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

Context-Aware Recommender System Using Post-Filtering And Fuzzy Logic

Xochilt Ramirez-Garcia · **Mario Garcia-Valdez**

Received: date / Accepted: date

Abstract Contextual recommendations are implemented in recommender systems to improve user satisfaction. A recommender system makes accurate and suitable recommendations for a particular situation reaching personalized recommendations. The context provides information relevant to the recommender system and is used as a filter for selection of relevant items for the user. This paper presents a context-aware recommender system, which uses techniques based on collaborative filtering and content-based as well as fuzzy rules to recommend items in context. The dataset used to test the system is Trip Advisor. The accuracy in the recommendations was evaluated with the Mean Absolute Error.

Keywords Algoritms · Content-based · Context · Collaborative filtering · Fuzzy logic · Recommender systems

1 Introduction

Nowadays, Internet provides users convenient tools to improve tasks in daily life in order to know the needs of millions of users, this has disadvantages that leads to overload information. For example, an online store can offer thousands of items in differ ent categories for the user, as result, the user could find in a complex situation, where the items that the user want to buy are contained in a list of thousands of items in the online store. Then, recommender systems are designed for suggest items that are adapted to needs and preferences of users [23]. There are two common techniques for recommendations: 1) collaborative filtering and, 2) content-based. These algorithms can be combined in a hybrid system [11]. The highlight of traditional recommender systems is that considers only two types of entities (users and items) with no taking in account the context. Subsequently, in a way to improve these recommender

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

Mario Garcia-Valdez
mariosky@gmail.com

systems, the efforts are based in the integration of the context in the recommendation process, someway. For example, the use of applications such as recommendation of restaurants [22], sightseeing [6], [4], personalized content on a web page [21], a movie [18], or a song list [3], it can be insufficient to consider only information about users and items, it is important contextual information in the referral process in order to recommend items in specific circumstances or situations [32]. When context is added in an application, context-aware recommender systems meet its function to get suitable items for users, specially when the context information has a high relevance for the domain such as in [4]. For achieve this purpose, researchers propose new techniques such as in [24], [29] and [8] , where a splitting of rating matrix is used to sepate the context and it adjust it in recommendation process. In recent works such as [31] and [7] , the matrix factorization technique have been improved for adding context, getting better results than standard matrix factorization technique [19]. Subsequently, have been developed location-aware recommender system such as in [20] and [17], because of the importance of the location factor when the items are recommended taking in account the user geographical position and the distance of items such as restaurants or some places of interest. The time factor has a high impact in recommender systems due to the changes that happen in a context throuhg the time, such that nowadays time-aware recommender systems [16] are integrated in new mobile applications such as InCarMusic [3], ReRex [5], UbiquiTOUS [9] and Appazaar [10] in order to meet the needs of users.
The content of this paper is presented as follows: section 2 describes the techniques for Recommender Systems, section 3 explains the elements of the method proposed for Context- Aware Recommender System, section 4 describe the settings of development environment, the datasets used to test the Context-Aware Recommender System, section 5 explains the results of experiments to evaluate the system, the main features of the data and finally, the section 6, explains the conclusions and proposals for future work.

2 Recommendation techniques

In recent years, Recommender Systems help to solve the problem of overload information experienced by consumers on the Internet. Recommender Systems working with Collaborative Filtering provided recommendations based on the profile and preferences of users. These Recommender Systems have been tested to be one of the most used techniques to help users find relevant items for them [14], [11].

2.1 Context-aware recommender systems

Context-Aware Recommender Systems (CARS) have been developed to solve the limitations of traditional Recommender Systems that do not consider the context when an item is recommended, for example, to recommend movies specifically for Cristmas week [1]. In literature, the definition of A. Dey [15] about context is widely adopted: *"Context is any information that can be used to characterize the situation* *of an entity. An entityis a person, place, or object that is considered relevant to the interactionbetween a user and an application, including the user and applications themselves."*

A context-aware recommender system considers contexts, e.g. time, place, companion, etc., using paradigms of recommendation [2]such as: 1) Pre-filtering, where the contexts are used for selecting the data before applying the standard CF, 2) Post-filtering, where contexts are incorporated for refine the result of CF, and 3) Context modeling, where the context is directly integrated to the predictions of the users in the items [25]. A traditional Recommender System utilizes to represent the rating the 2-dimensional function: $R : (Users \times Items \rightarrow Rating)$, while CARS, add context for each rating and is represented as a multi-dimensional function:
$R : (Users \times Item \times Context \rightarrow Rating)$.

