Mining Social and Affective Data for Recommendation of Student Tutors

Elisa Boff[1] and Eliseo Berni Reategui[2]
[1]*Universidade de Caxias do Sul (UCS) – Brazil*
[2]*Universidade Federal do Rio Grande do Sul (UFRGS) – Brazil*

***Abstract* — This paper presents a learning environment where a mining algorithm is used to learn patterns of interaction with the user and to represent these patterns in a scheme called item descriptors. The learning environment keeps theoretical information about subjects, as well as tools and exercises where the student can put into practice the knowledge gained. One of the main purposes of the project is to stimulate collaborative learning through the interaction of students with different levels of knowledge. The students' actions, as well as their interactions, are monitored by the system and used to find patterns that can guide the search for students that may play the role of a tutor. Such patterns are found with a particular learning algorithm and represented in item descriptors. The paper presents the educational environment, the representation mechanism and learning algorithm used to mine social-affective data in order to create a recommendation model of tutors.**

***Keywords* — Collaboration, Learning Environment, Recommender Systems, Social-Affective Data.**

I. INTRODUCTION

MINING data in educational environments is often used with two main purposes:

(1) to give educators a better understanding of how users learn with the system;

(2) to define different paths of study according to students' profiles learned from data.

The first goal may be achieved by using mining algorithms to identify patterns and represent them in a scheme that is easy to understand. The second goal can be pursued by employing a mechanism capable of using the patterns found to suggest topics related to the subjects being studied.

We used mining algorithms here in order to accomplish both purposes (1 and 2), and also to identify suitable student tutors that may help other students needing assistance. The use of data mining in Education has expanded considerably in the last decade mostly because of the growing number of systems that store large databases about students, their accesses to material available, their assignments and grades. Such expansion in the field yielded the establishment of a community concerned mostly with the development of methods for exploring data coming from educational settings, and employing those methods to better understand students and learning processes [4].

Current research has shown the potentiality of cooperative learning, demonstrating that group work is fundamental for the cognitive development of the student [7] [8]. It is known that knowledge composition occurs on an individual basis, but cooperation (subjects acting together over the same topic, with common goals, interacting and exchanging ideas) is capable of involving all participants in learning [18]. In this perspective, motivating the students to interact can lead to an effective learning practice.

The recommendation service of tutors works in the sense of motivating group formation among the students. According to Andrade [1], a group can be formed due to similarity and empathy of its members or to the necessity of support for the accomplishment of some task. The latter can be motivated by prestige or status, economic benefits or the necessity and desire of contribution. [1] also says that the affective states of the individuals have significant importance in the interaction process. The author complements affirming that some dimensions of the personality seem to have certain connections with the social performance in the interaction, but establishing an accurate relationship between them seems to be a complex task.

Our tutor recommendation service explores the social-affective dimension through the analysis of emotional states and social behavior of the users. A recommender system analyses students' interactions and finds suitable tutors among them as well as contents to be recommended. A specific algorithm was built to identify behavioral patterns in the students interaction, and to store this knowledge in structures called item descriptors [19]. The method proposed shows a good performance with respect to processing time and accuracy, and has an advantage over other techniques when it comes to understanding the knowledge elicited and letting users modify it. The first section of the paper gives an overview of the types of data collected from the interaction with the users. Then, the mechanism employed to represent knowledge is explained, in addition to its learning algorithm and recommendation process. Finally, preliminary results are discussed, as well as conceptual advantages and drawbacks of the approach. The last section of the paper offers conclusions and directions for future work.

DOI: 10.9781/ijimai.2013.214

II. Collecting Interaction Data

When students navigate in our learning environment (Fig.2), different types of data are collected from their interaction. By keeping the navigation history of every student, for example, we are able to identify navigation patterns and to use them in real-time recommendation of contents. For the recommendation of tutor colleagues, six other types of data are collected: Social Profile; Acceptance Degree; Sociability Degree; Mood State; Tutorial Degree and Performance.

The Social Profile (SP) is built during the communication process among students. The following information is collected during the interaction of the students through an instant message service:

• *Initiatives of communication*: number of times that the student had the initiative to talk with other pupils.

• *Answers to initial communications*: in an initial communication, number of times that the student answered.

• *Interaction history*: individuals with whom the student interacts or has interacted, and number of interactions.

• *Friends Group*: individuals with which the student interacts regularly, and number interactions.

