A Model for a Collaborative Recommender System for Multimedia Learning Material

Nelson Baloian[1], Patricio Galdames[1,2], César A. Collazos[3], and Luis A. Guerrero[1]

[1] Department of Computer Science
Universidad de Chile
Blanco Encalada 2120, Santiago, Chile
{nbaloian,pgaldame,luguerre}@dcc.uchile.cl
[2] Department of Electrical Engineering
Universidad de Chile
Tupper 2007, Santiago, Chile
[3] Department of Systems
Universidad del Cauca
FIET-Sector Tulcan, Popayán, Colombia
{ccollazo}@unicauca.edu.co

Abstract. In a cluster of many servers containing heterogeneous multimedia learning material and serving users with different backgrounds (e.g. language, interests, previous knowledge, hardware and connectivity) it may be difficult for the learners to find a piece of material which fit their needs. This is the case of the COLDEX project. Recommender systems have been used to help people sift through all the available information to find that most valuable to them. We propose a recommender system, which suggest multimedia learning material based on the learner's background preferences as well as the available hardware and software that he/she has.

1 Introduction

As the amount of available information in the world increases, recommender systems are becoming more important to help us receive that information which is more important to us. Recomender systems may be based on content analysis or on collaborative filtering. In the first case the content of a document is usually analyzed automatically to extract its relevant characteristics by the application of different metrics. The characteristics are then compared with those desired. In the second case, these systems base their decisions on opinions given by users that previously. Both types of systems use user-modeling methodologies. Some techniques, which have been successfully used to accomplish this task, are Bayesian networks with Markov chains [15] and similarity metrics de [14]. In a computer supported learning environment consisting of many servers located in different parts of the world containing rapidly changing learning material of very heterogeneous characteristics a user will certainly have some problems locating suitable learning material for him/her. Moreover, if the learners have very different backgrounds, learning needs, equipment, and connectivity a recommender system may be a very useful tool. This is exactly the case of the COLDEX [6] project, which aims to develop distributed collaborative learning envi-

G.-J. de Vreede et al. (Eds.): CRIWG 2004, LNCS 3198, pp. 281–288, 2004.

ronments by remote distributed experimentation for learning communities in Europe and South America. The COLDEX network consists of interconnected networks of server nodes, each one supporting a particular learning community. Experimental equipment however maybe distributed across the entire network. For example, there are telescopes in Chile and Spain, Seismographs in Chile, and a greenhouse in Sweden, all them connected to the network. Appropriate software provides interfaces for collaboratively work and learn with them.

In this paper we present a preliminary work which aims to develop a recommender system for the above-described environment. Since the learning material consists mainly in learning objects of different multimedia nature, an automatic analysis of the content is not practicable. Therefore human experts do a first characterization of the learning objects of the network. It is important to note that this characterization aims to capture the essential learning potentials of a learning object, as opposed to [7], where a methodology for evaluating multimedia is described which takes in account only the presentation and usability of the material. There have been some works aimed to predict the degree of acceptation a user may have of a multimedia file, like the work presented in [13] with images but it does not consider a the importance this material has for the user in the context of his/her work.

This work proposes a methodology for characterizing multimedia learning material based on the use of collaborative techniques in order to define a vector of characteristics for a certain document. This vector will reflect the opinion the people who have seen this document before and will evolve as new people express their opinion about the document. However, not all users will get the same vector as description of one document. In order to construct the vector for a certain user, the opinion given by those with similar interest will have more weight. Current recommender systems mostly do not use implicit ratings, nor is the ability of implicit ratings to predict actual user interest well understood. An adaptive method should be able to learn and "calibrate" the learner's preferences based on her/his behavior. Apart from the preferences about the content of the learning material expressed by the learner the system will also consider the possibilities he/she has to display/perform a certain kind of learning object. This corresponds to the characteristics of the software, hardware and connection the learner has available by the moment a searching requirement is expressed or a certain learning material suggested. These might or might not be taken in account by the learner in order to decide whether to download the material or not. The next section presents some works related to recommender systems. In section 3 we illustrate the model proposed. Section 4 describes how the model works in the real situation of the COLDEX project. Finally section 5 presents some conclusions and further work.

