

# Point-of-interest Recommendation in a City

Latifa Baba-Hamed<sup>1</sup>, Djihene Bourenane<sup>1</sup> and Lamia Hamoudi<sup>1</sup>

<sup>1</sup> Computer Science Department, University of Oran 1  
[lbabahamed@yahoo.fr](mailto:lbabahamed@yahoo.fr)

**Abstract.** The recommendation for points of interest has become an important way to help people discover interesting places, especially when users want to visit a city. The purpose of this paper is to provide a recommendation system to help residents of a city or tourists learn about the events/outings available in this city. An evaluation of this system, in terms of precision, is also presented.

**Keywords:** recommendation, content-based filtering, collaborative filtering, profile, preferences, geo-location, geographic database, precision

## 1 Introduction

The recommendation of points of interest is one of the most important topics due to the high value in both research and academy. Recommendation Systems (RS) are powerful tools that help users, in online mode, solve the problem of the information overload they face today with the advent of the Internet, providing them with the most relevant content. To do this, these systems use some information about past user preferences. Recommendation strategies are generally based on collaborative filtering (CF) [2], [3], content-based filtering (CBF) [4], [5] or a combination of these two approaches [1]. The CF recommends products based on the similarity of the preferences of a group of customers (neighbors). This implies that users with common interests in the past will probably continue to share the same tastes in the future. CBF systems use the matching between a user's profile and content descriptors to recommend the appropriate products.

RSs operate on a user-item matrix. When a user enters new votes, his profile is updated by adding new information to that user's current voting vector. The recommendation of points of interest has been addressed, essentially, using GPS trajectory logs [6], [7], [8], [13] or well, using data from social networks location-based (LBSN), such as Foursquare, Loopt and Facebook Places, where users can register at points of interest and share their experiences in the physical world via mobile devices. A synthesis of this work is given in [17].

In this paper, we present a new approach for the recommendation of places of interest in a city to a given user. This method is different from that of the works discussed in the literature as it does not use GPS trajectory logs or LBSN data. We have applied it for the recommendation of places of interest of the Oran city, but the approach remains valid for any city. The scenario envisaged is: a user connects to the

platform and asks for a specific recommendation (eg: I want to go to the cinema) or a general one (eg: where can I go out tonight?). The system will try to exploit the preferences of this user and makes recommendations to him. In addition, it is a question of ensuring the precision of the system's recommendations. Our system makes the match between geo-localization and the recommendation of places in the case of the application of content-based filtering using the category of places (eg. restaurant, beach, etc.) preferred by the active user. In the case of collaborative filtering, our system is based on the preferences of the active user and those of his similar users.

The rest of the paper is organized as follows. In section 2, we present the proposed approach with the different stages of the recommendation process. Section 3 is devoted to the implementation and evaluation of the recommendation process in terms of precision. In section 4, we expose some related work. Finally, Section 5 concludes this paper, recalls our main contributions and gives some perspectives.

## 2 Proposed approach

We recall that our approach aims to provide an online platform to help the inhabitants or visitors of the city of Oran to learn about its events/outings. The first step is to collect the Point-of-interest of different categories of the Oran city. Once the collection is completed, we opted for a geographic database (DBG) for the storage of these places. The next step is to create a graphical interface for querying the database. For this interface we have proposed a geographical map of the city of Oran in which the different places, which respond to the request initiated by user, appear. In this section, we detail, first the structure of the DBG. Then we present our system through the description of its architecture.

### 2.1 Building the database

The construction of the DB required administrative documents relating to the information of the outings of Oran city collected from different administrations. The collect of events was done from websites, social networks or even from advertising posters. In order to visualize these places and these socio-cultural activities on a map, we have geo-located, each of them, with two important dimensions (latitude and longitude). We structured this data in the form of tables. We have thus conceived a main table called "Point-interest" which brings together all the places collected. This table admits the attributes: *Pointid* (identifier of the point of interest), *Pointname* (its name), *Category* (its category), *Address* (its address), *phone* (its phone number), and *Latlong* (geometric column containing its latitude and its longitude). We have also, created the tables specifying each category of the interest point, namely, the tables: Hotel, Restaurant, Amusement\_park, Beach, Green\_area, Cinema, Theater, Museum, Culture, Library, Mall. Figure 1 summarizes the schema for this database.

