Context Personalization for Context-Aware Recommender Systems

Aram B. González and Jorge A. Ramírez Uresti

Departamento de Tecnologías de Información y Computación
Campus Estado de México del Tecnológico de Monterrey
Carretera al Lago de Guadalupe Km. 3-5, Atizapán, Estado de México, 52926, México
{a00965707,juresti}@itesm.mx

Abstract. Current Context-Aware Recommender Systems (CARS) manage context variables as a rigid set of environment, user, and item characteristics used to improve the recommendation process. Current CARS also assume that the context variables must always be present and are equally relevant among all the users. In this paper, we analyze the possibility of a flexible context model for CARS that allows individual personalization through context priorities.

Keywords: Recommender Systems, Context-Aware Recommender Systems, Personalization

1 Introduction

With the internet era, the amount of information and its availability have become, in some way, both a blessing and a liability. Information sources can be easily found through the web, for this reason, data can literally be obtained in just a few seconds. However, the users often become overwhelmed by the quantity of possibilities, and not all of them contain what they are looking for. For these reasons, Information Retrieval and Recommender Systems (RS) have both become popular research areas in the recent years. A RS can be defined as a group of tools and techniques that work together in order to present the user a set of suggested items. The suggested items are not tied to a specific subject so they are found in different domains. RS are usually created to help users evaluate and choose from a baffling number of possible choices.

Context inclusion has become one of the prominent research areas for RS [1], even having international challenges [2]. Traditional RS involve a rating function depending on two factors: users and items. CARS include diverse variables inside the rating function, some common examples include time, location, temperature, companions, etc. Research done in this area validates the inclusion of context variables to improve the recommendation process [3]. Since most of the current research focuses in showing the usefulness of including context variables, they tend to ignore the following considerations:

- Current CARS implementations include context variables as a fixed set of variables, which must be included in each of the cases used for the rating function [4, 5]. Since the rating function uses a rigid set of variables, there is no consideration of what may happen if those variables are missing when calculating the rating or how to handle possible new variables.

- CARS assume that their context variables have the same relevance for all the users. An example in a movie recommendation system: there may be a user who prioritizes the location of the movie theater while putting aside everything else. Meanwhile, another user may consider location the least relevant factor. This case is not considered by current CARS, missing a new level of personalization that may improve their recommendation.

2 Research objective

Development of a Context-Aware Recommender System (CARS) with a flexible number of context variables and individual personalization by prioritizing available context variables.

The main contributions of this proposal are:

1. A flexible context model.
2. Further individual personalization by sorting context relevance for each user.
3. A flexible rating function that can deal with a variable number of contexts.

3 Current progress

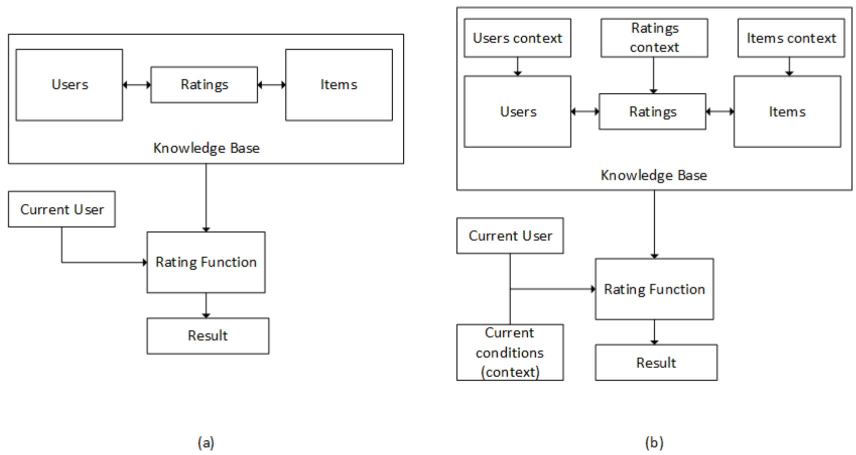

Fig. 1. *(a)*: Traditional Recommender System structure. *(b)*: Context Aware Recommender System structure.

