

# Towards an Incremental Recommendation of POIs for Mobile Tourists without Profiles

**Nassim DENNOUNI**

Computer Science department, Hassiba Benbouali University of Chlef, Algeria  
E-mail: [n.dennouni@univ-chlef.dz](mailto:n.dennouni@univ-chlef.dz)

**Yvan PETER, Luigi LANCIERI**

NOCE team, CRISTAL Laboratory, Lille1 University, Lille, France  
E-mails: [yvan.peter@univ-lille1.fr](mailto:yvan.peter@univ-lille1.fr), [luigi.lancieri@univ-lille1.fr](mailto:luigi.lancieri@univ-lille1.fr)

**Zohra SLAMA**

EEDIS Laboratory, Djilali Liabes University of Sidi Bel Abbes, Algeria  
E-mail: [slama@univ-sba.dz](mailto:slama@univ-sba.dz)

Received: 17 November 2017; Accepted: 09 February 2018; Published: 08 October 2018

**Abstract**—Mobile tourism or m-tourism can assist and help tourists anywhere and anytime face the overload of information that may appear in their smartphones. Indeed, these mobile users find difficulties in the choice of points of interest (POIs) that may interest them during their discovery of a new environment (a city, a university campus ...). In order to reduce the number of POIs to visit, the recommendation systems (RS) represent a good solution to guide each tourist towards personalized paths close to his instantaneous location during his visit. In this article, we focus on (1) the detection of the spatiotemporal context of the tourist to filter the POIs and (2) the use of the previous notations of the places. These two criteria make it possible to integrate the evolutionary context of the visit in order to predict incrementally the most relevant transitions to be borrowed by the tourists without profile. These predictions are calculated using collaborative filtering algorithms that require the collection of traces of tourists to better refine the recommendation of POIs. In our software prototype, we have adapted the SLOPE ONE algorithm to our context of discovering the city of Chlef to avoid problems like data scarcity, cold start and scalability. In order to validate the use of this prototype, we conducted experiments by tourists in order to calculate indicators to assess the relevance of the recommendations provided by our system.

**Index Terms**—M-tourism, recommendation system, collaborative filtering, SLOPE ONE algorithm, POI rating prediction.

## I. INTRODUCTION

In recent years, the evolution of mobile equipment (smartphones, tablets ...) and localization technologies (4G, GPS ...) has invaded many areas of our daily lives

such as commerce, tourism, training, etc. This evolution helps and supports mobile users anywhere, anytime. Moreover, with the explosion of tourism on the WEB, it is becoming increasingly difficult to meet the growing needs of mobile users. This difficulty lies in the overload of information that may interest users and in the choice of places close to their instantaneous locations [1].

In this context, the applications to be developed must answer questions such as: (1) what are the places to visit? (2) Which services (restaurant, hotel ...) are closest? (3) Which are the sites best appreciated by tourists? Etc. To answer such questions, Recommendation Systems (RS) can effectively eliminate large information search spaces so that mobile users are directed to the places that best meet their expectations. Indeed, these systems make it possible to predict the personal interests of each mobile user thanks to collaborative filtering algorithms that are essentially based on navigation statistics (places visited by tourists, previous ratings of places ...). One of the features offered by this kind of system is the recommendation of points of interest (POI) [2]. These POIs are identified by their physical locations and present a specific interest to mobile users such as historic monuments, restaurants, hotels, stadiums, hospitals, banks, etc. In this context, the m-tourism scenario can be defined around the concept of POI, which allows associating the geographical location and the offers of products/services available. In addition to the concept of POIs, the definition of this type of mobile scenario also relies on the similarities of POIs to better target the products or services to be promoted based on the history generated by mobile users. However, this type of system suffers from many problems such as (1) the cold start problem where the first recommendations are not very significant and (2) the problem of scalability, which evaluates the performance of the recommendation algorithms in the face of the increasing number of users and POIs [3].

In this article, we focus on filtering POIs based on instant location and previous location ratings to support the evolving context of tourists through collaborative filtering algorithms. To better explain our approach, we organized the rest of this article in five sections: In the first section, we explain the use of RS in a context of mobile tourism. Then, we make in the second section an overview of the works that use the recommendation of the POIs during the visit of the tourists. Then, in the third section, we describe a conceptual view of our system to explain our contribution. Finally, before concluding, we present the architecture used for the implementation of our POIs recommendation system and the interfaces related to the test phase of our software prototype.

## II. MOBILE TOURISM AND RECOMMENDATION SYSTEMS

### A. Mobile tourism

According to Bourgeon-Renault, m-tourism concerns the use of smartphones in the field of tourism to have a renewed access to the tourist offer. This type of real-time access to information is likely to modify the behavior of tourists before, during and/or after their travels [4]. On the other hand, Bonneau explains that m-tourism represents a new form of access to tourist information after the advent of e-tourism [5]. In [6], the authors define m-tourism as a means that allows visitors to access real-time information adapted to the place visited using different mobile devices such as smartphones, tablets...

In our opinion, the definition of m-tourism is always evolving, as it has to adapt each time to new physical, technological and tourist contexts that are closely linked to the various fields of application. As part of this article, we define m-tourism as a tool that integrates criteria of time (day/night, winter/summer ...) and space (mode of travel, instantaneous position) in order to offer the tourist the ability to access relevant information at the right time and in the right place. This information can be represented through a POI (Point Of Interest). This term refers to a potentially interesting place identified by its geographical coordinates. This place can be a tourist attraction, a hotel, a restaurant, a pharmacy, a medical center, a shop, a petrol station, a school, etc. [7].

### B. Recommender System

In practice, most recommendation systems (RS) offer lists of resources to users. Such resources can correspond to different types of data such as movies [8], music [9], books [10], restaurants [11], news [12], jokes [13], scientific articles [14], web pages [15], etc. In the field of m-tourism, the RSs can be used to suggest POIs that allow visitors to discover a new city, a new university campus [16,17] ... Indeed, these systems are able to provide personalized recommendations allowing guiding the user to interesting or useful resources within an important data space [18]. Then he receives the new system recommendations in the form of a list of POIs he has not visited.

According to Singh and al. [19], RSs evolve with the Web. Indeed, since the development of Web 1.0 RSs, the

recommendation of articles in the e-commerce application framework has become the main area of concern for researchers who have studied many approaches to recommendations. Then, with the advent of Web 2.0, RSs begin to use social relationships between users and the tagging of contextual information to provide diverse and accurate recommendations. Web 3.0 RSs have emerged through the increasing use of mobile devices and are using technologies such as the Internet of Things and Location Based Systems (LBSs). The m-tourism applications use smartphones that integrate GPS as a means of localization therefore they require the implementation of Web 3.0 RS to allow the recommendation of POIs for mobile tourists. In [20,21], the authors classify RSs into three categories: (1) Content-based RSs, (2) Collaborative filtering RSs, and (3) hybrid RSs. In some applications of m-tourism, each user has a profile describing it through their interests. To recommend POIs relevant to his visit, content-based RSs try to match POI categories with user preferences and interests. For this reason, the system calculates the proximity of each POI to the user's profile using a metric that estimates the distance between the POI to be recommended and the interests of the tourist [22]. However, during the discovery of a new city, mobile tourists are often without prior profiles and therefore this type of RS cannot assist them in the choice of POIs to visit [23].

