Noname manuscript No.
(will be inserted by the editor)

A method for Context-Aware Recommender System

Xochilt Ramirez-Garcia · Mario Garcia-Valdez

the date of receipt and acceptance should be inserted later

Abstract In recent years the context was integrated in recommendation techniques to get suitable recommendations for users in an enviroment where the specific situation of user and all the factors that are implicit in that situation are fundamental. This research proposes a context-aware recommender system for restaurants that involves post- filtering and fuzzy logic. Post-filtering involves two techniques of recommendation: collaborative filtering and content-based, the common problems in these algorithms are cold-start and over-specialization, respectively. In order to reduce the effects of these problems the hybrid method is proposed. The context-aware recommender system tries to help users to find relevant restaurants considering context. Overall, the goal is to improve the user satisfaction. 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

Recommender systems are a technique to provide suggestions of useful items for a user. The suggestion relate to various decition-making processes, for instance, what items to buy, what music to listen to or what on-line news to read. *Item* is the general term to denote what product or service the system recomends for each user. A recommender system normally focuses in a kind of

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

Mario Garcia-Valdez
mariosky@gmail.com

item. Recommender sistems arise to cover the lack of personal experience to evaluate the overwhelming number of alternative items that a Web site may offer [36]. Recommender systems development initiated from a rather simple observation: individuals often rely on recommendations provided by others in making routine and daily decisions [28], [31]. For example people commonly relies on suggestions when selecting a book to read; for hire human resources in a company, the department counts on recommendation letters in their recruiting decisions; and when selecting a movie to watch, people tend to read and relies on the movie reviews that a film critic has written.
Recommender systems tries to imitate this human behaviour, to recommend a product in daily life is normally the most effective manner to achieve in this systems that the user prefers an item in a Web site. Over the time, the improvements for recommender systems are focused in differents techniques that involves the context in the recommendation process. The importance of contextual information has been recognized in different domains and disciplines. In recommender systems, context implies the situations around of the user when it is interacting with the system and all the information that this situation represents. The most formal definition is proposed by Annin K. Dey [17]: *"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 to define the important context in a specific recommender system.
However, some authors make an understandable comparison of different contexts such as Bazire et.al. [9] that compares different definitions of context in different fields and conclude that is complicated makes a unifying definition of context because of the nature of the concept in the disciplines. In computer sciences Fischer G. [19] takes the context such as the interaction between humans and computers in socio-technical systems that takes place in a certain context referring to the physical and social situation in which computational devices and environments are embedded. Also identifies the important aspects to consider when the context is used in its framework: how it is obtained the context, how is represented the context and what objectives and purposes has the context in a particular application is used.
Context-aware recommender systems are gaining ever more attention because of its performance and adaptability for different domains, the way to improve personalized recommendations based in contextual factors is an important technique to increase the benefits in many domains. Nowadays many companies are incorporating some kind of context in their recommendation engines, the application covers fields such as E-commerce [39], [10], music [37], [6], places of interest [8], movies [18], vacation packages [27], [26], travel guides [38],e-learning [33] and restaurants [15].
The majority of these recommender systems are focused on recommending relevant items for users without additional contextual information such as time, location, companion or place. However, recent systems incorporate the contextual information to make recommendations in order to deliver items adjusted

to the users' current context. For instance, the site Sourcetone interactive radio [22] when selecting a song for the customer takes in consideration the listener's current mood in order to provide a better recommendation.
The context is used for increase important aspects such as user satisfaction, recommendation quality, usefulness and more.
This research presents an architecture proposed for a context-aware recommender system in the restaurant's domain. The objetive is to demonstrate how the level of user satisfaction is increased through the context when the system recommends restaurants in a specific context. The architecture contains three techniques for this process:

1. *Fuzzy Inference System* to recommend such as an expert in restaurants, it considers the inputs *ratings average*, *price of restaurant* and *number of votes* to infer how relevant a restaurant is for the user.
2. *Content-based technique* utilize the restaurant profiles to compare how *similar* is a restaurant with respect of another, i.e. the restaurants that are *similar* to restaurants that the user rated with high rating. The idea is to find restaurants with similar features to recommend.
3. *Collaborative filtering technique* is based in the user profile to identify the user preferences and to find neighbors that have the same tastes. The recommendation consist in the suggestions of other users with similar tastes that rated restaurants that are not rated for the current user. A top-N list of restaurants is obtained to recommend for the user.