In addition, we have created another table, in this base, called "Evenement" listing all the cultural events of Oran city. This table includes the attributes: *E\_id* (Event identifier), *E\_title* (its title), *E\_category* (its category; e.g. art, music, etc.), *Start\_Date*

(its start date), *End\_Date* (its end date), *Start\_time* (its start time), *End\_time* (its end time), *Type\_person* (the type of person destined for this event) and *Pointid* (identifier of the point of interest where the event takes place). This table can extend to other events added (by the administrator) from a form (see Figure 2).

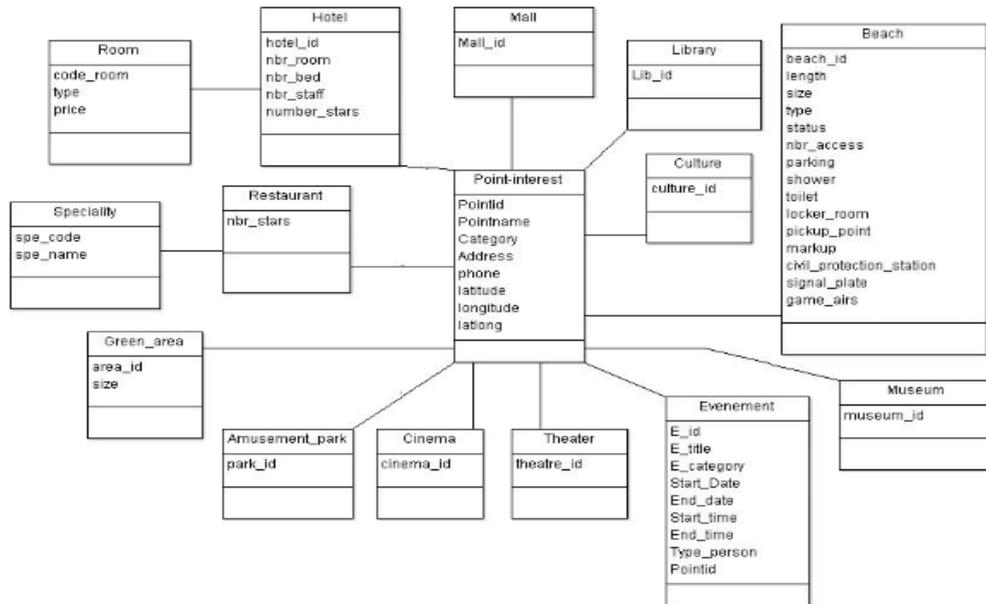


Fig. 1. Database schema.

### Add new event

Event title:

Category:

Start date:

End date:

Start time:

End time:

Type person:

Pointid:

Fig. 2. Event Entry form.

## 2.2 Architecture of the system

The architecture of our system is divided into two parts: a user space and an administrator space. We detail its components in the Figure 3.