2.2 Collaborative filtering algorithm

This technique obtains specific recommendations for the user, based in ratings or usage patterns (e.g. shopping), without exogenous information on any of the items or users involved. As was mentioned in introdution, collaborative filtering uses two entities: items and users, but previous to get recommendations for users collaborative filtering gets similarity among users. In order to calculate this similarity two approaches are proposed: 1) the nearest neighbors and, 2) latent factor model. The nearest neighbors approach is based in the relation between items, or users. The latent factor model, such as a matrix factorization comprises an alternative approach by transforming both elements and the users with the same latent factor space. The latent space tries to explain the ratings through user and item characterizations, about the inferred factors automatically of the user feedback [23].

2.3 Content-based algorithm

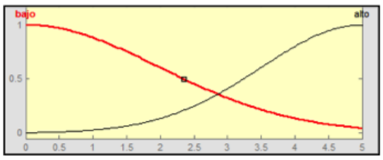
The system learns to recommend items that were relevant for the user in the past. The similarity of items is based on the features associated with the items compared. For example, if a user liked a comedy film, then the system can learn to recommend other comedy films for the same user. This technique analyzes a set of documents and descriptions previously rated by the user and makes a user profile based on the characteristics of the objects rated by that user. The recommendation process basically consists of comparing the attributes of the user profile and the attributes of an object, if these attributes are in the user profile; this is an advantage to process information. For example, it could be used to filter search results to determine if a user is interested or not in a specific Web page, if user do not like, it is possible to prevent data from that page are displayed [23].

2.4 Recommendation by popularity

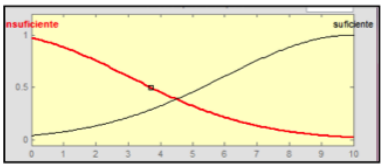
To make recommendations by popularity means consider the popularity of an item in the users community [25], [13], i.e., how many people have evaluated a single item. Prediction by popularity works on the principle that an item is popular because has been rated by many people, it means that can be more informative and relevant within the space of items.

3 Proposed method

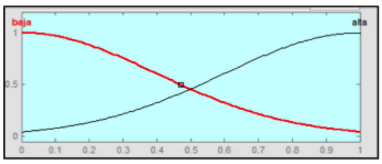
The proposed method consists of three algorithms to recommend: Fuzzy Inference System (FIS), collaborative filtering and content-based. Each one uses rating matrix to get recommendations. The contetx-aware recommender system uses the post-filtering paradigm [2] for adjust recommendations into the context. The recommendation by popularity is through the FIS, the FIS contains the variables that are probably involved in a process to make a suggestion in a human interaction, this process is the same that the recommender system does. The output represents how matter each item into the users community, i.e. if it was a popular item for users. The FIS uses fuzzy rules to infer the inputs and output (numeric value) that represents the weight of the recommendation. The FIS has Gaussians membership functions and are depicted in Fig.1.



(a)



(b)



(c)

Fig. 1: Gaussian Membership functions in the input are: a) RatingAverage, b) UserParticipation, and an output: c) Recommendation.

The fuzzy rules in the FIS (Fig.2) are:

1. `If RatingAverage is low and UserParticipation is insufficient then recommendation is low.`

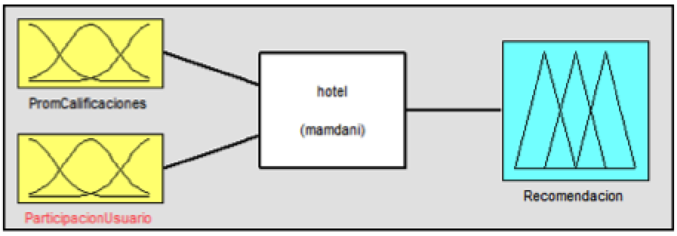


Fig. 2: Fuzzy Inference System.

