Information Sciences Journal manuscript No.
(will be inserted by the editor)

A method for Context-Aware Recommender System

Xochilt Ramirez-Garcia · Mario Garcia-Valdez

Received: date / Accepted: date

Abstract In recent years the context was integrated in recommendation techniques to get suitable recommendations for users in an enviroment where the main role is the specific situation of user and all the factors that are implicit in that situation. This research proposes a context-aware recommender system for restaurants that uses post- filtering and fuzzy logic, this method involves two techniques of recommendation: collaborative filtering and content-based. A hybrid method is proposed to reduce the weaknesses of recommendation techniques. The context-aware recommender system tries to help users to find relevant restaurants taking in account the context. The goal is to improve the user satisfaction in the user's interaction with the system. To evaluate the context- aware recommender system 10 users and 176 restaurants was used for an on-line test. Two performance metrics were used: task-success and task-on-time, in order to measure the level of user satisfaction.

Keywords Content-Based · Context · Collaborative Filtering · Fuzzy Logic · Recommender Systems

1 Introduction

The importance of contextual information has been recognized in different domains and disciplines. Many applications could be improved with the implementation of context such as e-commerce [5], [15]; vacation package [12], [11]; movies [7] or music [4] is not enough to consider users and items in the recommendation process because the circumstances could be changing in time. The majority of recommender systems focus on recommending relevant items for

Xochilt Ramirez-Garcia
E-mail: xochilt.ramirez@gmail.com

Mario Garcia-Valdez
mariosky@gmail.com

users without any additional contextual information such as time, location, companion or place, however, its important to incorporate the contextual information to get recommendations in order to deliver items that are in the user context.
Due to the huge usefulness of context in recommender systems, more companies started incorporating some contextual information in recommendation engines. For example when selecting a song for the customer, Sourcetone interactive radio [9] takes in the consideration the current mood of the listener previously specified. When Sourcetone obtains recommendations, the context about listeners mood is applicable for providing better recommendations. Today, many application are using context to increase the level of important aspects as user satisfaction, recommendation quality, usefulness and more, because of the amount of information available in Internet that it should be filtered and personalized for users.
This research presents an architecture proposed for a context- aware recommender system in the domain of restaurants. The objetive is to demonstrate how the level of user satisfaction is increased using context.
The section 2 explains some fundamentals related to context-aware recommender systems, the section 3 describes the proposed method for context-aware recommender system and each one of its components, the section 4 explains the experiments and results of the system evaluation, finally, the section 5 presents the conclusions and future work proposed.

2 Context-Aware Recommender Systems

2.1 Context definition

The interaction among persons and computers in social applications takes place in a specific context referring the physical and social situation where computational devices are involved. The context is determined by the persons interaction, the objectives of their interaction, the time and place where the interaction is. Context is everything about the relevant situation for an application and their set of users. The important aspects are not possible to define because all situations will be change from situation to situation. This research adopts the context definition proposed by A. Dey [6]:
"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."
This definition makes it easier for an application developer to define the important context in a specific application. Then, if any information can be used to describe the situation of the user in an interaction, that information represents the context.

3 Proposed method

3.1 Restaurant model

An effective online recommender system must be based upon an understanding of consumer preferences and successfully mapping potential products into the consumers preferences [1]. Pan and Fesenmaier [14] argued that this can be achieved through the understanding of how consumers describe in their own language a product, a place, and the experience when they are consuming the product or visiting the place.
Traditionally, choosing a restaurant has been considered as rational behavior where a number of attributes contribute to the overall usefulness of a restaurant. For example: food type, service quality, atmosphere of the restaurant, and availability of information about a restaurant, plays an important role at different stages in consumers desitions making [3]. While food quality and food type have been perceived as the most important variables for consumers restaurant selection, situational and contextual factors have been found to be important also. Due to this in Kivela [10]identifies 4 distinct types of restaurants: 1) fine dining/gourmet, 2) theme/atmosphere, 3) family/popular, and 4) convenience/fast-food; and Auty [3] identifies 4 types of dining out occasions: 1) namely celebration, 2) social occasion, 3) convenience/quick meal, and 4) business meal.
Taking in account the context, the restaurant model proposed for context-aware recommender system contains 55 attributes about the restaurants features. An exploration about contents of models of others works were compared to define the suitable information in the model. Therefore, the restaurant model is a binary vector with the following attributes: 1) price range, 2) payment type, 3) alcohol type, 4) smoking area, 5) dress code, 6) parking type, 7) installations type, 8) atmosphere type, and 9) cuisine type. An example of restaurant model in the application is depicted in Fig.1 with possible values of the context represented by a binary vector where 1 means that the restaurant has the property that corresponds to the position value. Additionally, the restaurant model contains contextual information such as reviews of users, ratings average, and geographycal location.

