A Hybrid Recommender System Architecture for Learning Objects

Mario García-Valdez and Brunett Parra

Tijuana Institute of Technology, Tijuana México
rparra_galaviz@hotmail.com, mariog@tectijuana.mx

Abstract. In this paper we present the architecture of a hybrid recommender system to support an adaptive hypermedia educational (AHE) system. Currently the instructor (using fuzzy rules) specifies the sequence in which learning objects are presented to students. The instructor can also give students a chance to choose from a pool of objects and helps them make their selection by assigning to each object a recommendation rating based on the student's profile. We propose a hybrid recommender system that uses collaborative filtering techniques together with fuzzy inference systems to provide recommendations, considering the instructor's experience as well as the ratings given by similar students.

1 Introduction

The goal of Adaptive Hypermedia systems (AH) is to enhance the functionality of hypermedia, tailoring to each user's needs the navigation and presentation of resources [7]. In a previous work an Adaptive Hypermedia Educational (AHE) system based on learning objects was proposed by the authors [1]. Learning objects in this context are reusable web based resources (i.e. a web page, a video or images) that support a certain learning activity, these resources are authored as components, so they can be combined with others. In a course each unit of instruction can be supported by many learning objects, and instructors normally define the sequence in which these learning objects are going to be presented to students. In this current implementation of the system, instructors can personalize to each student, the sequencing and selection of learning objects using a rule-based sequencing model based on the Simple Sequencing specification [8]. Instructors can specify rules that give permission to students so they can choose which objects they want to visit. In this paper we present a preliminary design of a hybrid recommender system to help students make their selections. The proposed recommender systems consider a personalized rating given to the learning object by the instructor and also the ratings given by other students. The aim of this paper is to present an overview of the main components of the recommendation process and the algorithms used. In section 1 a brief overview of the field of recommender systems is presented; a more in-depth survey can be found in [2]. An overview of

O. Castillo et al. (Eds.): Evolutionary Design of Intelligent Systems, SCI 257, pp. 205–211.
springerlink.com

the proposed recommender system is given in section 2, and in section 3 the details of the recommender algorithm are presented. Finally some conclusions are presented in section 4.

2 Recommender Systems

Recommender systems (RS) help users deal with information overload by providing personalized recommendations of content and services [2]. These systems are used by commercial websites and e-marketing tools to increase sales, by presenting to users those products that they more probably want to buy. The majority of RS use a *collaborative filtering* approach, which is the method of making automatic predictions (filtering) about the interests of a user by collecting information on the tastes of many users (collaborative) and also preferences he liked in the past. For example, a collaborative filtering system of musical taste, can make predictions about which music a user would want, given a partial list of preferences of other users and his own. Other RS use a *content-based* approach where items recommended to the user are similar to those the user liked in the past. Other RS use a hybrid approach. Formally the recommendation problem can be stated as follows [2]: Let *C* be the set of all users, and *S* the set of items that can be recommended, both sets can be very large. Let u be the utility function that measures the usefulness of item *s* to user u. i.e. u: $C \times S \rightarrow R$ where R is a totally ordered set. Then for each $c \in C$, we want to choose an item $s' \in S$ that maximizes the user's utility:

$$\forall c \in C, \quad s_c{}' = arg\max_{s \in S} u(c, s)$$

Usually u is represented by a rating in a recommender system and is only defined for a subset of the $C \times S$ space, because not all users give a rating to all the items. A rating matrix of $C \times S$ has the ratings of items indicated by users, and to indicate that a user has not rated an item the symbol "$\emptyset$" can be used. The RS engine should be able to estimate these missing combinations and issue recommendations based in these predictions.

3 Recommender System Architecture

In this section the proposed recommender system (RS) is presented. This system is used in an Adaptive Hypermedia Educational system, in which students must complete certain learning activities previously specified by the instructor. These learning activities can have a recommended value based on the student's profile which has information about the student's performance and learning style. Instructors specify their recommendations using a Mamdani fuzzy inference system with rules which have a recommended fuzzy recommended value as their consequent:

```
IF Visual IS Strong AND Verbal IS Mild THEN
      Recommended IS Low
IF Visual IS Mild AND Verbal IS Strong THEN
      Recommended IS High
```

These rules are static, as the instructor has defined the membership functions i.e. `Strong`, `Medium` and `Mild` corresponding to `Visual` and `Verbal` linguistic variables. Learning activities are multimedia resources (i.e. video, text or audio) in this example a learning activity presented in text format has a higher recommended value for students with a strong verbal learning style. These heuristic based recommendations rely on the instructor's subjective appreciations about the preferences of students. One way to add an adaptive behavior to the system is by changing the parameters of membership functions in response to student's feedback. In this paper another approach is explored, adding another recommendation value to learning activities, this time the recommendation is given by a collaborative filtering algorithm. Now each learning activity has a recommended value that takes into account the instructor and also their peers' ratings. In this algorithm the same features as the heuristic recommendation are considered:

- **Learning style.** The learning style is previously assessed by a test, which gives students a grade from 0 to 20 in each of the learning styles (visual, verbal, aural, physical, logical, social, solitary).
- **Student Performance.** Learning activities record the performance of students, and the percentage of objectives reached by students.