2. `If RatingAverage is low and UserParticipation is sufficient then recommendation is high.`
3. `If RatingAverage is high and UserParticipation is insufficient then recommendation is low.`
4. `If RatingAverage is high and UserParticipation is sufficient then recommendation is high.`

Content-based algorithm uses cosine similarity to compare the binary vectors representing the profile of each item, thereby obtaining a numerical value that determines similarity, based on a threshold. In other words, it makes a comparison of profiles of each item to determine the *most similar* to items the user has rated with highest score, context-aware recommender system proposed has a scale from 1.0 to 5.0. The formula of cosine similarity used is depicted in equation (1).

$$Cos(\mathbf{I_x},\mathbf{I_y}) = \frac{\mathbf{I_x}\cdot\mathbf{I_y}}{\parallel \mathbf{I_x} \parallel_2 \cdot \parallel \mathbf{I_y} \parallel_2} = \frac{\sum_{i=1}^{n} I_{i,x} I_{i,y}}{\sqrt{\sum_{i=1}^{n} I_{i,x}^2}\sqrt{\sum_{i=1}^{n} I_{i,y}^2}} \tag{1}$$

Where *CosSim(x,y)* represents the cosine of the angle between the two vectors (profiles) of the items being compared. Collaborative filtering algorithm obtains a user neighborhood (KNN-nearest neighbors) and uses the Pearson correlation for the similarity between neighboring users and the current user, the *more similar* users will be providing information for the user prediction.The formula of Pearson correlation [23] is depicted in equation 2.

$$P(U_x,U_y) = \frac{\sum_{i=1}^{n}(U_{xi}-\overline{U})^2(U_{yi}-\overline{U})^2}{\sqrt{\sum_{i=1}^{n}(U_{xi}-\overline{U})^2\sum_{i=1}^{n}(U_{yi}-\overline{U})^2}} \tag{2}$$

Where $P(U_x,U_y)$ is the correlation coefficient from -1.0 to 1.0, users with the highest positive correlation are considered *more similar*. Then, the process when collaborative filtering makes predictions [26] of an item *i* for the user *u*, uses equation 3.

$$P(u,i) = \frac{\sum_{i=1}^{n}(S_{j,u} * C_{j,i})}{\sum_{i=1}^{n}(S_{j,u})} \tag{3}$$

Table 1: A rating matrix example.

Users	Item1	Item2	Item3	Item4
User1	5	4	5	3
User2	4	2	4	5
User3	5	3	5	Null

Where *P(u,i)* is the prediction for the item *i* for the user $u, S_{j,u}$ represents the similarity of the user $j \in J$, and $C_{j,i}$ is the rating of the user *j* for the item *i*. A graphic example of a rating matrix used to predict the rating is depicted in Table 1.

In this example, context-aware recommender system makes a prediction for *user3* on *item4*, using the Pearson correlation coefficient to determine if the *user1* is more similar to the *user3* than the *user2*, therefore, uses the profile *user1* (nearest neighbor) to make a prediction of *3.0* in the *item4* for *user3*.

In the next step the outputs of every recommender algorithm is represented by a list of recommended items. Subsequently apply the context filter and context- aware recommender system gets the final contextual recommendations. To apply the context, context-aware recommender system identifies contextual data of the user profile (see Table 2), and compares recommended items to filter those items that are adjusted to the user context. The context filtering is the last step before to get the recommended items. The schema of architecture for context-aware recommender system is depicted in Fig.3.

Table 2: Example of contextual ratings in the user profile.

User profile		
Item1	Rating1	Context1
Item2	Rating2	Context2
Item3	Rating3	Context3

4 Settings

4.1 Datasets

The dataset used to evaluate the algorithm was TripAdvisor in two versions downloaded from [27], this datasets was used in other papers such as [30], [28] to evaluate the performance of context-aware recommender systems. The first dataset contains 4669 contextual ratings 1202 users and 1890 hotels; the second dataset contains 14175 contextual ratings hotels 2731 users and 2269. Data were collected from reviews online in tripadvisor.com. There is only one context: type of trip (family, friends, bussines, romantic and relax).