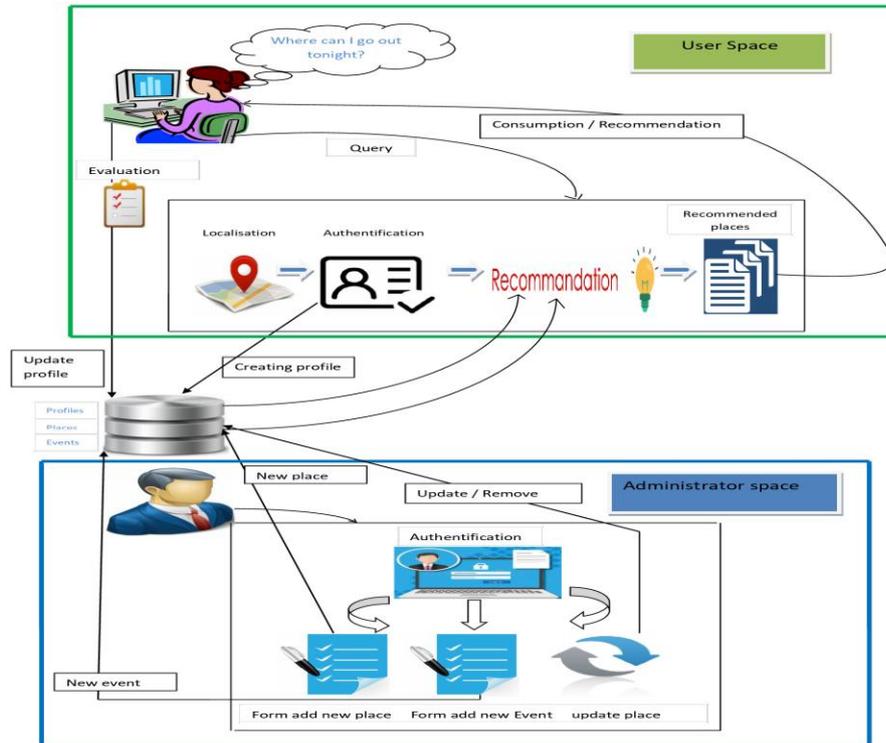


Fig. 3. Architecture of the recommendation system.

The user space is dedicated to active users who are looking to have recommendations for places of exits. The recommendation is calculated using two methods: Content-based filtering and collaborative filtering.

### 2.2.1 Content-based recommendation

In this method the recommendations are calculated according to the preference of the user as well as the distance between the latter and the different places of exit.

The reference is chosen from the different categories of the places listed in our database. The distance is calculated by the system by considering the position of the active user and that of the places of the category chosen by the latter.

The system sorts these places according to their distances and recommends to the user a list of the top-K places. This list will be consumed and evaluated by the user.

The evaluation consists of the assignment of a note between 1 and 5 at the place visited (1: The user has not liked at all and 5: The user has enjoyed much). This note will then be used to update the user's profile.

In order to calculate the distance between the user and the places of interest, we used the predefined function of POSTGRESQL "*St\_distance ()*".

This function has two parameters, the first containing the geographical point of the user, the second one contains that of the place of interest.

The calculation of the distance requires the geometric points of the places as well as that of the user. Those of the places are more or less fixed, the user meanwhile, he can be at any place and therefore each time his position must be geo-localized.

For geo-localization, we used a JavaScript function with the Google API. The function returns the latitude and longitude of the user and the system takes care of turning them into a geometric point.

## 2.2.2 Recommendation based on collaborative filtering

Collaborative filtering relies on the profiles of other system users that are similar to the current user.

The active user must enter his preferences by assigning a rating of 0 to 5 for a few exit places. Note 0 is given to a place not yet visited, 1 and 2 are negative values, 3 is a neutral value, 4 and 5 to express the fact that the user really appreciates the place. These preferences will serve as the initial profile for this user.

This profile will be completed later when the user evaluates the places that will be recommended. The places recommended to the user are the places appreciated by the users most similar to the latter. An appreciated place has a rating of more than 3.

In our application; we model the profile of a user  $U$  in the form of a vector  $V$  of size  $n$  where  $n$  represents the number of places of interest.  $V [i]$  represents the user's appreciation note for place  $i$ . This note is equal to 0 if  $U$  does not know place  $i$ ; it takes a value from 1 to 5 according to the degree of appreciation of  $U$  for place  $i$ .

The similarity between two users is calculated according to their respective profiles.

We have chosen, in our case, the cosine similarity measure which is frequently used in the domain of information retrieval for its satisfactory results (see Formula 1). If the similarity between two profiles exceeds the threshold set to 0.6 (in our application), then we deduce that these two users are similar.

$$\text{Cos}(x,y) = \frac{\sum_{i=0} x_i * y_i}{\sqrt{\sum_{i=0} x_i^2 * \sum_{i=0} y_i^2}} \quad (1)$$

With  $x$  and  $y$  representing two vectors.