The results of the three techniques are the list of recommendations for the user that are adjusted in the user context. This is the last step and is represented as *context filter* in the architecture, then the recommender system obtains a list of contextualized recommendations. The architecture proposed works simultaneously to obtain recommendations, the hybris method allows the recommender system to generate suggestions despite the scarse of user information. When the system face up to the cold-start problem or the over-specialization problem, the hybrid method responses properly to show results for the user. An expert recommendation contributes to this goal, the rules specified in the Fuzzy Inference System helps to inferring the most suitable restaurant considering the users' opinions.
The context implementation makes better recommendations, the restaurants matched in the same context of the user will be added within the final recommendation list. The context-aware recommender system tries to increase the level of user satisfaction. To meet this goal two metrics were utilized to test the system in an on-line test. Ten users were selected to interact with the system, subsequently an analysis about the system performance and main issues highlited by the users was realized.
The rest of this paper is organized as follows: the state of the art is explained in next section, section 3 describes the proposed method for context-aware recommender system and each one of its components, the section 4 describes the system evaluarion, the manner how the experiments were done and the

results obtained, finally, the section 5 presents the conclusions and future work proposed.

2 State of the art

Recommender system is a technology used by many authors for personalization of information. The amount of information in Internet is rising quickly and the users every time has less capability to search the most relevant information among big databases. Many recommender system are used in several domains as mentioned in section 1. The fundamental is the profit that provides in many of these domains that sometimes represents the key of success for the application. Some works utilize social information to recommend such as M. Manca et.al. [29] where the friend recommender system is applied in the social bookmarking domain, its goal was to infer the interest of users from content selecting the available information of the user behavior and analyzing the resources and the tags bookmarked for each user, therefore the recommendations are through mining user behavior in a tagging system, analyzing the bookmarks tagged of the user and the frecuency for each used tag. J.Yao et.al. [42] proposes a new product recommendation approach for new users based on the implicit relationships between search keywords and products. The relationships between keywords and products are represented in a graph and relevance of keywords to products is derived from attributes of the graph. The relevance information is utilized to predict preferences of new users. J. Golbeck et.al. [21] presents FilmTrust, a website that integrates Semantic Web-based social networks, augmented with trust, to create prediction movie recommendations. Trust takes on the role of a recommender system forming the core of an algorithm to predict a rating for recommendations of movies. This is an example of how the Semantic Web, and Semantic trust networks in particular, can be exploited to refine the user experience.
Traditional recommender techniques has its pros and cons, for instance, the ability to handle data sparsity and cold-start problems or considerable ramp-up efforts for knowledge acquisition and engineering. Establish hybrid systems that combine the strengths of algorithms and models to overcome some of the shortcomings and problems has become the properly manner to improve the difficults for each algorithm. An example is presented by L.Castro et.al. [13] a hybrid recommender system for the province of San Juan, Argentina, to recommend tourist packages based on preferences and interest of each user, artificial intelligence techniques are used to filter and customize the information. The prototype of recommender system utilizes three techniques to recommend: demographic, collaborative and content-based. The goal is to recommend tourist packages that matches with the user profile. L. Martinez et al. [30] presents REJA, a hybrid recommender system that involves collaborative filtering and knowledge-based model, that is able to provide recommendations in some situation for user; besides it provides georeferenced information about the recommended restaurants. Balabanovic et.al. [5] presents Fab, a hybrid