To solve this problem, memory-based collaborative filtering algorithms seem to be a good solution because they use the old notations of POIs already visited by tourists regardless of their profiles [24]. This type of RS calculates the similarities between the users (or between the POIs) to be able to make the predictions of the future notations of POI to be recommended.

In this article, we are interested in the discovery of several categories of POIs (hotels, theme parks, historical monuments...) by tourists without prior profiles. For this reason, our choice fell on the technique of collaborative filtering based on the neighborhood between the users to make the recommendation of the POIs. These emplacements are repeated by the GPS (available in most smartphones) to facilitate their localization by the tourists.

## III. RELATED WORKS

In this article, the works of interest to us concern the recommendation of POIs in the form of a list ordered according to the interest of the mobile tourist.

These interests are indicated in the user's profile or retrieved from the visited POIs. In this state of the art, we distinguish between two types of RS approaches: (1) RSs that rely only on the history generated by its users and (2) RSs that use data from networks social.

As part of the first approach, Barranco et al. [25] propose a RS based on LBS to recommend POIs according to the current location of tourists and their driving speeds. For example, if a user moves along a highway and wants to stop at a restaurant then this system uses an AOI (area of interest) centered on its current

location to suggest POIs along its route.

Noguera et al. [26] present another type of mobile RS based on a hybrid recommendation engine and a GIS<sup>1</sup> architecture. This system uses a 3D map interface to make real-time POI recommendations based on user preferences.

To face the eventualities that arise during the discovery of a new city, Linas et al. [27] propose a new approach in which tourists are asked to judge whether a contextual factor really influences their pathways (for example, whether accompanying children influences the decision to visit a POI). This approach of RS uses a predictive model (based on matrix factorization) for the recommendation of tourist POIs according to the factors of influence chosen dynamically by the visitors.

In this context, the RS designed in [28] can help visitors to identify the areas of tourist attraction that most closely match their personal profiles. This system calculates the similarity between existing queries and new queries provided by visitors in order to provide acceptable recommendations.

In the same context, the RS described in [29] helps students to discover their university campuses through the recommendation of POIs that may interest them in their academic lives. This system recommends POIs based on: (1) the learner's position, (2) the instructor's pedagogy, (3) the learners' score, (4) the representativeness of the paths, and (5) the collaboration during the visit.

To take advantage of Linked Open Data (LOD), Fridi and Benslimane have proposed the addition of this type of semantic information to improve the efficiency of traditional collaborative filtering in conventional RSs [30].

On the other hand, to better consider characteristics such as hobbies, topics of interest ... Sharma et al. [31] propose an RS named Research Work Area Recommender System (RWARS) that suggests POIs based on the similarity calculation between users using the cosine similarity approach of collaborative filtering.

However, the implementation of this kind of algorithm poses problems when scaling. For this reason, Walunj et al. [32] use Apache mahout to offer flexibility in the use of pre-existing algorithms and allow a good resolution of the problem of scalability by ensuring acceptable recommendations.

The second approach is based on networks such as (1) Foursquare<sup>2</sup>: a social network for recording places and (2) Flickr<sup>3</sup>: another social network for sharing geo-localized photos. These two types of Location-Based Social Networks (LBSNs) generate useful data for dynamic POI recommendation applications in geographical graphs [33].

However, this kind of application must use light methods (compatible with the computing capabilities of mobile devices) and effective against the large masses of data generated by the use of these social networks by tourists.

To achieve these objectives, Norma Saiph et al. [34]

proposes an RS named " I'm feeling LoCo " (I'm feeling Location and Context) that exploits the profile of a person in the Foursquare social network and the physical constraints like the location of the user and the mode of transport (to foot, bike or car) detected automatically (based on measurements taken by the accelerometer sensor of a smartphone). The tourist only has to explicitly define the type of places likely to interest him and the RS proposes POIs to visit with the help of a decision tree.

Recently, [35] [36] uses recommendation algorithms that exploit combinations of factors such as geographic influence, changing user tastes, etc. but they do not take into account POIs temporal availability.

For this reason, Sang et al. [37] have addressed the problem of recommending POIs sequences taking into account the time constraints associated with POIs such as the opening and closing time of tourist sites.

On the other hand, Gavalas and kenteris [38] implement a Mobile Tourism Recommendation System (MTRS) that provides several services to mobile users, taking into account contextual information such as the location of the user, the current time, weather conditions and POIs already visited by the user. This system has enabled the creation of a mobile travel guide platform named MyMytilene that allows users to explicitly select the tourism content to be included in a personalized mobile application.

In [39], the recommendation of spatio-temporal activity sequences applies to several geo-localized and time-limited events such as participation in conferences, attendance at festivals and exhibition visits. This approach jointly exploits the users' interests, the set of spatial and temporal factors and the user's habits in order to build a personalized itinerary of POIs.

The works presented in this state of the art recommends sequences of POIs according to two technical types: (1) the construction of an optimal POIs path according to constraints coming from the field of operational research [37] and (2) calculating the probability of transition from one POI to another [40]. However, these techniques do not take into account neither the evolution of the number of POIs during the visit nor the users without profiles that feed the browsing history.

#### IV. PROPOSED APPROACH

In this article, we propose a new approach that makes it possible to incrementally recommend POIs for tourists without profiles during their visits.

For this reason, we adapt the SLOPE ONE algorithm [41] to take into account user preferences and spatiotemporal constraints related to the geographic distribution and availability of POIs.

Our approach is composed of 3 main phases: (1) identification of the actors who will interact with our RS thanks to a UML diagram of the use cases (2) use of a UML class diagram to describe the tables which will contain the history of our RS and (3) implementation of an algorithm for the recommendation of POIs that takes

<sup>1</sup> Geographic Information System

<sup>2</sup> <https://fr.foursquare.com>

<sup>3</sup> <https://www.flickr.com/>

into account the instantaneous location and evolution of interactions between users.

A. Identification of the actors of our RS

The actor represents an external element that interacts with our RS. In our case, we have identified three types of actors: (1) the anonymous user (without profile) who does not wish to register to be able to use our RS, (2) the identified user who does not declare his profile and (3) the user with profile. The anonymous user can only see POIs that may appear near his instantaneous location. These POIs can be classified in a list according to the number of views or ratings assigned by mobile users. After registration, the user identified by the information he has declared may rate and comment on existing POIs in the database. The user with profile can suggest other POIs but it is the system administrator who can introduce these new POIs into the recommendation process. In Figure 1, the UML diagram of use cases highlights the functional relationships between the actors and our RS.