We presented user ratings in the form of an evaluation matrix. The rows represent the users and the columns represent different places to evaluate. A small example is presented in Figure 4. The evaluation matrix lists all users of the system. It helps us to make the recommendation of places to an active user. In effect, when a user connects to our system, his profile is compared to all user profiles in the evaluation matrix to

retain only similar users (closest neighbors). Then the system recommends to him all the places of outings appreciated by his neighbors and that he has not yet visited.

It should be noted that a change of scale is necessary to reduce the values from 1 to 5 between 0 and 1. Preferences must be between 0 and 1.

		Places						
		Hotel Meridien	Restaurant Martinez	Cinema Murjajo	Amusement park Miramar	Beach Madegh	Mall Ritaj	Library Cathedrale
Users	U1	0.6	0	0.2	0.2	0.8	0.4	0
	U2	0.8	0	0.4	0.2	0.6	0.6	0
	U3	0	0.4	0.6	0	1	0	0.8
	U4	0.4	0	1	0.4	0	0	0.2

Fig. 4. Example on the evaluation matrix.

The administrator space is reserved for the administrator who is responsible for managing the contents of the database. Different treatments have been assigned to the latter so that the recommendations posted to the users may be adequate.

The administrator has the option to add a new place using the form in Figure 5: any new place must be registered in the database. The administrator can also delete a place (for example, a place that has been demolished), or to change information about a place (for example, a hotel that changes the phone number must be updated).

### Add new place

Point name :

Category : Library ▼

Adress :

Phone :

Latitude :

Longitude :

Fig. 5. Form to add a new place.

### **3 Implementation and evaluation**

In this section we expose the various development tools and languages that we have used to implement our application. We also show how we proceeded to evaluate our system.

#### **3.1 Tools and languages used**

The languages used are: HTML to create the interface of our recommendation system, JavaScript to determine the position of the user, and also for the definition of the Google APIs, and finally PHP to manage the interactions of the user with our System.

The tools used are: The Apache server that allows the use of several languages interpreted as PHP. It also helps to manage the interactions between users and the system. To manage our database, we used the DBMS Postgres with its extension PostGIS which allows the manipulation of spatial data and supports geographic objects.

#### **3.2 Evaluation of the recommendation process**

Evaluation of recommendation systems is an important step in the recommendation process. Its purpose is to reflect the performance of the system which must effectively predict items and contribute to the satisfaction of the needs of the users. The criteria chosen for evaluating our recommendation system is precision.

Precision measure consists in evaluating the difference between the predicted notes and the real notes. High accuracy implies that the recommendation system is accurate.

The predicted values in our case correspond to the average of the ratings assigned by similar users for each active user.

To carry out our experiments, we first started by preparing the data corpus. The first corpus "A" represents the actual data entered by the users in our evaluation matrix. The second corpus "B" contains, however, predicted values.

Corpus "B" was generated from corpus "A" using the SQL query:

```
Create table B as Select * From A;
```

The matrix "A" therefore represents the learning corpus and the matrix "B" contains the test corpus. For each user of the table generated "B" we selected 30 of the 100 places randomly to form a pseudo-profile. This distribution was performed to keep a maximum number of real data (70% more precisely).

The value assigned to each of the selected places is the average value of the real values assigned by the 10 most similar users to the same place.

We have selected, in the Matrix "A" as well as in "B", the top-K of the places taken randomly. For each user, we calculated the intersection of the top-K of the two matrices. This last one was useful for the calculation of the precision (see Formula 2).