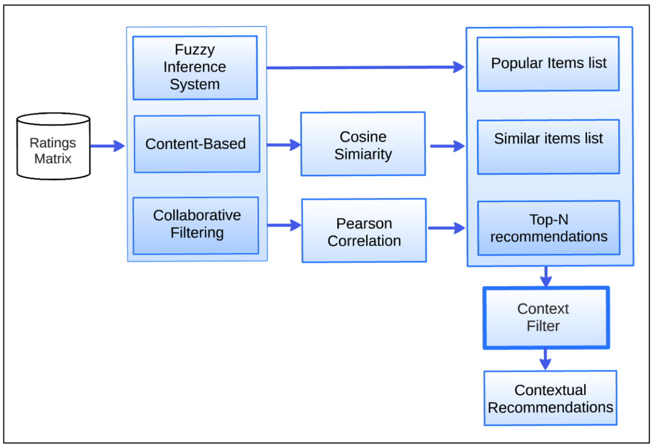


Fig. 3: Recommender system architecture

4.2 Metrics

Recommender Systems are widely used in different domains, therefore, the goals vary for each application, and the evaluation metrics used depend on the general goals of the system. For some applications the prediction accuracy is essential and the evaluation of the system is focused on the accuracy of the recommendations. There are several indicators to measure accuracy: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and, Precision and Recall are some examples [23], [12]. On the other hand, whether the system tries to measure qualitative aspects such as customer satisfaction, recommendations quality or system utility for a user group or community, other metrics are considered such as is mentioned in [12].
The accuracy of prediction is the controversial property in the literature of recommender systems. The core of the most of recommender systems is a prediction engine. This engine can predict the user reviews on the elements (for example of this many researchers have proposed finding algorithms to provide better predictions in this aspect. The prediction accuracy is typically independent of the user interface, which can be measured in an off-line experiment. The precision measuring in a user survey means accuracy measuring given a recommendation. This is a different concept of behavior prediction of users without recommendations, and is more accurate to the accuracy in the real system [23].

5 Results

Two experiments were performed using the dataset described in section 4.1. Table 3 describes the data sets used and the scarcity percentage of the specified data. Scarcity of 99 percent mean that there are problems to recommend items.
By other side, in Table 4 the comparison shows that the algorithm has a acceptable performance, i.e., the error falls into the range of results obtained with others al-

Table 3: Datasets description.

Dataset	Users	Items	Ratings	Scarcity (percent)
TripAdvisor v1	1202	1890	4669	99.79
TripAdvisor v2	2731	2269	14175	99.77

Table 4: Comparison of RMSE.

Dataset	Algorithm	RMSE
TripAdvisor v2	FC + Post-filtering	0.504
	FC	0.994
	Pre-filtering + Relaxation	0.985

Table 5: Level of similarity in datasets.

Dataset	Similarity	Avg.votes per user.
TripAdvisor v1	0.448	5
TripAdvisor v2	0.508	8

gorithms. Then, contextual recommendations were evaluated with the Root Mean Square Error in order to compare the results with context relaxation algorithm [28] that is evaluated with the same dataset.

The fundament of content-based algorithm is the cosine similarity; this means that if similarity value among items is high, the recommendations will improve the degree of user satisfaction. This is observed when calculating the similarity average in each dataset as shown in Table 5.

FIS can provides a list of popular items for each dataset, recommendations through averages are obtained, and recommendations are conditioned to show it when the collaborative filtering and content-based are not delivering recommendations because of data scarcity. However, the majority of popular items of dataset were rated in contexts: romantic, family and bussines, that means that the dataset has biases.

6 Conclusions and future work

In this research a context-aware recommender system is proposed and involves the paradigm of post-filtering for contextual recommendations. The structure of the datasets facilitated the evaluation of recommendations although the rating matrix has been scarce in both cases. Anyway, information of items and users was used to test the system and a good performance of the system was done. With respect the performance, post-filtering allow us to select relevant items that are adjusted into the context, indeed, post-filtering and implementation of different recommendation techniques the system has suitable performance and the datasets help the processes performed. The results have been satisfactory in this work; in the future we are going to apply the context-aware recommender systems in other domains such as e-learning.