recommender system for automatic recognition of emergent issues relevant to various groups of users. It also enables two scaling problems, pertaining to the rising number of users and documents, to be addressed. Claypool et.al. [16] presents P-Tango system that utilizes content-based and collaborative filtering techniques, it makes a prediction through the weighted average that includes content-based prediction and collaborative filtering prediction. The weights of predictions are determined on a per-user basis, allowing the system to determine the optimum mixture of content-based and collaborative recommendation for each user. Pazzani M. [35] presents Entree as a hybrid recommender system that it does not use numeric scores, but rather treats the output of each recommender (collaborative, content-based and demographic) as a set of votes, which are then combined in a consensus scheme. The recommender system includes information such as the content of the page, ratings of users and demographic data about users. Others works with hybrid recommender systems are ProfBuilder [2], PickAFlick [12] and [41], where are presented multiple recommendation techniques. Usually, recommendation requires ranking of items or selection of a single best recommendation, at this point some technique must be employed to recommend.

Traditional recommender systems such as above mentioned, tend to use simple user models. For example, user-based collaborative filtering generally models the user as a vector of item ratings. As additional observations are made about users preferences, the user models are extended, and the user preferences is used to generate recommendations. This approach, therefore, ignores the notion of any specific situation, the fact that users interact with the system within a particular context and that preferences of items might change in another context.

Overall, the context is able to make the recommender system be powerful that is adaptable to the changing user's situation.

The context is defined in the domain of the application and the system has a context model that provides the information for the recommender system. For instance Ricci et.al. [6] uses the context in music domain using a model-based paradigm, in this context-aware recommender system the context was defined as a set of independent contextual factors(independent in order to get a mathematical model) such as *driving style, road type, landscape, sleepiness, traffic conditions, mood weather and natural phenomena* to specifies the relevant context for the music recommendation. In order to estimate the relevance of selected contextual factor, the users were requested to evaluate music tracks in different contextual situations for each genre. The prediction takes in account this relevance to recommend music tracks prefered by the user according the genre and the contexts mentioned.

In restaurant domain Chung-Hua et al. [15] presents a context-aware recommender system for mobiles using a post-filtering paradigm, the architecture involves a model client-server that works with a request of data in the client side for the server side. Subsequently, taking in account the contextual factors to filter the properly restaurants to recommend. The context-aware recommender system uses such as contextual factors *location and season*, also utilize

the user preferences to personalize the recommendations in the user context. Baltrunas et.al. [7] presents ReRex for tourism, a context-aware recommender system based in a model-based paradigm, the system recommends and provides explanations about the why the places of interest(PoI) are recommended. The proposed model computes a personalized context dependent rating estimation. Subsequently, in order to generates the explanation of recommendation the system uses the factor that in the predictive model has the lasgest positive effect on the rating prediction for the point of interest. The set of contextual factors considered in ReRex are *distance*, temperature, weather, season, companion, time day, weekday, crowdedness, familiarity, mood, budget, travel length, transport and travel goal. The main issue in ReRex system is the low user satisfaction because of the explanations not able to be understood, however the users recognize that the explanation is a very important component that it influence the system acceptance.
Noguera et. al. [32] presents a context-aware recommender system for tourism based in REJA that utilizes the location through a 3D-GIS system, the application uses progressive downloading and rendering of 3D maps over mobiles networks. It is also in charge of tracking the users location and speed based on GPS and the requesting. The system utilizes pre-filtering and post-filtering paradigm. Pre-filtering is used to reduce the number of items considered for the recommendation according to the users location, and post-filtering is applied to re-rank the previous top-N list according to the physical distance from the user for each recommended restaurant. The disadvantage in this system is the lack of user reviews, because the recommendations are based only in the location point without consider the experience of other diners concerning the recommended restaurant. Cena et al. [14] presents a tourist guide for context in intelligent content adaptation. UbiquiTO system is a tourist guide that integrates different forms of context-related adaptation: *to the media device type, to the user characteristics and preferences, to the physical context of the interaction*. UbiquiTO uses a rule-based modeling approach to adapt the content of the provided recommendation, such as the amount, type of information and features associated with each recommendation. Bulander et.al [11] presents the MoMa-system that offers proactive recommendations using a postfiltering approach for matching order specifications with offers. When creating an order, the client application will automatically fill in the appropriate physical context and profile parameters, for example, *location* and *weather*, so that, for example, the facility should not be too far away from the current location of the user and beer gardens should not be recommended if it is raining. On the other side, advertisers suppliers put offers into the MoMa-system. These offers are also formulated according to the catalogue. When the system detects a pair of context matching order and offer, the end user is notified, in the preferred manner (for example, SMS, email). At this point, the user can decide whether to contact the advertiser to accept the offer. Finally, Schifanella et al. [40] develops Mob-Hinter, a *context-dependent* distributed model,where a user device can directly connect to other mobile devices that are in *physical proximity* through ad-hoc connections, hence relying on a very limited