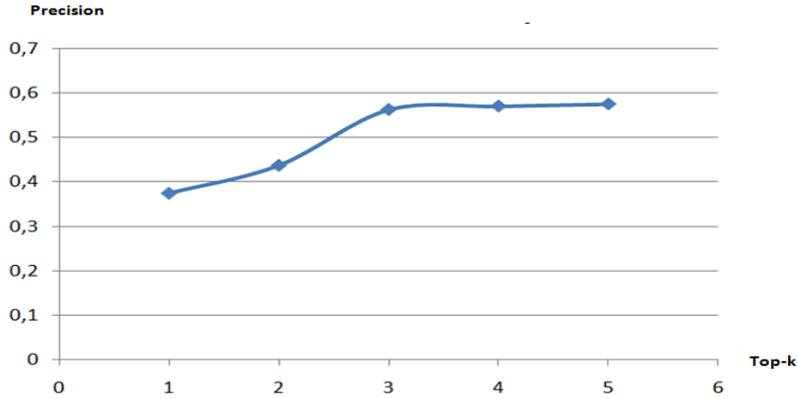
It should be noted that, the more a relevant item is far from the beginning of the list, the less useful it is for the user, because it has little chance to be retained.

$$\text{Precision\_profile} = \frac{\text{intersection (top-k A, top-k B)}}{k} \quad (2)$$

The precision of the system is equal to the sum of the precisions of all the profiles on the number of the latter (see Formula 3).

$$\text{Precision\_system} = \frac{\sum \text{precision\_profile}}{\text{number\_profiles}} \quad (3)$$

A graphical analysis to better understand the evolutions of precision has been adapted. After calculating the system precisions to recommend the top 1, top 2, top 3, top 4 and top 5, we plotted the precision curve (see Figure 6).



**Fig. 6.** Curve of system precisions using the top-K recommendations.

Figure 6 shows graphically the different precisions of the system according to the top-k. For the top 1 and 2 the precision is lower than the threshold 0.5. This precision has evolved from the top 3. We can see that the precision increases when the row k increases. This indicates an acceptable precision in our recommendation process.

## 4 Related works

Currently, there are two research trends to solve the problem of recommending points of interest in a city. One search line is conducted using GPS trajectory logs [6], [7], [8], [13]. GPS trajectory data typically includes a small number of users, but dense recordings [12], [9]. Many collaborative filtering algorithms, such as collective factorization of the matrix [7] tensor factorization [8] the memory-based collaborative localization model [13], etc., have been proposed to solve it and have considered the sites as the items of the traditional recommendation systems.

The other research trend concerns LBSN (location-based social networks) data, which are very sparse and large scale [10], [11], [14]. Currently, the geographical influence, for example, modeling the probability of recording at the distance of the total record history by distribution of the power law [11], Modeling behaviors multi-centric recording of users via multicenter Gaussians [14], etc., have been approached and merged with conventional collaborative filtering algorithms.

In this second axis, the authors of [16] proposed a model of region by subject (TRM), which is a unified probabilistic generative model, allowing to discover simultaneously the semantic, temporal and spatial patterns of the user registration activities and to model their joint effect on the decision-making, of the latter, for the selection of points of interest to be visited. In [15] the authors define formally the problem of the recommendations of successive personalized points of interest in the LBSN and analyze the spatiotemporal properties of two large LBSN data sets of the real world, Foursquare and Gowalla. They propose a new method of matrix factorization, namely FPMC-LR, to incorporate the two properties: the personalized Markov chain and localized region constraint. Thus, they significantly reduce the costs of calculating compared to previous methods.

Our solution for this problem is seen from a different angle. It is applied to the recommendation of places of exit for a user wishing to visit a city. This problem is handled in two different ways: collaborative filtering and content-based filtering.

## 5 Conclusion

The systems of recommendations have become, like the search engines, an indispensable tool. Their goal is to select the items most susceptible to interest a particular user. Our goal is to satisfy the latter by providing him with a personalized search that responds, at best, to his needs in terms of interesting exit places on the city of Oran using two types of filtering, cognitive and collaborative.

We have conceived a system that matches geo-localization with the recommendation. It is based on a linear distance for the content-based recommendation or the similarity of cosine for the collaborative recommendation.