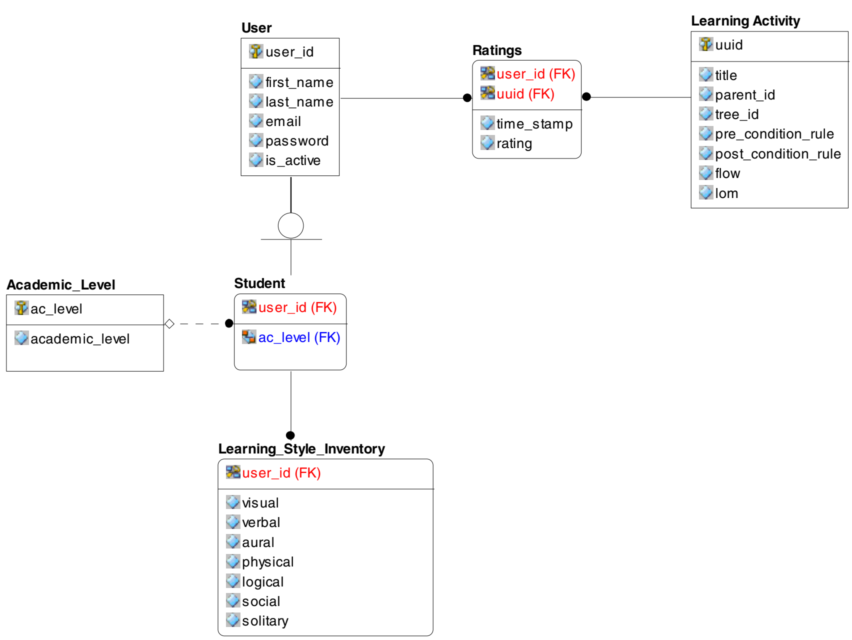

Fig. 1. Recommender System Data Model

Each student is given an opportunity to rate each learning activity they complete, giving it a rating from 1 to 5. These ratings are also considered by the collaborative filtering algorithms. The data model of the recommender system is illustrated in Figure 1.

Each student is represented by a vector of features in this case their learning style inventory; there is also a rating matrix with the ratings of all users. These two sets of data are used separately by two k-NN algorithms, one based on Euclidean distance for selecting students with a similar profile, and another based on Pearson Correlation to find students with similar preferences as there is a positive correlation between their ratings. The recommender algorithm, considers three special cases:

1. **A new student is added.** In order to make accurate recommendations collaborative filtering algorithms need to know the student's previous preferences, based on values of the rating matrix. When a new user is added to the system or he has not made certain number of ratings, there is not enough rating data to give an accurate recommendation. If a student is "new" in the system then the instructor's recommendation value and the ratings of similar students with respect to their learning styles is considered. The level of newness can be seen as a fuzzy variable based on the minimum number of ratings needed to make accurate predictions. This value can be estimated on-line.
2. **A new learning activity is added.** As in the previous case, there is also a problem when a new learning activity is added to the system, as again it does not have enough ratings. Each learning activity has standard metadata [3] indicating among other things: their intended audience, difficulty level, format, authors and version. This information can be used to make a content-based recommendation, when a new learning activity is added to the system, also the instructors recommendation is considered. As the learning activity is receiving ratings, the collaborative filtering recommendations become more accurate and gain more weight in the overall recommendation.
3. **Sparse rating matrixes.** This is a typical problem in other recommender systems and happens when there are a high number of users and items, and users only rate a small fraction of the items. In this case, students must rate learning activities as they finish them, but they only rate those learning activities in their path. This is illustrated in Figure 2, where only Student 1 and Student 2 rated the same learning activities, in this particular application the rating matrix not only has which students rated the same learning activities, but to some extent; also which students followed similar paths. As an option the proposed recommender system can also consider the performance evaluation of the student as an indirect rating value.