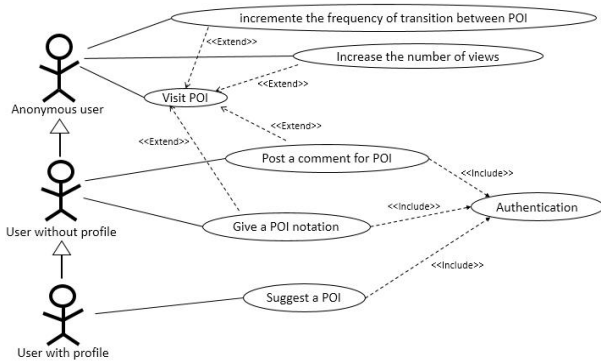


Fig.1. UML diagram of the use cases of our RS

B. Description of tables associated to our RS

Our RS makes it possible to suggest POIs to the three types of users already described in our use case diagram. Then these users may choose to follow or not these recommendations. Decisions made by these will be saved in the database described in the UML class diagram below (see Figure 2)

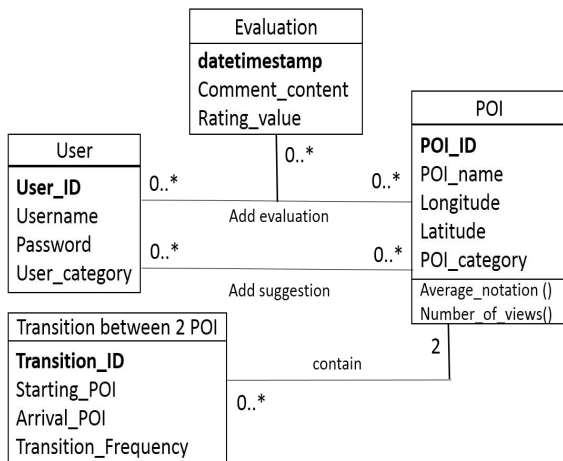


Fig.2. UML class diagram of our RS

C. Algorithm used by our RS

Our RS uses an adaptation of the SLOPE ONE algorithm to incrementally recommend POIs during the tour of tourists without profiles.

This recommendation is based on two steps: (1) the calculation of the similarities between the set of POIs and the POI visited by the tourist and (2) the prediction of the future notations that this same tourist will attribute to unvisited POIs.

In what follows, the figure 3 shows an example of a POI recommendation based on the similarity between two POIs.



Fig.3. Principle of recommendation of the POIs

Our RS calculates for each pair (POI<sub>i</sub>, POI<sub>j</sub>), the average of the differences of appreciations for all the users who evaluated these two POIs. In what follows, Diff (POI<sub>i</sub>, POI<sub>j</sub>) denotes this average and we can calculate it using formula 1:

$$Diff(POI_i, POI_j) = \frac{\sum_{u \in E_{ij}} (U_{POI_j} - U_{POI_i})}{Nb(E_{ij})} \quad (1)$$

Where

- U<sub>POI<sub>i</sub></sub> is the notation assigned to the POI<sub>i</sub> by the user U.
- E<sub>ij</sub> represents the set of the users having evaluated the pair (POI<sub>i</sub>, POI<sub>j</sub>).
- Nb (E<sub>ij</sub>) is the number of element of the set E<sub>ij</sub>.

To save the results of calculating similarities between POIs, our RS uses an intermediate table that is not described in the previous class diagram as indicated in the following SQL code:

```
CREATE TABLE dev
(POI_ID1 int (11) NOT NULL default ' 0',
POI_ID2 in t (11) NOT NULL default ' 0',
Count int (11) NOT NULL default ' 0',
Sum int (11) NOT NULL default ' 0 ',
PRIMARY KEY (POI_ID1, POI_ID2));
```

To explain the functioning of our RS, we use a matrix which contains the notation of 7 POIs by 10 users (see table 1).

Table 1. The vote of 10 users on 7 POIs

	POI <sub>1</sub>	POI <sub>2</sub>	POI <sub>3</sub>	POI <sub>4</sub>	POI <sub>5</sub>	POI <sub>6</sub>	POI <sub>7</sub>
User01	1	3	3	5	?	?	1
User02	2	3	?	4	5	4	?
User03	?	1	4	?	?	3	2
User04	1	3	?	?	5	2	4
User05	5	4	?	4	1	2	3
User06	3	?	?	3	4	4	3
User07	4	3	2	?	?	?	5
User08	?	?	5	1	?	1	?
User09	?	?	1	?	2	5	?
User10	2	5	3	5	3	3	?

Assuming that a user has chosen POI<sub>5</sub> as their starting point, our RS initially calculates the average of the differences associated with the ratings of each POI pair involving that POI. In what follows, we calculate from Table 2, the Diff (POI<sub>5</sub>, POI<sub>1</sub>) which represents the similarity value between the POI<sub>5</sub> and POI<sub>1</sub>.

Table 2. The Elements Involved in the Calculation of DIFF (POI<sub>5</sub>, POI<sub>1</sub>)

	POI <sub>1</sub>	POI <sub>5</sub>
User01	1	?
User02	2	5
User03	?	?
User04	1	5
User05	5	1
User06	3	4
User07	4	?
User08	?	?
User09	?	2
User10	2	3

$$\text{Diff}(\text{POI}_5, \text{POI}_1) = [(5 - 2) + (5 - 1) + (1 - 5) + (4 - 3) + (3 - 2)] / 5 = 1$$

From Table 1, our RS calculates the other similarities between each pair of POIs involving POI<sub>5</sub> as a starting point:

$$\text{Diff}(\text{POI}_5, \text{POI}_2) = [(5 - 3) + (5 - 3) + (1 - 4) + (3 - 5)] / 4$$

$$\text{Diff}(\text{POI}_5, \text{POI}_2) = -1/4$$

$$\text{Diff}(\text{POI}_5, \text{POI}_3) = [(2 - 1) + (3 - 3)] / 2 = 1/2$$

$$\text{Diff}(\text{POI}_5, \text{POI}_4) = [(5 - 4) + (1 - 3) + (4 - 3)] / 3 = 0$$

$$\text{Diff}(\text{POI}_5, \text{POI}_6) = [(5 - 4) + (5 - 2) + (1 - 2) + (4 - 4) + (2 - 5) + (3 - 3)] / 6 = 0$$

$$\text{Diff}(\text{POI}_5, \text{POI}_7) = [(5 - 4) + (1 - 4) + (4 - 3) + (3 - 5)] / 4$$

$$\text{Diff}(\text{POI}_5, \text{POI}_7) = -3/4$$

Note that the denominators "5", "4", "2", "3", "6" and "4" represent the numbers of pairs involved in calculating each difference between two POIs. Similarly, our RS calculates the matrix of counters of POIs pairs and the matrix of differences between POIs as shown in table 3 and table 4.