In order to ensure the satisfaction of the user's expectations we calculated the performance of our system in terms of precision.

Our system is evaluated on the basis of an approach that we have proposed, which considers the top-K recommendations of the corpus of learning and the corpus of test. The result shows that our system ensures a good precision.

Having no real data, we had to build our own dataset and our own evaluation matrix by collecting information from students, close people or friends.

In the future, we envisage: (i) making recommendations considering the fluidity of road traffic (to avoid traffic jams, accidents, etc.); (ii) Use the content-based recommendation for places not for their categories; (iii) Optimize computational time by forming user clusters, and (iv) integrate other evaluation criteria in order to improve the quality of the recommendations and the predictive capacity of the system.

## References

- [1] M. J. Pazzani.: A framework for collaborative, content-based and demographic filtering. In: technical report, University of California, Irvine., 1999.
- [2] Herlocker, J. L.; Konstan, J. A.; Terven, L. G.; and Riedl, J. T. Evaluating Collaborative Recommender Systems. In: ACM Transaction on Information Systems, 2004.
- [3] A. T. Nguyen. COCoFil2: Un nouveau système de filtrage collaboratif basé sur le modèle des espaces de communautés. Thèse de Docteur. Université Joseph Fourier, Grenoble I. November, 2006.
- [4] Kr. Markov and Kr. Ivanova.: An ontology-content-based filtering method. In: Proceedings of the Fifth International Conference “Information Research and Applications”, Varna, Bulgaria. June 2007.
- [5] P. Shoval, V. Maidel, B. Shapira. An ontology-content-based filtering method, I.Tech-2007, Information Research and Applications, 2007.
- [6] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma.: Mining interesting locations and travel sequences from gps trajectories. In: WWW, pages 791–800, 2009.
- [7] Vincent Wenchen Zheng, Yu Zheng, Xing Xie, and Qiang Yang.: Collaborative location and activity recommendations with gps history data. In: WWW, pages 1029–1038, 2010.
- [8] V.W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang.: Collaborative filtering meets mobile recommendation: A user-centered approach. In: Proceedings of the 24rd AAAI Conference on Artificial Intelligence, 2010.
- [9] X. Cao, G. Cong, and C.S. Jensen.: Mining significant semantic locations from gps data. Proceedings of the VLDB Endowment, 3(1-2):1009–1020, 2010.
- [10] M. Ye, P. Yin, and W.C. Lee.: Location recommendation for location-based social networks. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, pages 458–461. In: ACM, 2010.
- [11] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik Lun Lee.: Exploiting geographical influence for collaborative point-of-interest recommendation. In: SIGIR, pages 325–334, 2011.
- [12] Yu Zheng and Xing Xie.: Learning travel recommendations from user generated gps traces. ACM TIST, 2(1):2, 2011.
- [13] KennethWai-Ting Leung, Dik Lun Lee, and Wang-Chien Lee. Clr: a collaborative location recommendation framework based on co-clustering. In SIGIR, pages 305–314, 2011.
- [14] C. Cheng, H. Yang, I. King, and M.R. Lyu.: Fused matrix factorization with geographical and social influence in location-based social networks. In Twenty-Sixth AAAI Conference on Artificial Intelligence, pages 17-23. In: AAAI Press, 2012.
- [15] Chen Cheng, Haiqin Yang, Michael R. Lyu, Irwin King.: Where You Like to Go Next: Successive Point-of-Interest Recommendation. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, pages 2605-2611. AAAI Press, 2013.
- [16] Hongzhi Yin, Bin Cui, Xiaofang Zhou, Weiqing Wang, Zi Huang and Shazia Sadiq.: Joint Modeling of User Check-in Behaviors for Real-time Point-of-Interest Recommendation. In: ACM Transactions on Information Systems (TOIS), 2016.
- [17] Shenglin Zhao, Irwin King, and Michael R. Lyu.: A Survey of Point-of-interest Recommendation in Location-based Social Networks. In: ArXiv, 2016.