3.2 User profile

The user's profile is derived from the ratings matrix. Let $U = [u_1, u_2, ...u_n]$ the set of all users and $I = [i_1, i_2,...i_n]$ the set of all items, if R represent the ratings matrix, an element $R_{u,i}$ represents a users rating $u \in U$ in a item $i \in I$. The unknown ratings are denoted as $\neq$. The matrix R can be decomposed into rows vectors, the row vector is denoted as $\overrightarrow{r_u}$=$[R_{u,1}...R_{u,|I|}]$ for every $u \in U$. Therefore, each row vector represents the ratings of a particular user over the items. Also denote a set of items rated by a certain user u is denoted as $I_u = i \in I | \forall i : R_{u,i} \neq \emptyset$. This set of items rated represents the user preferences

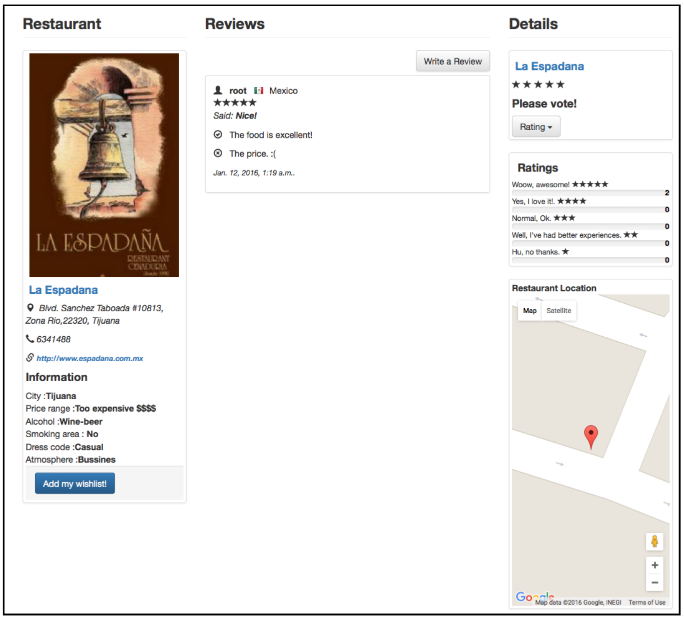


Fig. 1: Example of the user interface for restaurant model.

where for each domain element $R_{u,i} \in [1-5]$ represents the intensity of the user interest for the item.

In context-aware recommender system, user profile has contextual information such as: 1) price range, 2) current location, 3) cuisine types, 4) attributes or features of restaurants that the user want, 5) the reviews posted, and 6) the favorite restaurants list. The user profile is stored in database and it is available for queries request, and it can be changed by users many times in a session. The information used to recommendations is the last one register stored.

3.3 Expert Recomendation

Fuzzy logic is a methodology that provides a simple way to obtain conclusions of linguistic data. Is based on the traditional process of how a person makes decisions based in linguistic information. Fuzzy logic is a computational intelligence technique that allows to use information with a high degree of inaccuracy; this is the difference with the conventional logic that only uses concrete and accurately information [16].

In this work, fuzzy logic is used to model fuzzy variables that highligh in the popularity of a restaurant. The context-aware recommender system has implemented a FIS that represents the expert recommendation. The expert FIS generates recommendations when the recommendation techniques (collaborative filtering, content-based) are not getting results because of the cold start