References

1. Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *Handheld and ubiquitous computing*, pages 304–307. Springer, 1999.
2. Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer, 2011.
3. 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.
4. 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.
5. Linas Baltrunas, Bernd Ludwig, Stefan Peer, and Francesco Ricci. Context-aware places of interest recommendations for mobile users. In *Design, User Experience, and Usability. Theory, Methods, Tools and Practice*, pages 531–540. Springer, 2011.
6. Linas Baltrunas, Bernd Ludwig, and Francesco Ricci. Context relevance assessment for recommender systems. In *Proceedings of the 16th international conference on Intelligent user interfaces*, pages 287–290. ACM, 2011.
7. Linas Baltrunas, Bernd Ludwig, and Francesco Ricci. Matrix factorization techniques for context aware recommendation. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 301–304. ACM, 2011.
8. Linas Baltrunas and Francesco Ricci. Context-based splitting of item ratings in collaborative filtering. In *Proceedings of the third ACM conference on Recommender systems*, pages 245–248. ACM, 2009.
9. Paolo Bellavista, Antonio Corradi, Mario Fanelli, and Luca Foschini. A survey of context data distribution for mobile ubiquitous systems. *ACM Computing Surveys (CSUR)*, 44(4):24, 2012.
10. Matthias Böhmer, Moritz Prinz, and Gernot Bauer. Contextualizing mobile applications for context-aware recommendation. 2010.
11. Robin Burke. Hybrid web recommender systems. In *The adaptive web*, pages 377–408. Springer, 2007.
12. Elisa Campochiaro, Riccardo Casatta, Paolo Cremonesi, and Roberto Turrin. Do metrics make recommender algorithms? In *Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on*, pages 648–653. IEEE, 2009.
13. Òscar Celma and Perfecto Herrera. A new approach to evaluating novel recommendations. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 179–186. ACM, 2008.
14. Anind K Dey. Understanding and using context. *Personal and ubiquitous computing*, 5(1):4–7, 2001.
15. Gerhard Fischer. Context-aware systems: the'right'information, at the'right'time, in the'right'place, in the'right'way, to the'right'person. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 287–294. ACM, 2012.
16. Ahmed Abdeen Hamed, Rebecca Roose, Marlon Branicki, and Alan Rubin. T-recs: Time-aware twitter-based drug recommender system. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, pages 1027–1031. IEEE Computer Society, 2012.
17. Marius Kaminskas and Francesco Ricci. Location-adapted music recommendation using tags. In *User Modeling, Adaption and Personalization*, pages 183–194. Springer, 2011.
18. Kyung-Rog Kim, Ju-Ho Lee, Jae-Hee Byeon, and Nam-Mee Moon. Recommender system using the movie genre similarity in mobile service. In *Multimedia and Ubiquitous Engineering (MUE), 2010 4th International Conference on*, pages 1–6. IEEE, 2010.
19. Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.
20. Justin J Levandoski, Mohamed Sarwat, Ahmed Eldawy, and Mohamed F Mokbel. Lars: A location-aware recommender system. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 450–461. IEEE, 2012.
21. Luis Martínez Marina, Juan Antonio Calles García, and Estefanía Martín Barroso. Ontology-based web service to recommend spare time activities. In *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, pages 67–70. ACM, 2010.

22. Xochilt Ramirez-Garcia and Mario García-Valdez. Post-filtering for a restaurant context-aware recommender system. In *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*, pages 695–707. Springer, 2014.
23. Francesco Ricci, Lior Rokach, and Bracha Shapira. *Introduction to recommender systems handbook*. Springer, 2011.
24. Alan Said, Ernesto W De Luca, and Sahin Albayrak. Inferring contextual user profiles-improving recommender performance. In *Proceedings of the 3rd RecSys Workshop on Context-Aware Recommender Systems*, 2011.
25. Guy Shani, Max Chickering, and Christopher Meek. Mining recommendations from the web. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 35–42. ACM, 2008.
26. Guy Shani and Asela Gunawardana. Evaluating recommendation systems. In *Recommender systems handbook*, pages 257–297. Springer, 2011.
27. Yong Zheng. Context-aware datasets. url: http://students.depaul.edu/ yzheng8/datasets.html. 2015.
28. Yong Zheng, Robin Burke, and Bamshad Mobasher. *Differential context relaxation for context-aware travel recommendation*. Springer, 2012.
29. Yong Zheng, Robin Burke, and Bamshad Mobasher. Splitting approaches for context-aware recommendation: An empirical study. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pages 274–279. ACM, 2014.
30. Yong Zheng, Bamshad Mobasher, and Robin Burke. Context recommendation using multi-label classification. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*, volume 2, pages 288–295. IEEE, 2014.
31. Yong Zheng, Bamshad Mobasher, and Robin Burke. Correlation-based context-aware matrix factorization. 2015.
32. A Zimmermann, M Specht, and A Lorentz. Personalization and context-management user modeling and user-adapted interaction. *Journal of Personalization Research (UMUUAI)*, 2000.