portion of the users community and just on a subset of all available data (prefiltering). The relationships between users are modeled with a similarity graph. MobHinter allows a mobile device to identify the affinity network neighbors from random ad-hoc communications. The collected information is then used to incrementally refine locally calculated predictions, with no need of interacting with a remote server or accessing the Internet. The Recommendations are computed using the availables rating of the user neighbors.

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 [34] 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 [4]. 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. Kivela [23] identifies 4 distinct types of restaurants: *1) fine dining/gourmet, 2) theme/atmosphere, 3) family/popular, and 4) convenience/fast-food*; and Auty [4] identifies 4 types of dining out occasions: *1) namely celebration, 2) social occasion, 3) convenience/quick meal, and 4) business meal.* Lewis [25]considered five important factors: *food quality, menu variety, price, atmosphere, and convenience factors* for a restaurant. Jang et.al. [24] suggested three factors to consider: *service quality, product quality and, atmospherics* as restaurant attributes affecting perceived quality of restaurant experiences. The total dining experience in a restaurant is comprised of not only food itself, but also the *atmosphere, instalations and the service quality.*
Taking in account the context proposed by authors, the restaurant model proposed for context-aware recommender system was unified 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

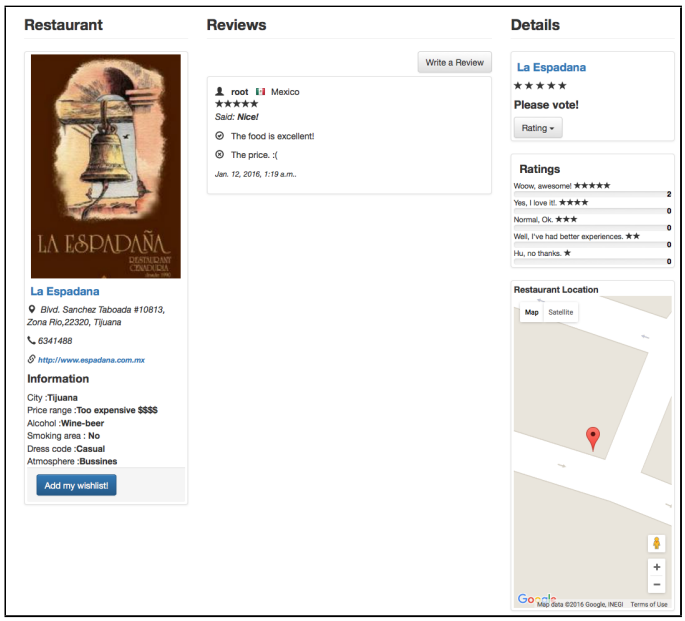


Fig. 1: Example of system interface for restaurant model.

the restaurant has the property that corresponds to the position value. Additionally, the restaurant model contains relevant 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 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.