Table 3. Matrix of differences between POIs

	POI <sub>1</sub>	POI <sub>2</sub>	POI <sub>3</sub>	POI <sub>4</sub>	POI <sub>5</sub>	POI <sub>6</sub>	POI <sub>7</sub>
POI <sub>1</sub>	-	-1	-1/3	-8/5	-1	-2/5	-2/5
POI <sub>2</sub>	1	-	0	-3/4	1/4	2/5	-1/5
POI <sub>3</sub>	1/3	0	-	0	-1/2	1/4	1/3
POI <sub>4</sub>	8/5	3/4	0	-	3/4	3/5	5/3
POI <sub>5</sub>	1	-1/4	1/2	0	-	0	-3/4
POI <sub>6</sub>	2/5	-2/5	-1/4	-3/5	0	-	-1/4
POI <sub>7</sub>	2/5	1/5	-1/3	-5/3	0	1/4	-

Table 4. Matrix of counters of POIs pairs

	POI <sub>1</sub>	POI <sub>2</sub>	POI <sub>3</sub>	POI <sub>4</sub>	POI <sub>5</sub>	POI <sub>6</sub>	POI <sub>7</sub>
POI <sub>1</sub>	-	5	3	5	5	5	5
POI <sub>2</sub>	5	-	4	4	4	5	5
POI <sub>3</sub>	3	4	-	3	2	4	3
POI <sub>4</sub>	5	4	3	-	4	5	5
POI <sub>5</sub>	5	4	2	4	-	6	3
POI <sub>6</sub>	5	5	4	5	6	-	4
POI <sub>7</sub>	5	5	3	3	3	4	-

For example, if the user has chosen the POI<sub>5</sub>, our RS can classify the POIs according to the similarity criterion (difference absolute value in table 3) from the smallest to the largest as follows:

$$|\text{Diff}(\text{POI}_5, \text{POI}_6)| \leq |\text{Diff}(\text{POI}_5, \text{POI}_4)| \leq |\text{Diff}(\text{POI}_5, \text{POI}_2)| \leq |\text{Diff}(\text{POI}_5, \text{POI}_3)| \leq |\text{Diff}(\text{POI}_5, \text{POI}_7)| \leq |\text{Diff}(\text{POI}_5, \text{POI}_1)|$$

Then, our RS indicate to the user that POI<sub>6</sub> or POI<sub>4</sub> are the best POI to visit and class the rest of POIs by their order of importance.

$$(\text{POI}_6 \text{ or } \text{POI}_4) \rightarrow \text{POI}_2 \rightarrow \text{POI}_3 \rightarrow \text{POI}_7 \rightarrow \text{POI}_1$$

After determining the matrix of similarities between the POIs, our RS calculates the prediction of the notes that will be assigned by the users to the unvisited POIs using the formula 2:

$$PN(\text{User}_x, \text{POI}_y) = \frac{\sum_{i \in E_x} (\text{Diff}(\text{POI}_y, \text{POI}_i) + \text{Not}(\text{User}_x, \text{POI}_i)) * Nb(E_{y_i})}{\sum_{i \in E_x} Nb(E_{y_i})} \quad (2)$$

Where

- $PN(\text{User}_x, \text{POI}_y)$  is the prediction of the note that the user<sub>x</sub> will assign to the POI<sub>y</sub> attribute.
- $\text{Diff}(\text{POI}_y, \text{POI}_i)$  represents the similarity between the POI<sub>y</sub> and the POI<sub>i</sub>.
- $\text{Not}(\text{User}_x, \text{POI}_i)$  is the notation assigned by the user "User<sub>x</sub>" to the POI "POI<sub>i</sub>".
- $E_x$  represents the set of POIs evaluated by the

“User<sub>x</sub>”.

- E<sub>yi</sub> represents the set of pairs (POI<sub>y</sub>, POI<sub>i</sub>) such that i is an element of the set E<sub>x</sub>.
- Nb (E<sub>yi</sub>) is the number of elements of the set E<sub>yi</sub>.

For example, to calculate the prediction of the score that will be assigned by User02 to POI<sub>3</sub>, we use Formula 2 and the data from table 5 and table 6 as follows:

Table 5. User02's Assigned Ratings

	POI <sub>1</sub>	POI <sub>2</sub>	POI <sub>3</sub>	POI <sub>4</sub>	POI <sub>5</sub>	POI <sub>6</sub>	POI <sub>7</sub>
User02	2	3	?	4	5	4	-

Table 6. Calculation of POI<sub>3</sub>'s similarities (differences and counters)

	POI <sub>1</sub>	POI <sub>2</sub>	POI <sub>3</sub>	POI <sub>4</sub>	POI <sub>5</sub>	POI <sub>6</sub>	POI <sub>7</sub>
POI <sub>3</sub>	1/3 3	0 4	-	0 3	-1/2 2	1/4 4	1/3 3

$$PN(\text{User02}, \text{POI}_3) = [(1/3+2)*3 + (0+3)*4 + (0+4)*3 + (-1/2+5)*2 + (1/4+4)*4] / (3+4+3+2+4) = 3, 56.$$

In the same way, our algorithm calculates the different predictions of the notes that the users will give to the unvisited POIs as it is indicated in the table 7:

Table 7. Prediction Of Notes Attributed To Non-Visited POIs.

	POI <sub>1</sub>	POI <sub>2</sub>	POI <sub>3</sub>	POI <sub>4</sub>	POI <sub>5</sub>	POI <sub>6</sub>	POI <sub>7</sub>
User01	1	3	3	5	2.66	2.43	1
User02	2	3	3.56	4	5	4	3.35
User03	1.77	1	4	3.2	2.4	3	2
User04	1	3	2.93	3.85	5	2	4
User05	5	4	3.36	4	1	2	3
User06	3	3.56	3.53	3	4	4	3
User07	4	3	2	4.6	4	3.38	5
User08	1.07	2.15	5	1	1.5	1	1.7
User09	2.30	3.07	1	3.5	2	5	2.9
User10	2	5	3	5	3	3	3.39

To facilitate the previous calculations, our RS uses three functions: the *Filter\_of\_POIs* function, the *Pair\_Difference\_Calculation* function and the *Prediction\_notations* function. Each function corresponds to a main step in the recommendation process of our system as shown in the figure 4:

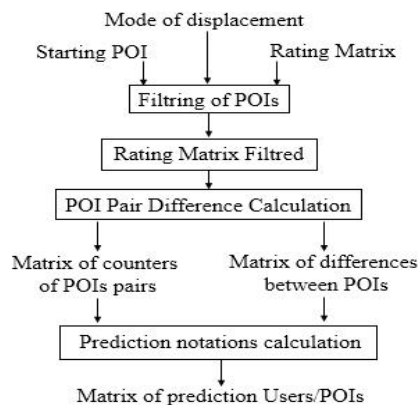


Fig.4. The main steps of our recommendation process

*Filter\_of\_POIs* function makes it possible to filter the evaluations of the POIs according to (1) the GPS location of the user, (2) the mode of displacement detected according to the speed calculated by the accelerometer of the smartphone and (3) the choice POI start of the visit.