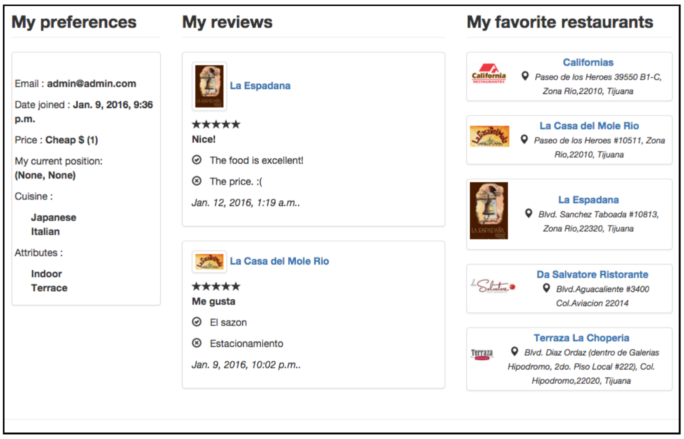
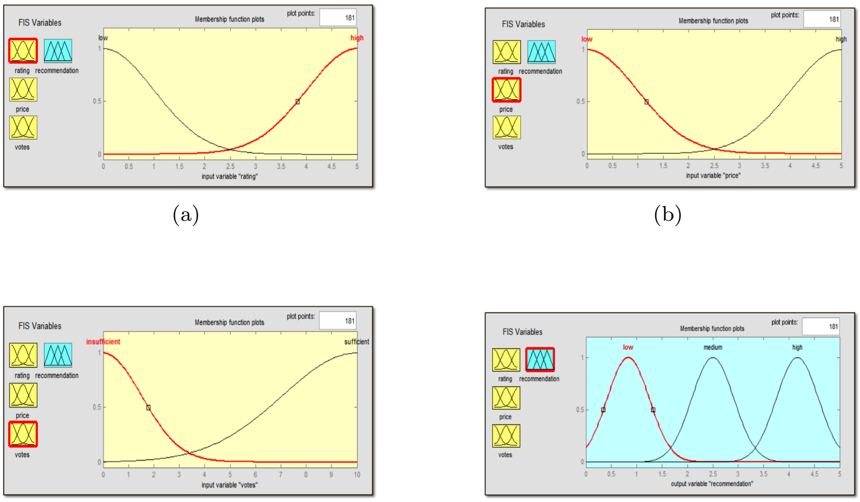


Fig. 2: Example of user interface for user profile.

problem.
The FIS proposed has 3 input variables [8]: 1)*rating* is an average of ratings that has a particular restaurant inside the user community; the domain of variable is 0 to 5 and contains 2 membership functions labeled as *low* and *high*(Fig.3a), 2)*price* represents what kind of price has a particular restaurant; the domain of variable is 0 to 5 and contains 2 membership functions labeled as *low* and *high* (Fig.3b), and 3)*votes* is used to measure how many items have been rated by the current user, i.e., the participation of the user, if the user has rated few items (less than 10) is not sufficient to make accurate predictions(Fig.3c), the domain of variable is 0 to 10 and contains 2 membership functions labeled as *insufficient* and *sufficient*. The output variable is *recommendation*, represents a weight for each restaurant proposed by the expert considering the inputs mentioned above, the domain of variable is 0 to 5 and contains 3 membership functions labeled as *low*, *medium* and *high* (Fig.3d).

The proposed FIS in this research(Fig.4) represents the users experience and their knowledge about restaurants. This factors are considered important for users that visiting a restaurant. This information is recovered of user profile and restaurant profile; and the system uses this information to get weights that influence in the final recommendations. The FIS uses 5 inference rules that involve the variables of inputs and output.The input variables determine the recommendation activation; each input variable contains labels as *low* and *high* that also correspond to memberships functions of Gaussian type. For the output variable *recommendation* the labels *low*, *medium*, and *high* are used with membership functions Gaussian type also. The rules are:

1. *If rating is high and price is low then recommendation is high.*
2. *If rating is high and votes is sufficient then recommendation is high.*
3. *If rating is high and votes is insufficient then recommendation is medium.*



(a) (b) (c) (d)

Fig. 3: The membership functions of the expert system.

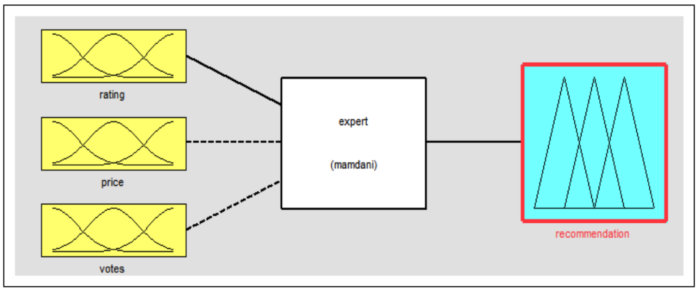


Fig. 4: Fuzzy Inference System of expert.