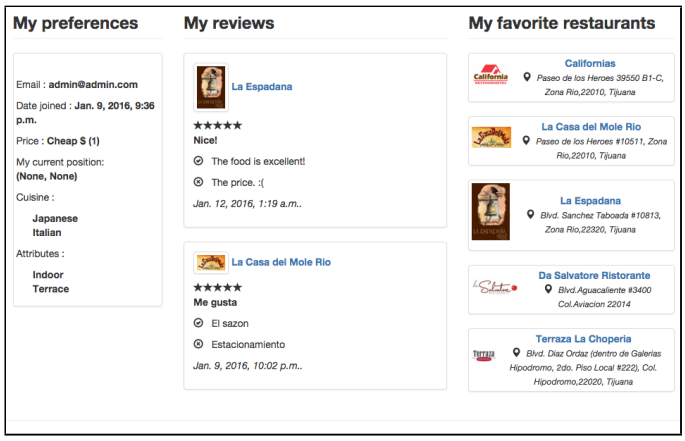
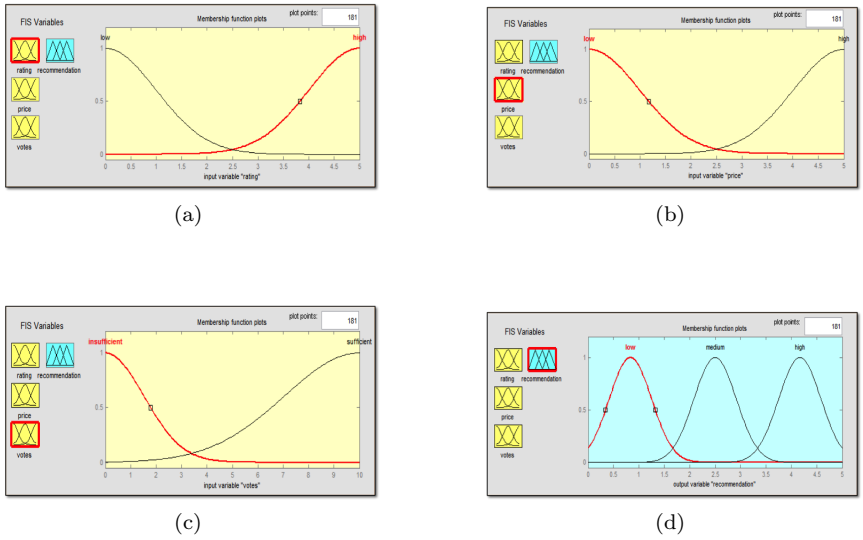


Fig. 2: Example of system interface for user profile.

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 [43].
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 problem.
The FIS proposed has 3 input variables [20]: 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).



(a) (b)

(c) (d)

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

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.*
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

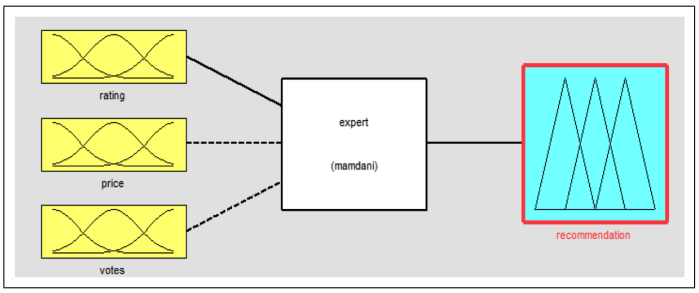


Fig. 4: Fuzzy Inference System of expert.

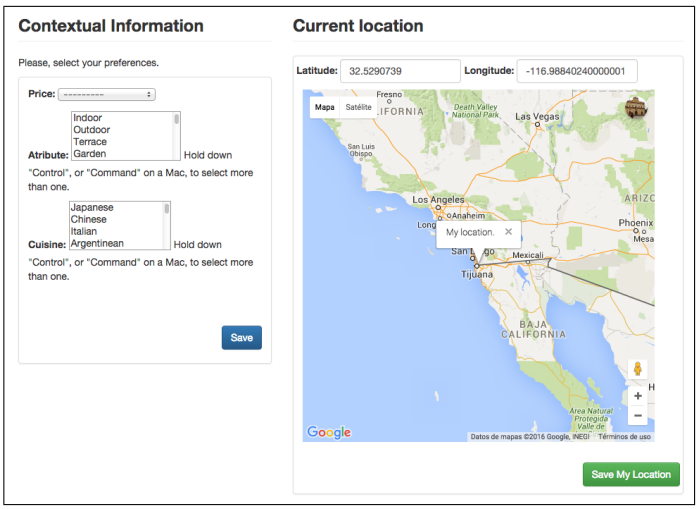


Fig. 5: System interface to collect contextual information.