**Function Filter\_of\_POIs**

**Input**

T1 : Users / POIs rating Matrix  
 POI<sub>d</sub> : starting POI selected by the user  
 R: the maximum distance that the user can travel

```

1: For all POIi ∈ T1 Do
2:   If distance (POIi, POId) < R Then
3:     T2 (i, d) ← T1 (i, d)
4:   End If
5: End Do
    
```

**Output**

T2 : Users / POIs rating Matrix

*Pair\_Difference\_Calculation* function calculates the average differences of user ratings for each pair of POIs.

**Function Pair\_Difference\_Calculation**

**Input**

T2 : Users / POIs rating Matrix

```

1: For all POIi ∈ T2 Do
2:   For all POIj <> POIi Do
3:     Som ← 0
4:     Count ← 0
5:     For all users evaluating POI i and POI j Do
6:       Som ← Som + (Rating POIi – Rating POIj)
7:       Count ← Count + 1
8:     End Do
9:     Moy_Diff (i,j) ← Som / Count ;
10:    NB_Pair_Diff (i,j) ← Count;
11:   End Do
12: End Do
    
```

**Output**

Moy\_Diff : Matrix of differences between POIs  
 NB\_Pair\_Diff: Matrix of counters of POIs pairs

*Prediction\_notations* function uses the result of these two functions for calculating the predictions of POIs not evaluated by users.

**Function Prediction\_notations**

**Input**

T2 : Users / POIs rating Matrix  
 Moy\_Diff : Matrix of differences between POIs  
 NB\_Pair\_Diff: Matrix of counters of POIs pairs

```

1: For all POI where user U did not assign a rating Do
2:   SC ← 0 ; | Sum of the pair counters |
3:   SN ← 0 ; | Sum of ratings and differences |
4:   For all POIj <> POIi where user U assigned a rating
5:     SN ← SN + [Notation (U, j) + Moy_Diff (i,j)]
6:     SC ← SC + NB_Pair_Diff (i,j)
7:   End Do
8:   Prediction (U, POIi) = SN / SC ;
9: End Do
    
```

**Output**

Prediction: Matrix of prediction Users/POIs



Our RS allows us to suggest POIs from the Database described in our class diagram. These POIs will be recommended to users already described in our use case diagram. As shown in Figure 5, this system is based on the following five steps:



Fig.5. Description of how our RS works

(a) Saving traces related to the browsing history of each user (notations by visited POI, transitions between POIs, user position, starting point selected).

(b) Calculation of POI/POI similarities and ratings predictions by users.

(c) Update of matrices and intermediate tables.

(d) After classification of predictions, the web service provides the list of POIs to recommend for the user.

(e) Save the POI chosen by the user.

Finally, our RS allows exploiting information collected during the visit of tourists such as: (1) the mode of travel (on foot, by bike, by car), (2) the starting POI, (3) the Recommended POIs, (4) the chosen POIs and (5) the ratings assigned to the different POIs. These data are used to recommend POIs that assist new tourists in their discoveries of a new environment.

## V. IMPLEMENTATION OF OUR SOFTWARE PROTOTYPE

Our software prototype uses the GPS position, the speed of movement provided by the Smartphone and the internet connection to access via a web service to our database server and to Google Maps.

For this reason, we have chosen an architecture composed of three parts: the first part concerns the mobile user interface, the second part contains the web services used for the invocation of our RS and the last part is composed of the DBMS MYSQL and Google Maps as shown in Figure 5:

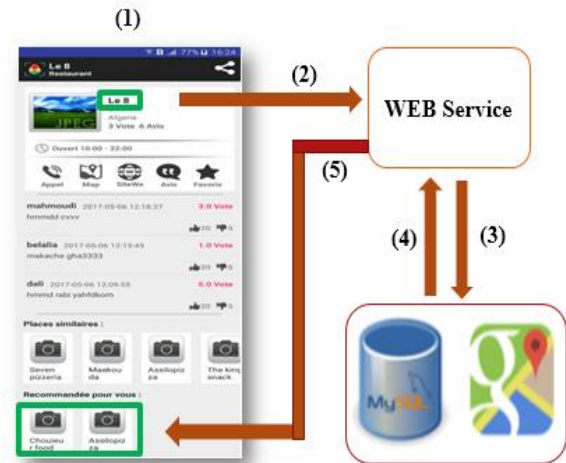


Fig.6. Architecture of our software prototype

To explain this architecture, we detail the progress of the recommendation process using the following steps:

(1) This is an interface allows the user of our mobile application to: (a) declare his profile or not, (b) choose an existing POI, (c) suggest a new POI, rate a POI, post a comment on a POI, ... (The figure shows that the user has chosen the POI named "Le 8")

(2) The web services can be called via the `URLConnection` object as follows:

```
URL url =new URL
("https://test.com/getPOISimilaire.php?"
+ "id_poi="+ id_poi);
HttpURLConnection connection =
(HttpURLConnection) url.openConnection();
```

(3) Connect our mobile application to the Database Server via our Web Service.

```
Include'connxionBDD.php';
$sql2 =
"SELECT d.Id_Poi1, d.Id_Poi2, ABS
(d.sum/d.count) AS moyenne
FROM diff_Entre_Poi d
WHERE (Id_Poi1=$id_poi
AND Id_Poi1!= Id_Poi2)
ORDER BY moyenne
ASC LIMIT $n";
```

(4) The database returns the data requested by the WEB service in the JSON format.

```
Echo json_encode (array ("POISimilaire"
=>$typearray), JSON_UNESCAPED_UNICODE);
```

(5) This response in JSON format will be transformed by our system to allow its reading through the client interface.

```

InputStreamReader stream =new
InputStreamReader
(connection.getInputStream ());
BufferedReader bufferedReader = new
BufferedReader(stream);
StringBuilder sb = new
StringBuilder();String line = null;
while ((line=bufferedReader.readLine()
) != null)
{sb.append(line + "\n");}
stream.close();
final String result =sb.toString();
    
```