2 Related Work

A variety of collaborative filters or recommender systems have been designed and deployed. The Tapestry system relied on each user to identify like-minded users manually [10] and is one of the earliest implementation of collaborative filtering-based recommender systems. However, because this system depends on each person knowing the others, it is not suitable for large. Later several rating-based automated recommender systems were developed. Grouplens (Resn94) and Ringo (Shar95) developed independently systems, where the first CF algorithms for automatic prediction were used. The work of Breese et al. [5] identifies a general class of Collaborative Filtering algorithms called model-based algorithms. The authors describe and evaluate two probabilistic models, which they term the Bayesian clustering and Bayesian network models. In the first model, like-minded users are clustered together into classes. Given his/her class membership, a user's ratings are assumed to be independent (i.e., the model structure is that of a naive Bayesian network). The second model also employs Bayesian network, but of a different form. Variables in the network are titles and their values are the allowable ratings. Bayesian networks create a model based on training set with a decision tree at each node and edges representing user information. The model can be built off-line over a matter of hours or days. The resulting model is very small, very fast, and essentially as accurate as nearest neighbor methods [5]. Other technology that has been used is Horting, that is a graph-based technique in which nodes are users, and edges between nodes indicate degree of similarity between the users [1]. Walking the graph nearby nodes and combining the opinions of the nearby users produces predictions.

3 The Systems's Model

The recommender system will be used for performing the following tasks:

1. An agent proactively proposes certain learning material to the learner, triggered by a tutor of the learning community.
2. The user makes a search of relevant learning material based on a keyword list.
3. Given a document, the system evaluates it for the user according to her/his profile.

Now we will describe the principal components of the model on which the recommender system will be based and their functionality.

3.1 Learner's Profile or Metadata of the Learner

The learner's profile is used to describe the characteristics of a certain learner in order to make an automatic evaluation of the available learning material in order to filter documents which will be of no use to her/him. The historical registers about the material the user has selected and evaluated in the past, as well as the preferences declared explicitly by her/him will automatically generate the description of the user's interests. In our system we consider two types of properties describing the learner's profile. On one hand we have the *user preference properties*, which describe the learner's preferences for a certain type of material, and on the other hand, we will have those describing the hardware and software which the learner has available to display it. We will call these ones the *user hardware properties*. The preference properties we are going to consider are: interest fields, described by a list of keywords, preferred multimedia format, language, date of profile creation, author, age, expected difficulty, expected time of learning, semantic density and context. The values for these properties are the components of the preference properties vector UPPi. Figure 1 shows the XML description, which is used to characterize this vector.

```
<userProfile
xmlns="http://www.d.cl/~pgaldame/tesis"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.d.cl/~pgaldame/tesis
http://www.d.cl/~pgald/userProfile1.xsd">
    <identifier id="000001">
        <version/>
    </identifier>
    <topic>
        <name>astronomy</name>
        <degreeTopic value="4"/>
    </topic>
    <topic>
        <name>chess</name>
        <degreeTopic value="5"/>
    </topic>
    <format/>
    <downloadTimeExpected tolerance=""/>
    <educationalLevel>
        <source/>
         <value> high school student</value>
    </educationalLevel>
    <expectedifficulty>
        <source> CS Thesis <source/>
        <value> medium </value>
    </expectedifficulty>
   </userProfile>
```

Fig. 1. Example of a user's preference profile.

The characterization of the learner's available hardware profile is done according to the CC/PP recommendation [11] and defines the user hardware properties vector UHP_i. Figure 2 describes the content of this vector and some example values. Therefore a user's profile UP_i is completely defined by the two vectors $\{UPP_i, UHP_i\}$. While UPP_i is defined and maybe normally occasionally changed by the user, UHP_i can be completely automatically defined by the system every time the user logs in or manually defined by the user, since it can vary as often as the learner changes the computer in which he/she is working.

3.2 Metadata of the Multimedia Learning Material

In our system all the learning material will have an associated metadata for description. In order to facilitate their manipulation these will be divided in metadata describing the content itself and those describing its contribution to learning a certain subject. Learning objects will be classified in a certain class according to the principal learning field they are supposed to be used for. A class is composed of a set of items describing more specific topics of the learning field. A document will be evaluated according to the contribution it makes for the learning of each topic of the field of the class it belongs to. This evaluation will be made collaboratively by all learners that used that learning object before and will be adapted to the preferences of the user to which this evaluation will be presented. For describing the learning object from the

Hardware	*Processor:* Computer's processor. Example values = {PPC, Intel-Pentium X; Athlon, Motorola}. *Memory*: Computer's RAM size in MB. *Screen:* Screen Resolution in pixels. *Free_hard_disk_space*: in MB.
Software	*Sound:* values = {ON, OFF} *Images:* Image viewing support, values ={ yes, no} OS : Operative System, values = {pc-dos, ms-windows, ..., other} *Browser*: Installed Web-browser, state={any, netscape, communicator, microsoft_internet_explorer, mosaic, mozilla, opera} *Version*: installed browser's version *Internet connection type*: values = {dial-up, leased link, wireless} *Bandwidth*: Maximal bandwidth provided in KB/s. *Mean download rate*: in KB/s.

Fig. 2. Description of the components of the user's hardware profile.

point of view of its content we take elements of the LOM [12] standard which describe the format, language, semantic density, size, installation requirements, and previous knowledge required.