the user. If context-aware recommender system considers another attributes 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

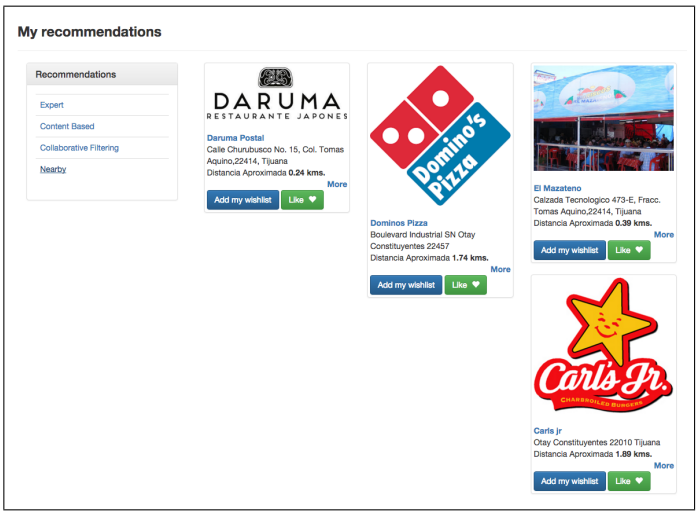


Fig. 6: System inferface of recommendations section.

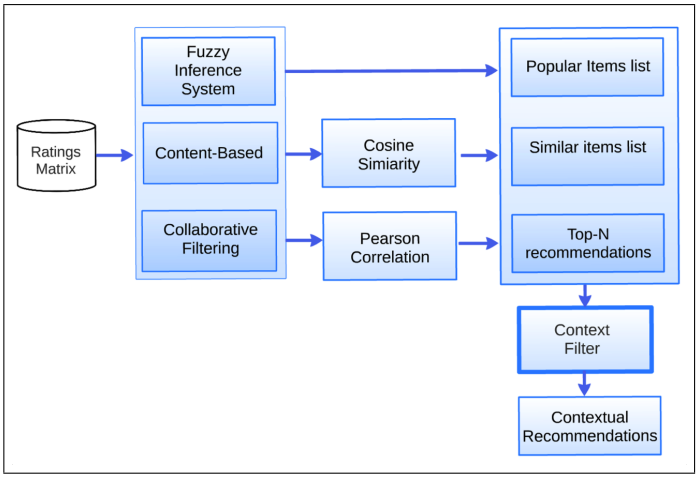


Fig. 7: Architecture proposed.

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).

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 [3].
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

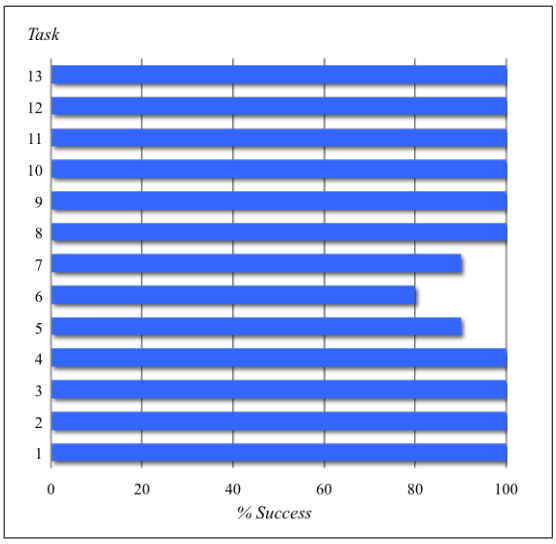


Fig. 8: Representation of the percent of success for each task.

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 [3]. Then, task-on-time was applied to measure time that an user did the task. A 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 [**?**] 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 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

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

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

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.