	POI <sub>1</sub> The king snack	POI <sub>2</sub> Seven pizzeria	POI <sub>3</sub> Le 8	POI <sub>4</sub> Maakouda Aroudj	POI <sub>5</sub> Chouieur food	POI <sub>6</sub> Assilo pizza	POI <sub>7</sub> Souflé
POI <sub>1</sub>	-	-1; 5	-1/3; 3	-8/5; 5	-1; 5	-2/5; 5	-2/5; 5
POI <sub>2</sub>	1; 5	-	0; 4	-3/4; 4	1/4; 4	2/5; 5	-1/5; 5
POI <sub>3</sub>	1/3; 3	0; 4	-	0; 3	-1/2; 2	1/4; 4	1/3; 3
POI <sub>4</sub>	8/5; 5	3/4; 4	0; 3	-	3/4; 4	3/5; 5	5/3; 5
POI <sub>5</sub>	1; 5	-1/4; 4	1/2; 2	0; 4	-	0; 6	-3/4; 3
POI <sub>6</sub>	2/5; 5	-2/5; 5	-1/4; 4	-3/5; 5	0; 6	-	-1/4; 4
POI <sub>7</sub>	2/5; 5	1/5; 5	-1/3; 3	-5/3; 3	0; 3	1/4; 4	-



POIs→ Users ↓	The king snack	Seven pizzeria	Le 8	Maakouda Aroudj	Chouieur food	Assilo pizza	Souflé
Maamar	1	3	3	5	2.66	2.43	1
Youcef	2	3	3.56	4	5	4	3.35
Ayoub	1.77	1	4	3.2	2.4	3	2
Ridha	1	3	2.93	3.85	5	2	4
Karim	5	4	3.36	4	1	2	3
Ahlam	3	3.56	3.53	3	4	4	3
Souhila	4	3	2	4.6	4	3.38	5
Hafessa	1.07	2.15	5	1	1.5	1	1.7
Khadija	2.30	3.07	1	3.5	2	5	2.9
Maryam	2	5	3	5	3	3	3.39

Fig.7. Correspondence between recommendation calculation results and POIs obtained through our RS

After implementing our prototype, we introduced the previous example (see Table 1, Table 3, Table 4 and Table 7) to see if the recommendations provided by our prototype (see Figure 6) match those obtained by the

calculation in using formulas 1 and 2.

In the figure 6, the user "Maamar" chose the POI<sub>3</sub> named "Le 8". For this reason, we calculate using formula 1 the similarity values between the POI<sub>3</sub> and the other POIs as shown in the green rectangle of the table. These values can be ranked in ascending order as follows:

Seven pizza → Maakouda Aroudj → Assilo pizza → The king Snac → Souflé → Chouieur Food

On the other hand, the interface of our prototype shows that our RS has obtained the same ranking but it displays only 4 POIs because of the reduced size of the screen of the smartphone.

In addition, our RS can also predict through formula 2 the ratings that the user "Maamar" will be assigned to POIs "Chouieur food" and "Assilo pizza" as shown in the purple rectangle of the table. This result is confirmed by the two elements displayed in the purple rectangle of the interface of our prototype.

This example made it possible to test the smooth running of the algorithms presented in the fourth section because it uses on the one hand the formula 1 for the calculation of the similarities and on the other hand the formula 2 for the estimation of the predictions of notation.

## VI. EXPERIMENTATION OF OUR SOFTWARE PROTOTYPE

Although there are test sets for the evaluation of recommendation algorithms, there are, to our knowledge, no suitable datasets for evaluating the recommendation of POIs in the context discovery of a new city.

LSBNs datasets provide a set of POIs and check-ins performed by users with social profiles but these profiles differ from the profiles of tourists wanting to discover a new city. In this context, we are building a new dataset, composed of several POIs (users with profile can add new POIs) organized according to the categories that may interest the tourists of the city of Chlef.

Our software prototype uses this dataset to recommend places to visit to mobile tourists while on the move using the SLOPE ONE algorithm [41] as the engine of our RS. Nevertheless, in order to have acceptable recommendations, we need to collect the history of tourist guide visits that may suggest visiting certain POIs instead of others.

For this reason, we invited tourism experts (guides, former inhabitants of the city ...) and new users to discover the city of Chlef using our software prototype. This experiment enriches users' browsing history and assesses the acceptance rate of the recommendations provided by our RS.

To calculate this rate, we use a boolean variable named "choisir\_Poi\_Recom" which is 1 if the user follows one of the POIs recommended by our system and is 0 otherwise.

For example in Figure 7, the variable "choisir\_Poi\_Recom" is equal to 0 because the point chosen by the user is different from that recommended by the RS.



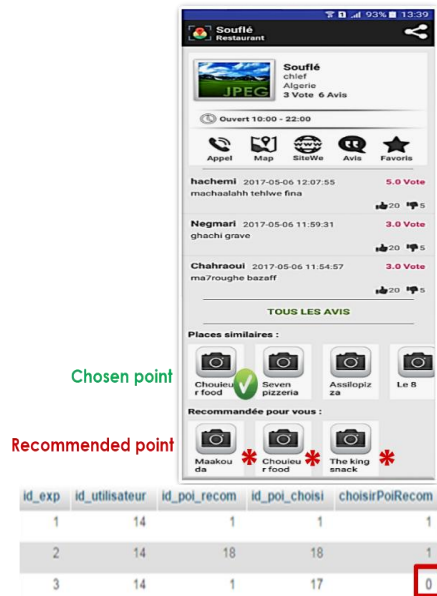


Fig.8. The interface allowing the choice of a POI.

During the experiment phase of our software prototype, each user chooses a starting point according to its instantaneous location. Then, during each transition, the user can choose a POI recommended by the RS or just a similar POI. As a result of our experiment, we decided to save for each user his identifier, the identifier of the recommended POI by our RS and the identifier of the POI chosen during the visit.

From these data, we can calculate the acceptance rate of the recommendations provided by our system. Indeed, if the user chooses the first POI of the list it means that he has followed the recommendation of our system so we update a Boolean variable "choisir\_Poi\_Recom" to 1. Otherwise (if the user does not follow the recommendation of our system), this Boolean variable will be 0. Then our system will calculate the sum of the choices of each user (the sum of the values of the Boolean variable "choisir\_Poi\_Recom") to divide it on the total number visited POIs. This calculation makes it possible to estimate the rate of acceptance of the recommendations for each user by using the table named "experimentation".

This table contains the following attributes:

- id\_ex: represents a primary key for the table.
- Id\_utilisateur: identifies the user and allows having his name.
- Id\_poi\_recom: contains the POI that appears first in the list of top Recommendation.
- Id\_poi\_choisi: contains the POI really chosen by user.
- choisirPoiRecom: this variable = 1 if (Id\_poi\_recom = Id\_poi\_choisi) or = 0 if (Id\_poi\_recom ≠ Id\_poi\_choisi).