4. *If rating is low and price is high and then recommendation is low.*
5. *If rating is low and votes is insufficient then recommendation is low.*

3.4 Contextual Recommendation

The interface of the system(Fig.5) allows to collect contextual information such as type of price, restaurant's attributes, type of cuisine and geographical location. The contextual information is used for adjust the final recommendations list. For example, geographical location is used to get restaurants around 2 kilometers of distance, next, the list of nearby restaurants is displayed for the user. If context-aware recommender system considers another attributes

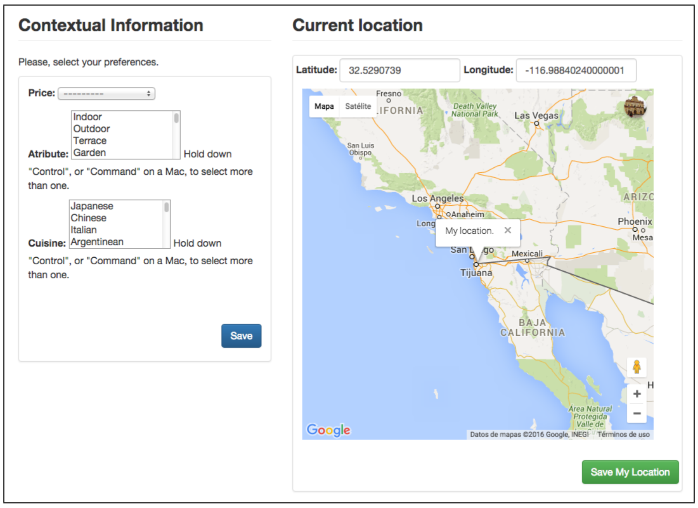


Fig. 5: System interface to collect contextual information.

as type of price and type of cuisine preferred by the user, the system gets restaurants matched in this context, and so on.

When the recommendation process is finished, the system displays the restaurants recommended according the information provides by the user. The context-aware recommender system contains four techniques to display recommendations. The interface in Fig.6 shows recommendations: 1) Expert, 2)Content- based, 3) Collaborative filtering and 4)Nearby.

3.5 Architecture

The architecture for poposed method is depicted in the Fig.7. In the first part, the three techniques of recommendations are suplied by the rating matrix to obtain the recommendation list of each one. In the middle,the content-based uses cosine similarity to calculate the similarity between the items, next, collaborative filtering uses the Pearson correlation to calculate similarity. In the last part, the recommendation lists for the user. Subsequently, the recommendation lists are reduced when filter context is applied, i.e., the recommendations are adjusted for the user context. At the end, a contextual recommendations list is displayed in the user interface(Fig.6).

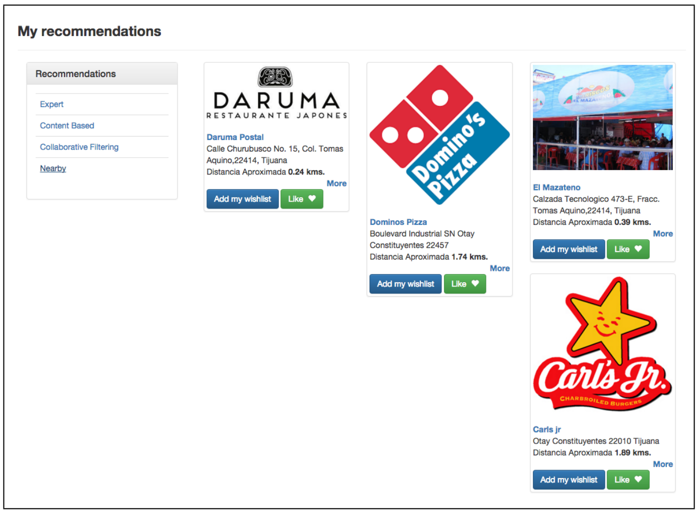


Fig. 6: System inferface of recommendations section.