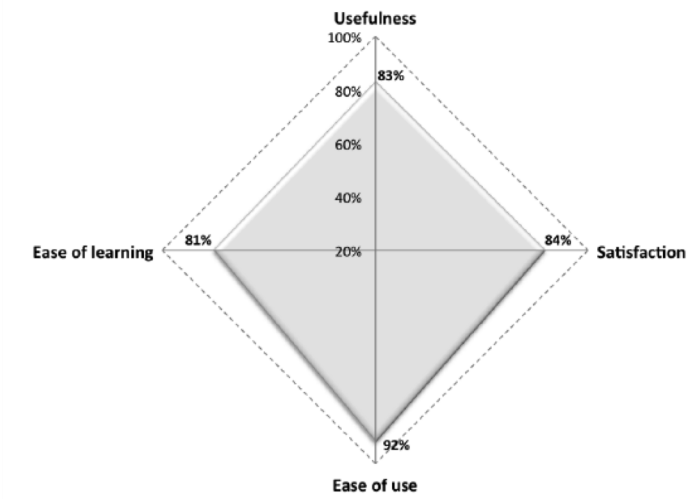


Fig. 9: The radar chart that depicts the four axis evaluated in the questionnaire.

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. Wasfi Al-Khatib, Y Francis Day, Arif Ghafoor, and P Bruce Berra. Semantic modeling and knowledge representation in multimedia databases. *Knowledge and Data Engineering, IEEE Transactions on*, 11(1):64–80, 1999.
3. William Albert and Thomas Tullis. *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. Newnes, 2013.
4. Susan Auty. Consumer choice and segmentation in the restaurant industry. *Service Industries Journal*, 12(3):324–339, 1992.
5. Marko Balabanović and Yoav Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.
6. 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.
7. Linas Baltrunas, Bernd Ludwig, Stefan Peer, and Francesco Ricci. Context-aware places of interest recommendations and explanations. In *Joint Proceedings of the Workshop on Decision Making and Recommendation Acceptance Issues in Recommender Systems (DEMRA 2011) and the 2nd Workshop on User Models for Motivational Systems: The Affective and the Rational Routes to Persuasion (UMMS 2011). CEUR Workshop Proceedings*, volume 740, pages 19–26, 2011.
8. Linas Baltrunas, Bernd Ludwig, Stefan Peer, and Francesco Ricci. Context relevance assessment and exploitation in mobile recommender systems. *Personal and Ubiquitous Computing*, 16(5):507–526, 2012.
9. Mary Bazire and Patrick Brézillon. Understanding context before using it. In *Modeling and using context*, pages 29–40. Springer, 2005.
10. 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.
11. Rebecca Bulander, Michael Decker, Gunther Schiefer, and Bernhard Kölmel. Comparison of different approaches for mobile advertising. In *Mobile Commerce and Services, 2005. WMCS'05. The Second IEEE International Workshop on*, pages 174–182. IEEE, 2005.
12. Robin Burke. Integrating knowledge-based and collaborative-filtering recommender systems. In *Proceedings of the Workshop on AI and Electronic Commerce*, pages 69–72, 1999.
13. Landro Castro and Silvana Aciar. Prototype of a tourism recommender system. In *Informatica (CLEI), 2012 XXXVIII Conferencia Latinoamericana En*, pages 1–7. IEEE, 2012.
14. Federica Cena, Luca Console, Cristina Gena, Anna Goy, Guido Levi, Sonia Modeo, and Ilaria Torre. Integrating heterogeneous adaptation techniques to build a flexible and usable mobile tourist guide. *AI Communications*, 19(4):369–384, 2006.
15. Chung-Hua Chu and Se-Hsien Wu. A chinese restaurant recommendation system based on mobile context-aware services. In *Mobile Data Management (MDM), 2013 IEEE 14th International Conference on*, volume 2, pages 116–118. IEEE, 2013.