To calculate the acceptance rate of the user "14", it is necessary to average the sum of the values of the variable "choosePoiRecom" as indicated in the following SQL code:

```
$query="SELECT SUM (choisirPoiRecom)/
COUNT (choisirPoiRecom) as s
FROM experimentation
WHERE e.id_utilisateur=14;
```

After the end of the participation of the testers of our software prototype, our system will calculate the average of the acceptance rates of all the users to estimate the effectiveness of our algorithm of recommendation as shows the following SQL code

```
$query="SELECT AVG (choisirPoiRecom) AS
moyenne FROM experimentation»;
```

## VII. CONCLUSIONS AND FUTURE WORK

This article presents an approach to assist mobile tourists without profiles based on their instant locations and POIs already visited.

This approach integrates the spatial-temporal profile (location, arrival time, means of transport ...) of tourists in the POI recommendation process.

This profile is detected automatically using the Smartphone (GPS, accelerometer). Then, the SLOPE ONE algorithm calculates similarities between users based on the existing POIs notation to predict future ratings of unvisited POIs.

In general, our RS is better suited to the context of discovering a new environment than the methods of recommending POIs cited in the state of the art of this article, but some points remain problematic:

- The scarcity of data:** If a tourist notes a single POI, our algorithm can make predictions while other algorithms require the notation of 2 POIs at least. Nevertheless, as the database of evaluations increases, the recommendation of our RS becomes more precise.
- The cold start:** this problem concerns on the one hand, the new users without profiles that our RS detects thanks to their locations and on the other hand, the new POIs unknown by our RS (without notation). To solve this problem, our RS may recommend to a tourist without a profile places similar to the POIs he has already visited. However, calculating these similarities may involve adding new POIs and accepting ratings of new tourists without profiles.
- Scalability:** despite the increasing number of POIs and users of our RS, the computation time associated with the recommendations provided to the current user does not exceed a fraction of a second. For the future, the problem of computational complexity will depend on the number of neighbors to be determined for the calculation of similarities. This number can be reduced by minimizing the filter radius of the POIs thanks to the position and speed of the tourists.

For future work, we plan to compare the POIs recommendation based solely on our prototype visitor history with that based on the LBSN user history. This comparison will test the behavior of our RS against these two situations through the calculation of new indicators of acceptance of the recommendations provided.

#### ACKNOWLEDGMENT

This work was supported in part by the NOCE team of the CRISTAL laboratory and by several master projects at the computer science department of the University of Chlef.

#### REFERENCES

- [1] N. Dennouni, Y. Peter, L. Lancieri and Z. Slama, "State of the art on orchestration techniques of mobile learning activities," Proceedings of the 4<sup>th</sup> International Symposium on Concepts and Tools for Knowledge Management (ISKO-Maghreb), November 9-11, 2014, Algiers, Algeria.
- [2] N. Dennouni, Y. Peter, L. Lancieri and Z. Slama, "To a Geographical Orchestration of Mobile Learning Activities," International Journal of Interactive Mobile Technologies (IJIM), Vol.8, No.2, pp. 35-41, 2014.
- [3] N. Dennouni, Z. Slama, Y. Peter and L. Lancieri, "Recommendation Techniques in Mobile Learning Context: A Review," International Journal of Modern Education and Computer Science (IJMECS), Vol.9, No.10, pp. 37-46, 2017. DOI: 10.5815/ijmeecs.2017.10.05
- [4] B. Renault, «Gratuité et valeur attachée à l'objet de consommation culturelle : le cas des musées et des monuments», pp. 54-55, 2009.
- [5] V. Bonneau, le Guide du m-tourisme, 2011.
- [6] S. Bourliataux-Lajoinie and A. Riviere « Le m-tourisme : une opportunité pour les destinations touristiques? Le cas des villes », 2013.
- [7] H. M. Chuang & C. H. Chang, "Verification of POI and Location Pairs via Weakly Labeled," Proceeding of the 24<sup>th</sup> International Conference on World Wide Web Pages 743-748 ISBN:978-1-4503-3473-0, 2015. DOI: 10.1145/2740908.2741715
- [8] A. Konstan and J. Riedl, "MovieLens Unplugged Experiences with an Occasionally Connected Recommender System," In Proceedings of the eighth international conference on intelligent user interfaces, pp. 263-266, 2013.
- [9] J. SU, H. YEH, P. YU and V. TSENG, "Music Recommendation Using Content and Context Information," IEEE Intelligent Systems, Issue 25, pp. 16-26, 2010.
- [10] R. Mooney and L. Roy "Content-based Book Recommending using Learning for Text Categorization," In Proceedings of the fifth ACM conference on Digital libraries, pp. 195-204, 2000.
- [11] B. Robin, "Hybrid Web Recommender Systems," pp. 377-408. 2007.
- [12] A. Das, M. Datar, A. Garg and S. Rajaram, "Google News Personalization: Scalable Online," WWW'07: Proceedings of the 16th International Conference on World, pp. 271-280, 2007.
- [13] K. Miyahara and M. Pazzani, "Collaborative Filtering with the Simple Bayesian Classifier," PRICAI 2000 Topics in Artificial Intelligence, pp. 679-689, 2000.
- [14] D. Pavlov, E. Manavoglu, D. Pennock and C. Giles, "Collaborative Filtering with Maximum", IEEE Intelligent Systems, Issue 19, pp. 40-48, 2004.
- [15] G. Bonnin, « Vers des systèmes de recommandation robustes pour la navigation Web: inspiration de la modélisation statistique du langage. Informatique », Université Nancy II, 2010.
- [16] N. Dennouni, "Orchestration des activités d'apprentissage mobile, " thèse de doctorat en cotutelle entre l'université de Lille1 et l'université de Sidi Bel Abbes, école doctorale sciences pour l'ingénieur, pp. 122-129, 2016.
- [17] N. Dennouni, Y. Peter, L. Lancieri and Z. Slama, "Towards to an Bio-inspired Orchestration of Mobile Learning Activities," International Journal of Modern Education and Computer Science (IJMECS), Vol.4, No.7, pp. 1-11, 2015. DOI: 10.5815/ijmeecs.2015.04.01
- [18] B. Robin, "Hybrid Recommender Systems: Survey and Experiments, User Modeling and User-Adapted Interaction," Kluwer Academic Publishers, pp. 331-370, 2012.
- [19] R. Singh, K. chuchra and A. Rani, "A Survey on the Generation of Recommender Systems," International Journal of Information Engineering and Electronic Business (IJIEEB), Vol.9, No.3, pp.26-35, 2017. DOI: 10.5815/ijieeb.2017.03.04.
- [20] M. Balabanović and Y. Shoham, "content-based, collaborative recommendation," Communications of the ACM Volume 40 Issue 3, March 1997, pp. 66-72. DOI: 10.1145/245108.245124
- [21] A. Gediminas and Alexander Tuzhilin. "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," pp 734 – 749, 2005.
- [22] M. J. Pazaani, "Framework for collaborative, contentbased and Demographic Filtering," Artificial Intelligent Review, vol. 13, pp. 393-408, 1999.
- [23] N. Dennouni, Y. Peter, L. Lancieri and Z. Slama, "Vers un système d'orchestration adaptatif et collaboratif des activités d'apprentissage en mobilité", 15<sup>ème</sup> conférence internationale sur l'extraction et la gestion des connaissances/Atelier GAST–Gestion et Analyse de données Spatiales et Temporelles, Luxembourg, 2015.
- [24] C. Desrosiers and G. Karypis, "Methods, A comprehensive survey of neighborhood-based recommendation," pp. 107-144, 2011.
- [25] M. J. Barranco, J. Noguera, C. Jorge and L. Martinez, "Advances in Intelligent Systems: A Context-Aware Mobile Recommender System," Management Intelligent Systems, First International Symposium, pp 153-162, 2012.
- [26] J. Noguera, M. J. Barranco, R. J. Segura and L. Martinez. "A mobile 3D-GIS hybrid recommender system for tourism", Information Sciences 215, pp 37-52, 2012.
- [27] L. Baltrunas, L. Bernd, S. Peer and R. Pers Francesco, "Context relevance assessment and exploitation in mobile recommender systems," Ubiquit Comput (2012) 16, 2012: pp.507-526, 2012.
- [28] T. Anteneh Alemu, A. Kumilachew Tegegne and A. Nega Tarekegn, "Recommender System in Tourism Using Case based Reasoning Approach," International Journal of Information Engineering and Electronic Business (IJIEEB), Vol.9, No.5, pp. 34-43, 2017. DOI: 10.5815/ijieeb.2017.05.05.
- [29] N. Dennouni, Y. Peter, L. Lancieri and Z. Slama, "Vers une approche Bio-inspirée pour l'orchestration pédagogique des activités mobiles d'apprentissage," La 7<sup>ème</sup> édition de la conférence EIAH (Environnements Informatiques pour l'Apprentissage Humain), 2-5 juin 2015, Agadir, Maroc.