Formally a document will be described in the following way: Let Doc_{jx} ,be the j-th document of class x. $FE_0(Doc_{jx})$ represents the vector initial evaluation of the document made by an expert. This evaluation is done having "normal" user profile in mind, for which this learning object is aimed in the context of the learning community. The expert evaluates the contribution of the document to all topics of the class giving a value between 0 and 1, in which 0 denotes no contribution and 1 a high contribution. For each user k that has used and evaluated that document before, there is an evaluation vector $FE_k(Doc_{jx})$ similar to $FE_0(Doc_{jx})$ which contains the evaluation the learner has made about the learning contribution of the document for each topic of the class. For the Doc_{jx} document, there is also a vector describing it form the point of view of the characteristics of its content taken from the LOM standard denoted by $CC(Doc_{jx})$. Let $CP_i(Doc_{jx})$ be the estimated learning contribution of the document Doc_{jx} to the user i automatically generated by the system calculated by the following equation:

$$CP_i(Doc_{jx}) = \frac{1}{N} * \sum_k FE_k(Doc_{jx}) * g(UP_i, UP_k) + FE_0(Doc_{jx}) \tag{1}$$

In this equation N is a normalization factor in order to have values between 0 and 1 for each component of $CP_i(Doc_{jx})$, so N is the number of vectors $FE_k(Doc_{jx})$ including $FE_0(Doc_{jx})$. The function g of the equation determines the degree of similitude between the profiles of the user i and user k. This function will give a value close to 1 if both profile tend to be similar and close to 0 if they tend to defer. As said, the vector $CP_i(Doc_{jx})$ is the estimation of the system about how would the user i evaluate the document. After exploring the document, the user may agree with this evaluation, thus declaring his final evaluation equal to the given by the system thus making $FE_i(Doc_{jx}) = CP_i(Doc_{jx})$ or providing a vector with fully new or partially modified values in order to be used by the system for calculating estimations for other members of the learning community.

3.3 Characteristics of the "g" Function

The g function described in the equation 1.0 should evaluate the degree of leverage an opinion a certain user has about a document (expressed by the vector $FE_k(Doc_{jx})$) to the user for which the vector $CP_i(Doc_{jx})$ is being calculated. We want, of course, that this function gives more importance to opinion expressed by people having similar interests, and backgrounds, which is reflected in the user's profile information stored in the UPi. In our context the system will define two user's profile as similar if the keywords contained and the values for those keywords in both vectors are within a certain defined "distance" or "threshold". In our system we use statistical correlation like those defined [14]. This metric incorporates not only the preferences of the users by also their background knowledge.

3.4 User Adaptive Filtering Techniques

For predicting the acceptance of a certain user will have of a certain document most of the existing recommender systems take in account only the user's requirements declared in the user's profile. Zhang [16] points out that this approach may be incomplete, because it does not take in account the contribution of the document to the knowledge of the user, that is, if the learner will learn something new with the learning object. In order to consider this aspect in our system we have to keep some information about what kind of material the user has already received. We do this by keeping for each user, a set of vectors TLC_{xi} which represents for a user i the total amount of information received for each topic of the class x from all documents she/he has already downloaded. For calculating the estimated increase to this value a certain document Doc_{jx} may cause by downloading and using it our system uses the vector TLC_{xi} and the estimated evaluation of the learning contribution of the document $CP_i(Doc_{jx})$. We can thus describe this estimated increment by the equation $EDTLC_{xi} = f(TLC_{xi}, CP_i(Doc_{jx}))$. The function f will follow the law of the decreasing returns, that is, the increment is smaller when TLC_{xi} is bigger. At the beginning, when the learner has downloaded only a few documents, the contribution of a new document might be higher than after the user has downloaded many documents. After the learner gives the final values for evaluating the learning contribution of a document $FE_i(Doc_{jx})$, the vector TLC_{xi} is updated by applying $TLC_{xi} = TLC_{xi} + f(TLC_{xi}, FE_i(Doc_{jx})$

For this calculations use some mechanisms borrowed from the field of Information Retrieval for modeling the user's behavior like vectorial spaces [14], diffuse logic based models and neuronal networks [13], and bayesian classifiers [2, 15]. In order to introduce a mechanism of automatic learning the model uses the algorithm of Bilmes [3].