16. Mark Claypool, Anuja Gokhale, Tim Miranda, Pavel Murnikov, Dmitry Netes, and Matthew Sartin. Combining content-based and collaborative filters in an online newspaper. In *Proceedings of ACM SIGIR workshop on recommender systems*, volume 60. Citeseer, 1999.
17. Anind K Dey. Understanding and using context. *Personal and ubiquitous computing*, 5(1):4–7, 2001.
18. Eyrun A Eyjolfsdottir, Gaurangi Tilak, and Nan Li. Moviegen: A movie recommendation system. *UC Santa Barbara: Technical Report*, 2010.
19. Gerhard Fischer. Context-aware systems-the right information, at the right time, in the right place, in the right way, to the right person. *AVI 12, Capri Island, Italy*, 2012.
20. 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.
21. Jennifer Golbeck, James Hendler, et al. Filmtrust: Movie recommendations using trust in web-based social networks. In *Proceedings of the IEEE Consumer communications and networking conference*, volume 96, pages 282–286. Citeseer, 2006.
22. 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.
23. Jaksa Jack Kivela. Restaurant marketing: selection and segmentation in hong kong. *International Journal of Contemporary Hospitality Management*, 9(3):116–123, 1997.
24. SooCheong Shawn Jang and Young Namkung. Perceived quality, emotions, and behavioral intentions: Application of an extended mehrabian–russell model to restaurants. *Journal of Business Research*, 62(4):451–460, 2009.
25. Robert C Lewis. Restaurant advertising-appeals and consumers intentions. *Journal of advertising research*, 21(5):69–74, 1981.
26. 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.
27. 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.
28. Tariq Mahmood and Francesco Ricci. Improving recommender systems with adaptive conversational strategies. In *Proceedings of the 20th ACM conference on Hypertext and hypermedia*, pages 73–82. ACM, 2009.
29. Matteo Manca, Ludovico Boratto, and Salvatore Carta. Mining user behavior in a social bookmarking system-a delicious friend recommender system. In *DATA*, pages 331–338, 2014.
30. Luis Martinez, Rosa M Rodriguez, and Macarena Espinilla. Reja: A georeferenced hybrid recommender system for restaurants. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 03*, pages 187–190. IEEE Computer Society, 2009.
31. Frank McSherry and Ilya Mironov. Differentially private recommender systems: building privacy into the net. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 627–636. ACM, 2009.
32. José M Noguera, Manuel J Barranco, Rafael J Segura, and Luis Martínez. A mobile 3d-gis hybrid recommender system for tourism. *Information Sciences*, 215:37–52, 2012.
33. Alvaro Ortigosa, Javier Bravo, Rosa M Carro, and Estefanía Martín. Entornos de aprendizaje móviles adaptativos y evaluación: Comole y geses (adaptive mobile learning environments and evaluation: Comole and geses). *Revista Iberoamericana de Educación a Distancia*, 13(2):167, 2010.
34. Bing Pan and Daniel R Fesenmaier. Online information search: vacation planning process. *Annals of Tourism Research*, 33(3):809–832, 2006.
35. Michael J Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5-6):393–408, 1999.
36. Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.
37. Francesco Ricci. Context-aware music recommender systems: workshop keynote abstract. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 865–866. ACM, 2012.

38. Norma Saiph Savage, Maciej Baranski, Norma Elva Chavez, and Tobias Höllerer. *Im feeling loco: A location based context aware recommendation system.* Springer, 2012.
39. 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.
40. Rossano Schifanella, André Panisson, Cristina Gena, and Giancarlo Ruffo. Mobhinter: epidemic collaborative filtering and self-organization in mobile ad-hoc networks. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 27–34. ACM, 2008.
41. Thomas Tran and Robin Cohen. Hybrid recommender systems for electronic commerce. In *Proc. Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, Technical Report WS-00-04, AAAI Press*, 2000.
42. Jiawei Yao, Jiajun Yao, Rui Yang, and Zhenyu Chen. Product recommendation based on search keywords. In *Web Information Systems and Applications Conference (WISA), 2012 Ninth*, pages 67–70. IEEE, 2012.
43. LA Zedeh. Knowledge representation in fuzzy logic. *Knowledge and Data Engineering, IEEE Transactions on*, 1(1):89–100, 1989.