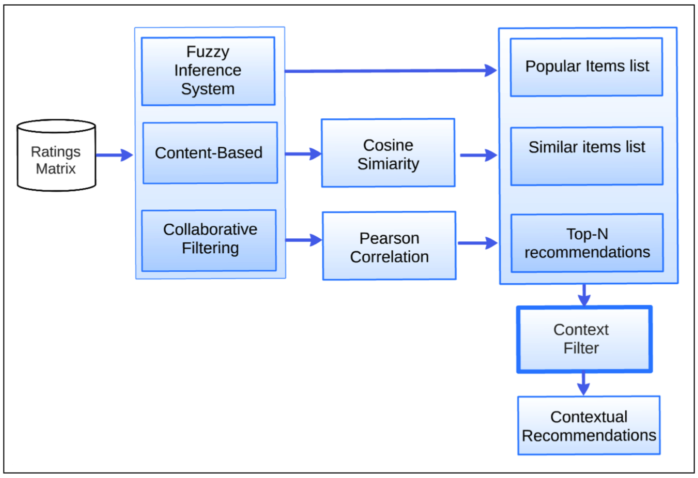


Fig. 7: Architecture proposed.

4 Evaluation of the system

4.1 Database

The database was collected from Web sites of Tijuana restaurants. There are 176 restaurants for evaluation by 10 real users. The rating matrix contains the users profiles, their tastes and preferences are stored such as a ratings vector.

4.2 Experiments and results

To evaluate context-aware recommender system was used the *task success* and *time-on-task* metrics.
The *task success metric* is perhaps the most widely used performance metric. It measures how effectively users are able to complete a given set of tasks. The *time-on-task metric* is a common performance metric that measures how much time is required to complete a task [2].
Task success is something that almost anyone can do. If the users cant complete their tasks, then something is wrong. When the users fail to complete a simple task can be an evidence that something needs to be fixed in the recommender system. The usability test consist of a list of simple tasks for users that they shall perform in the system to complete the test. Before to start, a minimal description about the system for user was explained. The tasks list are the following:

1. *Rated a restaurant without context.*
2. *Add context to the user profile.*
3. *Filter restaurants by favorite context.*
4. *Find information of a specific restaurant.*
5. *Find all the reviews of a specific restaurant.*
6. *Find section of my favorite restaurants.*
7. *Add a review of a restaurant.*
8. *Find the most popular restaurants.*
9. *Add a restaurant to your wishlist.*
10. *Get recommendations based on expert opinion.*
11. *Get the recommendations content-based.*
12. *Get the collaborative recommendations.*
13. *Get recommendations of the nearby restaurants.*

The users were students of Tijuana Institute of Technology, they never interacted with the system interface. To say if the user completed the task was based in the complete realization of the same without taking in account the time used. The simple tasks was doing in less than a minute but the result shows that the user had some problems to complete it.
Each user did the test according the task list, the result is depicted in Fig.8 where the axis represent the task number and percent of success. The chart shows that only 3 tasks weren't accomplished successfuly, the task 5, 6 and 7. The issue with task 5 and task 7 was that users can not found easily the reviews and add it was complicated also. In task 6 the issue was that the user chose the wishlist section in place of the favorites restaurants section.

The time it takes a participant to perform a task says a lot about the usability of the application. In almost every situation, the faster a participant can complete a task, the better the experience. In fact, it would be pretty unusual for a user to complain that a task took less time than expected [2].
Then, task-on-time was applied to measure time that an user did the task. A

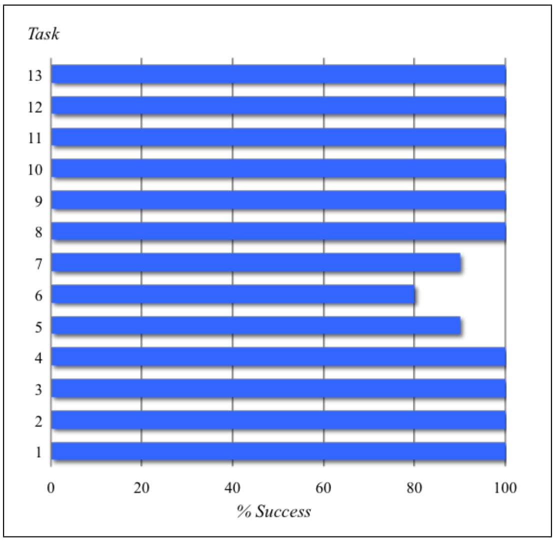


Fig. 8: Representation of the percent of success for each task.

