarity criterion. Depending on which class the new user is in, new items are recommended to the new user based on k users who are similar to that new user and which items are rated higher. The results of running the proposed algorithm and its comparison with other methods show that the proposed method outperforms in terms of defined fitness indices.

Given the proposed algorithm, advantages, and disadvantages of these methods, implications for future studies include:

- Using new interactive algorithms such as Gray Wolf, Dragonfly algorithms to increase clustering accuracy and running speed
- Using a neural network to respond faster to a new user in order to check the rates of previous users and analyze them for replacement with *K* nearest neighbor algorithm.

Using other methods for measuring similarity matrices to increase accuracy of recommender systems to determine similarity of new user to existing users in classes in order to increase accuracy of measurements in the considered class and consequently increase accuracy of recommendations to the new user

6 REFERENCES

[1] Tvete, A. K. & Pettersen, M. (2016). *A Hybrid Recommender System for Context-Aware Recommendations of Restaurants* (Master's thesis, NTNU).

[2] Gavalas, D., Konstantopoulos, C., Mastakas, K., & Pantziou, G. (2014). Mobile recommender systems in tourism. *Journal of network and computer applications*, *39*, 319-333. https://doi.org/10.1016/j.jnca.2013.04.006

[3] Al-Shamri, M. Y. H. (2016). Effect of Collaborative Recommender System Parameters: Common Set Cardinality and the Similarity Measure. *Advances in Artificial Intelligence (16877470)*. https://doi.org/10.1155/2016/9386368

[4] Roko, A., Almu, A., Mohammed, A., & Saidu, I. (2020). An Enhanced Data Sparsity Reduction Method for Effective Collaborative Filtering Recommendations. *International Journal of Education and Management Engineering*, *10*(1), 27. https://doi.org/10.5815/ijeme.2020.01.04

[5] Chaudhary, S. & Anupama, C. G. (2020). Recommendation System for Big Data Software Using Popularity Model and Collaborative Filtering. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems* (pp. 551-559). Springer, Singapore. https://doi.org/10.1007/978-981-15-0199-9_47

[6] Jin, Q., Zhang, Y., Cai, W., & Zhang, Y. (2020). A new similarity computing model of collaborative filtering. *IEEE Access*, *8*, 17594-17604. https://doi.org/10.1109/ACCESS.2020.2965595

[7] Tohidi, N. & Dadkhah, C. (2020). Improving the Performance of Video Collaborative Filtering Recommender Systems using Optimization Algorithm. *The International Journal of Nonlinear Analysis and Applications (IJNAA)*, 11(1), 283-295. https://doi.org/10.22075/IJNAA.2020.19127.2058

Authors' contacts:

Iman Kianinezhad
(Corresponding Author)
Department of Computer Engineering,
Faculty of Engineering,
University of Applied Sciences & Technology, Ahvaz, Iran
imankianinezhad@gmail.com

Mehdi Bayati,
Department of Computer Engineering,
Faculty of Engineering,
Karoon University, Ahvaz, Iran

Ali Harounabadi,
Department of Computer Engineering,
Faculty of Engineering,
Islamic Azad University Tehran Center Branch, Tehran, Iran

Donya Akbari,
Department of Computer Engineering,
Faculty of Engineering, Hamedan Branch,
Islamic Azad University, Hamedan, Iran