A Roadmap for Semantifying Recommender Systems Using Preference Management

Dilek Tapucu[1], Fatih Tekbacak[1], Murat Osman Ünalır[2], Seda Kasap[1]

[1] Izmir Institute of Technology, 35430 Urla, Izmir, Turkey,
[2] Ege University[2],35100 Bornova, Izmir, Turkey.

[1] {dilektapucu, fatihtekbacak,sedakasap}@iyte.edu.tr
[2] murat.osman.unalir@ege.edu.tr

Abstract. The work developed in this paper presents an innovative solution in the field of recommender systems. Our aim is to create integration architecture for improving recommendation effectiveness that obtains user preferences found implicitly in domain knowledge. This approach is divided into four steps. The first step is based on semantifying domain knowledge. In this step, domain ontology will be analyzed. The second step is to define an innovative hybrid recommendation algorithm based upon collaborative filtering and content filtering. The third step is based on preference modeling approach. And in the fourth step preference model and recommendation algorithm will be integrated. Finally, this work will be realized on Netflix movie data source.

Keywords: Recommender System, User Preference, Ontology.

1 Introduction

Preference management has a key role in order to provide effective personalization [1]. Our model proposes associating any preference model to any ontology resource model. It allows to manipulate the ontology model through its meta-model [4]. In this model, the ontology's instances are taken into account by referring to their corresponding ontology's entities.

Recommender systems represent user preferences for the purpose of suggesting items [2]. Effective recommendation includes filtering methods to predict if a suggesting item will please a user or not.

In this paper, an innovative solution in the field of recommender systems is examined. Our contribution is to propose architecture for using our preference model to represent user preferences. Furthermore, proposed architecture has been integrated with recommendation algorithm(s) in the literature. The main goal of our work consists of improving recommendation effectiveness. The remainder of this paper is organized as follows. In Section 2, we give the domain ontology analysis. In Section 3, we summarize existing work related with Recommender Systems. In Section 4, main feature of the Preference Modeling approach is given. In Section 5, we explain proposed architecture. Then conclusion and future work is determined.

E. Gelenbe et al. (eds.), *Computer and Information Sciences*, Lecture Notes in Electrical Engineering 62, DOI 10.1007/978-90-481-9794-1_20,

2 Domain Ontology Analysis

To obtain properties of ontology related individuals of domain classes, domain ontology should be analyzed. Users that log in the system can choose their preference attributes and fill them appropriately. Meanwhile, reusable design strategy has to be thought for an adaptive recommendation approach. So that changing domain ontology can't affect the whole system while searching the ontology in granular way. Then different domain ontologies can be loaded and analyzed by the system using stored preferences of users.

3 Recommender Systems

This section presents a general review of recommender system literature. These systems act as personalized decision guides, aiding users in decisions on matters related to personal taste [5]. The key concept to develop an efficient recommender system is better understanding of both users and items. However, traditional recommender systems consider limited data (ratings, keywords) to compute predictions and do not take into account different factors necessary to understand reasons behind a user's judgment. Actual Recommender Systems can be divided in three categories as Content Based (CB), Collaborative (CF) and Hybrid recommender Systems [2], [5], [6].

4 Preference Meta-Model

In this section we present an ontology-based preference modeling approach. This approach includes three model elements: **Ontology Model Resource**: In order to attach preferences to instances of a given ontology, we need to model semantic resource definitions. In this approach [4], the `Property_Or_Class` resource is used to represent data elements of the ontology model, and `Property_Or_Class_Instance` is defined as an instance of that resource definition. Notice that this resource can be defined in all existing ontology model. `Preference` and `Property_Or_Class` resources are composed to `Pref_link` for combining each other. **Preference Model:** The definition of our preference model compiles the different types of preferences (*boolean, interval, numeric etc.*) found in the literature [4]. **Preference Link**: The last part of model consists of a link establishing an association between the preference of the preference model and the class or property definition of the ontology model.

5 Integration Architecture

In this section, we propose why we need semantifying recommendation system to obtain user preferences. In our application, user's preferences are found implicitly in domain knowledge. The corresponding system architecture is a multi-tier application with a *front-end* and a *back-end* level. The *front-end* level includes all those modules that implement the communication between the user and system, while the *back-end* level refers to all those modules that implement the recommendation mechanism. This architecture is clarified in two steps that is explained below:

Analysis and Design: Increasing number of Recommender System approaches are based on text-based documents or database entries. By the need of accessing personal data around the Internet, each user wants to have an identifier document that includes his/her preferred knowledge. With the analysis of this problem, FOAF documents[1] have been determined as an appropriate solution. Recommender Systems use generally predefined algorithms in the literature and the hybrid solutions using them as told in Section 3. To measure performance of these algorithms, a flexible, fast collaborative filtering engine named as Apache Mahout Taste have been used in this project[2]. This project helps us to predict rating values for the user interests. To test our system, we firstly divided our Netflix[3] dataset in two pieces. After doing this process, usage of a semantic web enabled inference engine and a programming environment named Jena[4] has been decided.