Table 1: Time on task data for 10 users and 13 tasks.

Task	Us1	Us2	Us3	Us4	Us5	Us6	Us7	Us8	Us9	Us10
1	12	28	24	30	19	33	23	16	5	7
2	3	4	17	5	17	134	9	16	12	11
3	123	69	159	53	69	113	44	41	70	98
4	20	4	86	40	13	4	17	3	20	3
5	50	10	63	50	7	11	10	5	20	Null
6	10	30	28	27	5	46	Null	7	Null	34
7	10	20	16	8	15	Null	9	24	16	28
8	18	24	10	10	5	3	27	4	5	6
9	5	6	31	4	45	9	12	5	3	8
10	15	17	15	11	10	19	13	10	20	20
11	30	15	20	16	20	22	15	13	18	20
12	12	14	19	14	40	10	17	17	15	15
13	25	15	15	14	10	10	11	10	10	25

resume of the time tasks for each user it is in Table 1, null values mean that the user didn't the task.

To measure the efficiency of the metric it was chose an confidence interval. In this way, it is observed the time variability within the same task and also helps visualize the difference across tasks to determine whether there is a statistically significant difference between tasks. The obtained information is in Table 2, the median was used to calculate the confidence interval.

In the next step the USE *(Usefulnes, Satisfaction, and Ease of Use)* questionnaire [13] was applied in order to get the user's feedback and comments for to know about the difficults in the test. The USE questionnaire consists of 30

Table 2: Table of confidence interval per task with a confidence level of 95%.

Task	Median	CI 95%	Upper bound	Lower bound
1	20	5.96	25.96	14.04
2	11.5	0.81	12.31	10.69
3	69.5	25.57	95.07	43.93
4	15	16.34	31.34	-1.34
5	15.5	14.84	30.34	0.66
6	27.5	11.57	39.07	15.93
7	16	5.19	21.19	10.81
8	8	5.80	13.80	2.20
9	7	9.43	16.43	-2.43
10	15	2.44	17.44	12.56
11	19	3.00	22.00	16.00
12	14.5	5.51	20.01	8.99
13	12.5	3.89	16.39	8.61

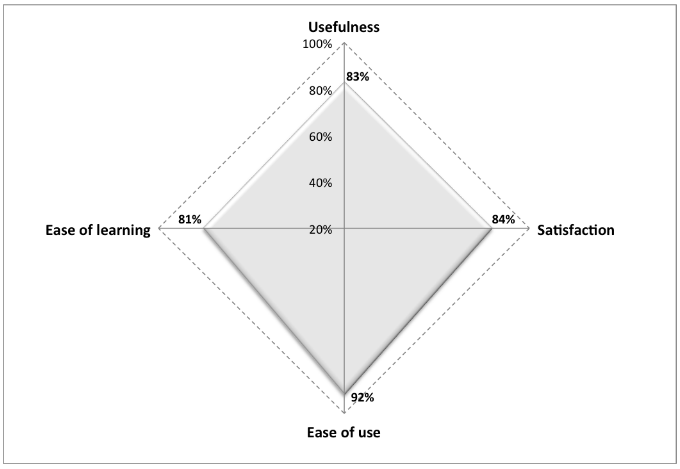


Fig. 9: The radar chart that depicts the four axis evaluated in the questionnaire.

rating scales divided into 4 categories: Usefulness, Satisfaction, Ease of Use, and Ease of Learning. Each is a positive statement to which the user rates level of agreement on a 7-point Likert scale. The USE questionnaire allows to get values for Usefulness, Satisfaction, Ease of Use, and Ease of Learning, the visualizing the results is in the Fig.9 , where the four axis of the radar chart represent the values of percent which users rated positively this factors with respect to their interaction with the context-aware recommender system.