4 How Does the System Works

When a new learning material is included in any of the distributed learning community servers, a tutor has classify it in a certain class x and generates the first evaluation vector $FE_0(Doc_{jx})$ and the content description of it $CC(Doc_{jx})$. According to equation 1.0 at this point, for any user i the evaluation vector for this document will be $FE_0(Doc_{jx})$ since there is no other user who has used this document yet. When a new learner i joins the community, he/she has to define the preferences vector UPP_i. The values for all components of the vectors LC_{xi} for all x classes are set to 0. When the system evaluates a new document for the learner i it first searches for other learners who have an evaluation vector $FE_k(Doc_{jx})$ for that document. Whit this information, plus the initial vector $FE_0(Doc_{jx})$ and the learners preference profile UPP_i it calculates $CP_i(Doc_{jx})$. Then it calculates the increment for the learner's $LC_i(Doc_{jx})$ and compares the document's content characteristics $CC(Doc_{jx})$ with the learner's UHP_i and UPP_i (for determining, for example, if according to the document's size, the current downloading rate of the user and his/her maximum download time tolerance the learner will be willing to download it). At this point the system is able to tell the learner how interesting is the document, which is its learning contribution, and if there are some problems for downloading and/or using it in the hardware/software he/she has currently available. With this information, an HTML page with this information and a link to the respective learning material is generated and presented to the learner. The learner may decide to download the document right now, save this information for deciding later or discard this information. When the learner decides to download the material then the systems requires him/her to provide an evaluation vector $FE_i(Doc_{jx})$ or confirm these evaluation in the near future.

5 Conclusions and Further Work

The system we have presented propose to use the recommender systems approach to support the exchange of information about learning objects available inside a virtual learning community like COLDEX project, in order to help learners find suitable learning material for them. Most existing recommender systems rely upon the exchange of texts between writers and readers in an ongoing discursive activity. The model we have developed tries to be closer to the way human beings operate, including different kinds of recommendations. One of the inconveniences we try to solve in our model is to avoid the information overloading. As Furnas [9] mentions, huge amount of information will generate several interface problems. If we present so much information, it will be difficult for the user to focus on the essential aspects of their work. In order to avoid information overloading we need to show only the relevant required information. In that way, some authors have recommended the use of awareness filters [8] to present only the most relevant information in a similar way as proposed by our model. Considering that not all details are relevant and open to outsiders, an awareness mechanism should somehow filter the information [4]. The system is now in its implementation phase. As future work we hope to carry out several experiments in order to validate our model, comparing our results with other CF models identifying when our proposed schema is more suitable. This work will be ready by the time we might present this work in the workshop.

Acknowledgments

This work was supported by MECESUP (Chile) project No. UCH0109.

References

1. Aggarwal, C., Wolf, J., Wu, K., Yu, P.,: Horting Hatches an Egg: A New Graph-theoretic Approach to Collaborative Filtering. Proc. of the ACM KDD'99, San Diego, CA, (1999) 201-212
2. Ahmad M., Ahmad Wasfy,: Collecting User Patterns for Building User Profiles and Collaborative Filtering. IUI 99, Redondo Beach, CA, USA, (1999)
3. Bilmes, J.A., A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, International Computer Science Institute, Berkeley California (1998)
4. Borges, M., Pino, J., Valle, C.: Support for Decision Implementation and Follow-Up. Accepted for European Journal of Operations Research, (2003)
5. Breese, J., Heckerman, D., Carl, K.: Empirical analysis of predictive algorithms for collaborative filtering. Proc. of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence, (1998) 43-52
6. http://www.coldex.info.
7. Crozat, S., Hu, O., Trigano, P.: A Method for Evaluating Multimedia Learning Software. IEEE Conference on Multimedia Computing and Systems Volume 1, (1999)
8. David, J., Borges, M.: Selectivity of Awareness Components in Asynchronous CSCW Environments. CRIWG'01, Darmstadt, Germany, (2001)
9. Furnas, G., Bederson, B.: Space-Scale Diagrams: Understanding Multiscale Interfaces. Proc. Of Human Factors in Computing Systems, CHI'95, ACM Press, Denver, CO, (1995) 234-241
10. Goldberg, D., Nichols, D., Oki, B., Terry, D.: Using collaborative filtering to weave an information tapestry. Communications. of the ACM, 35(12), (1992) 61-70
11. Kline, G., Reynolds, F., Woodrow, C., Ohto, H., Hielm, J., Butler, M., Tran, L.: Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. W3C Recommendation 15, (2004)
12. Computer Society/Learning Technology Standards Committee. IEEE Standard for Learning Object Metadata. Approved Publication of IEEE, (2002)
13. Mitaim, S., Kosko B.: Neural Fuzzy Agents that Learn a User's Preference Map. Proc. of the 4th International IEEE Forum on Research and Technology Advances in Digital Libraries, ADL (1997)
14. Savia, E., Kurki, T., Jokela, S.: Metadata Based Matching of Documents and User Profiles, in Human and Artificial Information Processing, Proc. of the 8th Finnish Artificial Intelligence Conference, August 1998, Jyväskylä, Finland, (1998) 61-70
15. Wong, S., Butz, C.: A Bayesian Approach to User Profiling in Information Retrieval. Technology Letters, 4(1), (2000) 50-56
16. Zhang Y., Xu W., Minka, T.: Novelty and redundancy detection in adaptive filtering. Proceedings of the 25th ACM SIGIR Conference, (2002)