The simplest representation of a RS is based on its rating function is shown in Fig. 1(a), where the RS usually receives a user request so it must generate a list of recommended items based on previous knowledge and the users profile. In Fig. 1(b), the structure of a Context-Aware Recommender System which also includes the representation of the types of context present in the process. The parts of a Recommender System are:

- **Knowledge base** - Composed of recorded cases of users, items and their corresponding ratings.
  - *Users* - Users included inside the knowledge base have already rated some of the items.
  - *Items* - It is the accepted term to denote the recommendation objective. A RS commonly focuses on a single and specific type of item. The knowledge base usually only includes items that have been previously rated by at least one user.
  - *Rating* - It is the user point of view of an item, it is often quantified.
- **Current user** - The user that is currently requesting a recommendation from the system, a user model is often used as the input.
- **Rating function** - This is the main part of a RS, it is the function that allows the system to generate a list of recommendations, it takes the previous cases (knowledge base) and the current user profile as inputs to create it. Collaborative filtering techniques are commonly used here.
- **Result** - This is the result of the recommendation process, it can be an item or a list of items that the user might find useful according to the knowledge processed through the rating function.

Context variables can be added in three different zones inside a RS, those are: *a*) *User context*: Some examples are: age, gender, education, relationships (friends, family, coworkers, etc.), marital status, etc. *b*) *Item context*: Some examples are: materials, country, quantity, genre, price, etc. *c*) *Rating context*: The rating context represents the circumstances in which the rating of a product was done. Some examples are: date, location, weather, etc.

The rating function can be seen as follows [6]:

$$R : User \times Item \rightarrow Rating \tag{1}$$

Where the combination of a User and an Item has an associated Rating. When including context variables, the function changes to:

$$R : User \times Item \times Context \rightarrow Rating \tag{2}$$

Where Context often is a fixed set of contextual variables:

$$Context : \{C_1, C_2, C_3, ...C_n\} \tag{3}$$

We propose changing the *Context* set for a data structure that allows us to organize and prioritize it depending on the users interests. Fig. 2 shows a reduced example of what the time contextual variable may look by replacing its original value in the *Context* set.

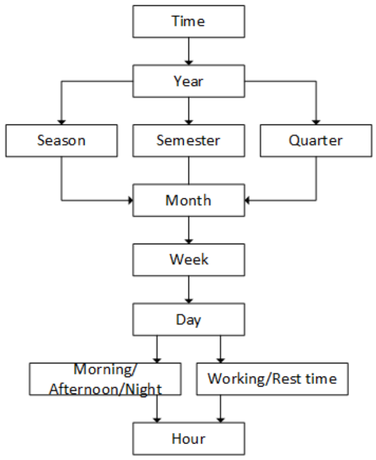

Fig. 2. Proposed structure for time variable.

The structure proposed in Fig. 2 can be used to organize and prioritize the contextual values for a single user by acting as a model to represent users, items and ratings contexts. It is designed to be flexible so that new contexts can be included and, if needed, some may be eliminated or changed. A pending issue is the way in which the rating function must evaluate the Context Structure presented and the possible outcomes of the learning progress that will be used.

References

1. Bobadilla, J., Ortega, F., Hernando, A., Gutirrez, A.: Recommender systems survey. Knowledge-Based Systems **46**(0) (2013) 109 – 132
2. Said, A., Berkovsky, S., De Luca, E.W.: Putting things in context: Challenge on context-aware movie recommendation. In: Proceedings of the Workshop on Context-Aware Movie Recommendation. CAMRa '10, New York, NY, USA, ACM (2010) 2–6
3. Shi, Y., Larson, M., Hanjalic, A.: Mining contextual movie similarity with matrix factorization for context-aware recommendation. ACM Trans. Intell. Syst. Technol. **4**(1) (February 2013) 16:1–16:19
4. Gantner, Z., Rendle, S., Schmidt-Thieme, L.: Factorization models for context-/time-aware movie recommendations. In: Proceedings of the Workshop on Context-Aware Movie Recommendation. CAMRa '10, New York, NY, USA, ACM (2010) 14–19
5. Su, J.H., Yeh, H.H., Yu, P., Tseng, V.: Music recommendation using content and context information mining. Intelligent Systems, IEEE **25**(1) (2010) 16–26
6. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In Ricci, F., Rokach, L., Shapira, B., Kantor, P.B., eds.: Recommender Systems Handbook. Springer US (2011) 217–253