5 Conclusions and future work

We observed the users behaviour to identify the most frecuently difficults and doubts about tasks. We did a brief interview with users after the test in order to understand their feelings or mood, their ideas about the experience, and overall, their opinion about the context-aware recommender system. The conclusions are based in user's comments, then the main errors in the system interface are summarized in three points:

1. Incomplete information for user, i.e., the system doesn't had enough and clear information to be a friendly interface, and therefore the user couldn't do easily a task.
2. Fails in design, because of unordered elements in the screen, in other words, the elements are not in the correct site into the screen to be easily identified per users.
3. Fails in the language and confusion, because of the english language is not the native language of the users.

The three points mentioned are related to the null values in data table (see Table 1), some users didn't the task because they were confused, so they decided to omit the task. The null values weren't took in account when the median was calculated (see Table 2).
The USE questionnaire was useful to identify the weaknesses in the context-aware recommender system. The percent is upper of the acceptable (80%), the results allow to say that the system has a good performance.
For the future work we proposed to improve the problems found in the user interface, so the proposals are the following:

1. Redesign the user interface could helps to be more friendly for users. Due to the issues, the redesign involves:
   (a) Analyze the amount of information enough for a easy understanding, i.e., how much information the user needs seeing without overload it.
   (b) Modify the tasks descriptions in the most simple way to avoid confusion.
   (c) Add more language functionalities for to facilitate the tasks for users.
2. To apply the usability test again with the changes in the interface in order to observe the level of improves and to compare the results.
3. Apply an statistical test to analize the results.
4. Add collaborative filtering based on model (matrix factorization technique) within the context-aware recommender system in order to improve the level of user satisfaction in the context.
5. Add any contextual factors (such as companion, time of day, budget, etc.) in order to include more context information that could be relevant in the recommendations.

References

1. Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. pages 217–253, 2011.

2. William Albert and Thomas Tullis. *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. Newnes, 2013.
3. Susan Auty. Consumer choice and segmentation in the restaurant industry. *Service Industries Journal*, 12(3):324–339, 1992.
4. Linas Baltrunas, Marius Kaminskas, Bernd Ludwig, Omar Moling, Francesco Ricci, Aykan Aydin, Karl-Heinz Lüke, and Roland Schwaiger. Incarmusic: Context-aware music recommendations in a car. In *EC-Web*, volume 11, pages 89–100. Springer, 2011.
5. Rebecca Bulander, Michael Decker, B Kolmel, and G Schiefer. Enabling personalized and context sensitive mobile advertising while guaranteeing data protection. *Proceedings of EURO mGOV 2005*, page 445C454, 2005.
6. Anind K Dey. Understanding and using context. *Personal and ubiquitous computing*, 5(1):4–7, 2001.
7. Eyrun A Eyjolfsdottir, Gaurangi Tilak, and Nan Li. Moviegen: A movie recommendation system. *UC Santa Barbara: Technical Report*, 2010.
8. Mario García-Valdez and Brunett Parra. A hybrid recommender system architecture for learning objects. In *Evolutionary Design of Intelligent Systems in Modeling, Simulation and Control*, pages 205–211. Springer, 2009.
9. Arefin Huq, Juan Pablo Bello, and Robert Rowe. Automated music emotion recognition: A systematic evaluation. *Journal of New Music Research*, 39(3):227–244, 2010.
10. Jaksa Jack Kivela. Restaurant marketing: selection and segmentation in hong kong. *International Journal of Contemporary Hospitality Management*, 9(3):116–123, 1997.
11. Qi Liu, Enhong Chen, Hui Xiong, Yong Ge, Zhongmou Li, and Xiang Wu. A cocktail approach for travel package recommendation. *Knowledge and Data Engineering, IEEE Transactions on*, 26(2):278–293, 2014.
12. Qi Liu, Yong Ge, Zhongmou Li, Enhong Chen, and Hui Xiong. Personalized travel package recommendation. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 407–416. IEEE, 2011.
13. Margaret Morris and Arnie Lund. Experience modeling: How are they made and what do they offer. *LOOP: AIGA Journal of Interaction Design Education*, (7), 2001.
14. Bing Pan and Daniel R Fesenmaier. Online information search: vacation planning process. *Annals of Tourism Research*, 33(3):809–832, 2006.
15. J Ben Schafer, Joseph Konstan, and John Riedl. Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce*, pages 158–166. ACM, 1999.
16. LA Zedeh. Knowledge representation in fuzzy logic. *Knowledge and Data Engineering, IEEE Transactions on*, 1(1):89–100, 1989.