- [30] A. Fridi and S. M. Benslimane, "Towards Semantics-Aware Recommender System: A LOD-Based Approach", *International Journal of Modern Education and Computer Science (IJMECS)*, Vol.9, No.2, pp.55-61, 2017. DOI: 10.5815/ijmeecs.2017.02.07
- [31] R. Sharma, S. Vinayak and R. Singh, "Guide Me: A Research Work Area Recommender System", *International Journal of Intelligent Systems and Applications (IJISA)*, Vol.8, No.9, PP.30-37, 2016. DOI: 10.5815/ijisa.2016.09.04
- [32] S. Walunj and K. Sadafale, "An Online Recommendation System for E-commerce Based on Apache Mahout Framework", *ACM*, 2013.
- [33] E. Guàrdia-Sebaoun, V. Guigue and P. Gallinari, "Recommandation Dynamique dans les Graphes Géographiques," *Road Materials and Pavement Design, RMPD* volume 22-No.1/2014. DOI: 10.3166/RMPD.22.1.-11.
- [34] N. S. Savage, N. E. Baranski, M. Chavez and T. Holle. "I'm feeling LoCo: A Location Based Context Aware Recommendation System," *Eighth International Symposium on Location-Based Services*, Vienna, pp. 37-54, 2011.
- [35] P. Yang and H. Fang, "Exploration of Opinion-aware Approach to Contextual Suggestion," in E. M. Voorhees, A. Ellis (eds), *NIST Special Publication 500-308 : The Twenty-Third Text REtrieval Conference Proceedings (TREC 2014)*, 2014.
- [36] X. Li, G. Cong, X. L. Li, T. A. N. Pham and S. Krishnaswamy, "Rank-GeoFM : A Ranking Based Geographical Factorization Method for Point of Interest Recommendation," *Proceedings of the 38<sup>th</sup> International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15*, ACM, New York, NY, USA, pp. 433-442, 2015.
- [37] J. Sang, T. Mei and C. Xu, "Activity Sensor: Check-In Usage Mining for Local Recommendation", *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 6, no 3, pp. 41:1-41:24, April 2015.
- [38] D. Gavalas and M. kenteris, "A web-based pervasive recommendation system for mobile tourist guides. *Pers Ubiquit Comput.*" pp. 759-770, 2011.
- [39] D. Nurbakova, L. Laporte, S. Calabretto and J. Gensel, « ANASTASIA : recommandation de séquences d'activités spatio-temporelles », *Rencontres Jeunes Chercheurs en Recherche d'Information (RJCRI CORIA-CIFED)*, Mar 2016, Toulouse, France. pp.10. <hal-01328062>
- [40] C. Zhang, H. Liang, K. Wang and J. Sun, "Personalized Trip Recommendation with POI Availability and Uncertain Traveling Time," *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM '15*, ACM, New York, NY, USA, pp. 911-920, 2015.
- [41] D. Menezes, A. Lacerda, L. Silva, A. Veloso and N. Ziviani, "Weighted Slope One Predictors Revisited," *International World Wide Web Conference Committee (IW3C2)*. May 13-17, Rio de Janeiro, Brazil. ACM 978-1-4503-2038-2/13/05, 2013.

## Authors' Profiles



of Chlef (n.dennouni@univ-chlef.dz).

**Nassim DENNOUNI** is a doctor in computer science since January 2016 and an old member in the NOCE team of the CRISTAL laboratory at Lille 1 university. He is also member in the ISIBA team of the EEDIS laboratory at the Djilali Liabes University. Actually, he works as teacher of computer science at Hassiba BENBOUALI University



**Yvan PETER** is an associate Professor in computer science and member in the NOCE team of the CRISTAL laboratory at Lille 1 university (Yvan.Peter@univ-lille1.fr).



**Luigi LANCIERI** is a Professor of Computer Sciences and the Head of the NOCE team of the CRISTAL laboratory at Lille1 University (Luigi.lancieri@univ-lille1.fr).

**Zohra SLAMA** is an associate professor in computer science and member in the ISIBA team of the EEDIS laboratory at the Djilali Liabes University of Sidi Bel Abbes (slama@univ-sba.dz)

**How to cite this paper:** Nassim DENNOUNI, Yvan PETER, Luigi LANCIERI, Zohra SLAMA, "Towards an Incremental Recommendation of POIs for Mobile Tourists without Profiles", *International Journal of Intelligent Systems and Applications (IJISA)*, Vol.10, No.10, pp.42-52, 2018. DOI: 10.5815/ijisa.2018.10.05