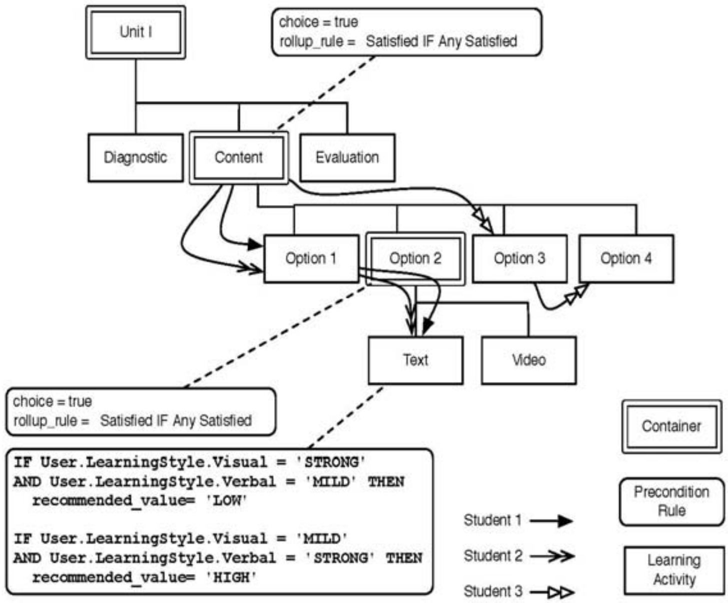

Fig. 2. Paths of different students

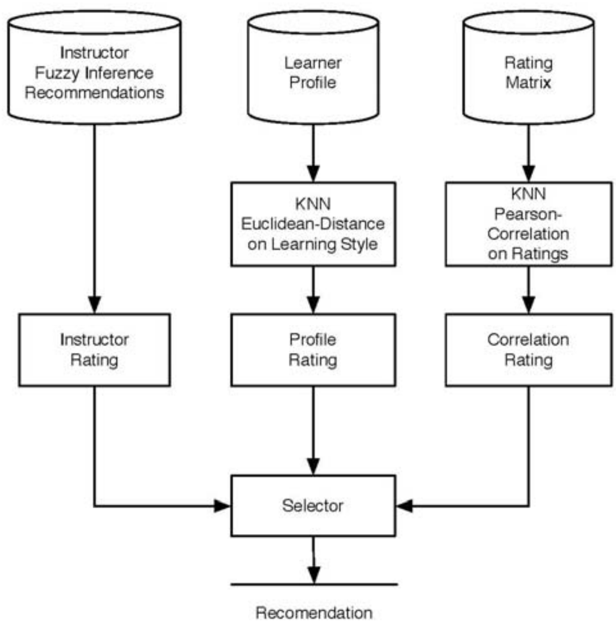

Fig. 3. Recommender System Process

An overview of the recommender process is illustrated in Figure 3, there are three different predictions of ratings: one that considers the instructor's rule-based system, other that considers the similarity of students regarding their learning styles and finally one that considers the correlation between student's ratings. These three predictions are then integrated by a fuzzy inference system which gives the final recommendation value.

4 Memory-Based RecommenderAlgorithm

Memory-Based algorithms have been proposed [4] for collaborative filtering. The objective of these algorithms is to predict the vote (rating) a particular active user is going to give to items, based on a sample or population of the voting of other users. Each of these votes are represented by v_{ij} which is the vote of user i on item j. This algorithm considers the mean vote of other users $\bar{v}_i$ and the active user $\bar{v}_a$ and assumes that the predicted vote $p_{a,j}$ of active user a to item j, is the weighted sum of the votes of other users:

$$p_{a,j} = \bar{v}_a + k \sum_{i=1}^{n} w(a,i)(v_{ij} - \bar{v}_i)$$

Where n is the number of users with non zero weights. Here the weights $w(a,i)$ correspond to the similarity between the active user a and user i (i.e. Pearson's correlation or Euclidean distance), k is a normalization factor such that the absolute value of the weights sum to unity. In this particular implementation the number of users considered are the K-Nearest Neighbors, and the weight function used are: Euclidean distance for the Profile Rating and Pearson's for the Correlation Rating. These ratings are then integrated by a Fuzzy Inference system that considers the number of items and ratings (given or received) to determine the newness of the user or learning activity, with this information a final recommendation value is finally assigned to the Learning Object.

5 Conclusions

In this paper we have presented the design of a hybrid recommender system, considering some of the problems associated with this type of systems, multiple recommendations are computed and then a selector module chooses the appropriate value for the intended item. Fuzzy inference is used for making heuristic recommendations and for the final selection. The implementation is still work in progress, and refinements can be made as a result of further experiments.

References

[1] Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, pp. 43–52 (1998)

[2] Sarwar, B., Karypis, G., Konstan, J.: Analysis of recommendation algorithms for e-commerce. In: Proceedings of ACM E-Commerce (2000)
[3] Memletics, Memletics® Learning Styles Inventory (2008), http://www.learning-styles-online.com/inventory/
[4] García Valdez, J.M.: Aprendizaje colaborativo basado en recursos adaptativos (2008)
[5] Adomavicius, G., Tuzhilin, A.: Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Trans. Knowl. Data Eng. 17(6), 734–749 (2005)
[6] Delgado, J., Ishii, N.: Memory-Based Weighted-Majority Prediction for Recommender Systems. In: ACM SIGIR 1999 Workshop on Recommender Systems: Algorithms and Evaluation (1999)
[7] Brusilovsky, P.: Methods and techniques of adaptive hypermedia. User Modeling and User-Adapted Interaction 6(2-3), 88–129
[8] IMS Global Learning Consortium, I. IMS Simple Sequencing Information and Behavior Model (Version 1.0 Final Specification) (2003b)