Implementation Layer: In Figure 1, our implementation architecture is described with a *front-end* and a *back-end* level in a detailed manner:

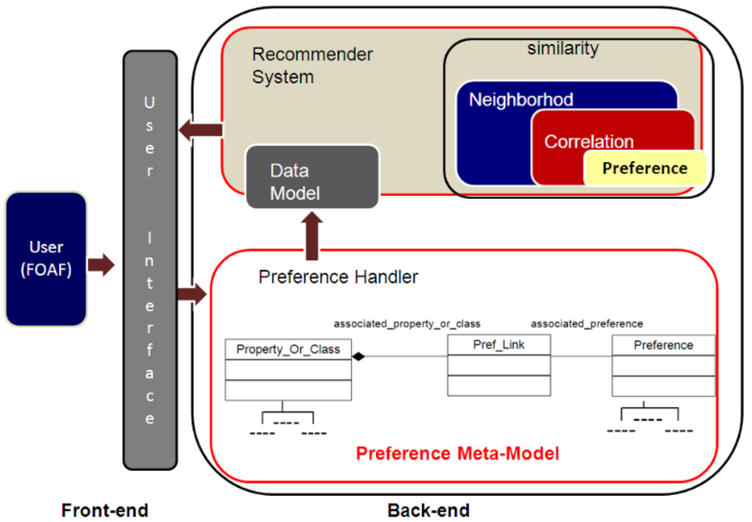

Fig. 1. System Architecture.

[1] http://www.foaf-projects.org/
[2] http://lucene.apache.org/mahout/taste.html
[3] http://www.netflix.com/Default
[4] http://jena.sourceforge.net/

Front-end level consists of;

- *The User Interface Module*: It is responsible for communicating between the users and system. This interface is used for users' login and new user registration. It obtains rating from the users and provides recommendation to the active user. It has two areas: 1. *Test interface:* Developer is able to see the performance of the algorithm on the previously collected "test" dataset. 2. *Real Time Recommendations*: A new user is able to create a FOAF document and rate a small set of M movies.

Back-end level includes;

– *The Preference Handler Module*: It is composed of preferences identified in the literature that is attached to domain ontology [3, 4]. In this paper `Numeric_Preference` type definition which is interpreted by *numeric value* is used. This type of preference is specified with two attributes; `number_value` is a defined type based on integer values, `pref_attributes` associates a `Numeric_Preference` with a list of `Preference_URI` identifier.

– *Recommender System Module:* This module uses the required preference type as an input. The final output of a recommender module is a set of recommended item for a user. The dataset for our study is a subset of the Netflix[4] collection.

6 Conclusion

In this paper, we proposed integration architecture for improving recommendation effectiveness. While movie characteristics are ignored, user preferences are used implicitly in domain knowledge. For this view, rating value is stored as a numeric preference. This work will be continued in three main directions: a), we plan to extend our example (*After Jack watched fantastic movie that is called AVATAR, he recommends it to John*), b) we want to use CB to show social net and CF for semantifying domain knowledge, c) we will try to discuss social search approach.

References

1. Toninelli, A., Corradi, A., Montanari R..: Semantic-based discovery to support mobile context-aware service Access. Computer Communications archive, **31(5)**, 935--949, (2008)
2. Resnick, P. and Varian, H. R.: Recommender Systems, Communications of the ACM, **40** (3), 56-58 (1997)
3. Tapucu, D., Can, Ö., Bursa O., Ünalir, M. O.: Metamodeling Approach to Preference Management in the Semantic Web", IAAA, M-PREF2008 USA, 116--123, (2008)
4. Tapucu, D., Jean S., Aït-Ameur Y., and Ünalir M. O., An Extention of Ontology Based Databases to Handle Preferences, ICEIS 2009, Milan, Italy, 208--214, (2009)
5. Lampropoulou P.S., Lampropoulos A.S., and Tsihrintzis G.A., A Mobile Music Recommender System based on a Two-level Genre-Rating SVM Classifier enhanced by Collaborative Filtering, Interactive Multimedia Sys., SCI 226, 361–-368, (2009)
6. Terveen, L. and Hill, W: 2001, Human-Computer Collaboration in Recommender Systems. In: J. Carroll (ed.): Human Computer Interaction in the New Millenium, (2001)