Based on Maturana [15] we defined the Acceptance Degree (AD), which measures the acceptance a student has for another one. Such data is collected through a graphical interface that enables each student to indicate his/her acceptance degree for other students. This measure may also be considered from a point of view of Social Networks, which constitutes one of the most popular approaches for the analysis of human interactions. The most important concept in this approach is centrality. If an individual is central in a group, he/she is popular and gets a great amount of attention from the group members. As the AD is indicated by the students themselves based on their affective structures, the measurement can indicate diverse emotions, such as love, envy, hatred, etc. The average of all AD received by a student influences his/her Sociability Degree (SD).

The Mood State (MS) represents our belief in the capability of a student to play the role of a tutor if he/she is not in a positive mood state (although the student may have all the technical and social requirements to be a tutor). We consider three values for the MS: "bad mood", "regular mood" and "good mood". These states are indicated by the students in a graphical interface through corresponding clip-arts.

After a helping session, a small questionnaire is submitted to the student who got assistance. The goal of this questionnaire is to collect information about the performance of the tutor. The questions made are based on concepts from Social Networks and Sociometry, and may be answered by four qualitative values: "excellent", "good", "regular", and "bad". They are:

• *How do you classify the sociability of your class fellow?*

• *How do you classify the help given by your class fellow?*

The answer to the first question together with the average of the ADs of a student, form his/her Sociability Degree (SD). This measure indicates how other individuals see the social capability of this student.

The Tutorial Degree (TD) measures a student's pedagogical capacity to help, to explain and teach. This value is obtained from the answers given for the second question of the questionnaire above and from the marks the tutor got when he/she studied the contents for which he/she was asked for help. These marks were called Performance (P) and were used in the computation of the TD because when a tutor is not able to help another student it does not necessarily mean that the student is a bad tutor. He/she may simply not know very well the content for which his/her help was requested. Therefore, the answers of the students have to be "weighted".

A mining process determines relationships among these factors, and represents such relationships in item descriptors, which are later used for recommendation purposes.

III. The Item Descriptors

An item descriptor represents knowledge about when to recommend a particular item (a topic of study, an exercise, or a tutor) by listing other items found to be related to it. Users have features that may be classified as:

• *demographic*: data describing an individual, such as age, gender, occupation, address;

• *behavioral*: data describing tutoring and social capacity, navigation and study patterns.

It has been shown that both types of data are important when building a user profile [13] and inferring user's needs [5] [6]. Demographic material is represented here in attribute-value pairs. Behavioral information is represented by actions carried out by the user, such as the selection of a topic for reading. Emotional states and social behavior can either be inferred or collected explicitly in questionnaires.

While attributes used to define demographic features are typically single-valued, behavioral data is usually multi-valued. For instance, a person can only belong to one age group (demographic), but he/she may be friendly and patient at the same time (behavioral). Nevertheless, both types of information are represented in our model in a similar way. Let us examine an example of an item descriptor and its related items (Table 1).

TABLE I
ITEM DESCRIPTOR AND RELATED ITEMS

DESCRIPTOR D_N	
Correlated terms	**Confidence**
t_a	0.92
t_e	0.87
t_c	0.85
t_d	0.84
t_b	0.77

The descriptor has a *target* (d_n), i.e. an item that may be recommended in the presence of some of its correlated terms. Each term's class and *confidence* (the strength with which the

term is correlated with the target item) is displayed next to its identification.

We use *confidence* as a correlation factor in order to determine how relevant a piece of information is to the recommendation of a given item. This is the same as computing the conditional probability $P(dj|e)$, i.e. the probability that the item represented by descriptor d_j is rated positively by a user given evidence e. Therefore, the descriptors can be learned through the analysis of actual users' records. For each item for which we want to define a recommendation strategy, a descriptor is created with the item defined as its target. Then, the confidence between the target and other existing demographic features and behavioral data is computed. This process continues until all descriptors have been created. For the recommendation of tutors, descriptors are built indicating the features of good and bad instructors.

IV. The Recommendation of Tutors

Collaborative Filtering, one of the most popular technologies in recommender systems [15], has been used in the past in several research projects, such as Tapestry [13], GroupLens [27], and more recently in related research focusing on the extraction of information from social networks [9][21]. The technique is based on the idea that the active user is more likely to prefer items that like-minded people prefer [28]. To support this, similarity scores between the active user and every other user are calculated. Predictions are generated by selecting items rated by the users with the highest degrees of similarity.

Here, a different approach has been followed, as the main idea in the project was not to keep track of users' interests, but to evaluate their willingness to collaborate. This task, called here recommendation of tutors, is explained below.

Given a list of possible tutors $U=\{u_1, u_2,..., u_m\}$, the recommendation process starts with the gathering of demographic and behavioral information about each of them. Next, the data collected for each user is matched against a descriptor d_j which lists the most important features of good instructors, according to the terms $T=\{t_1,t_2,...,t_k\}$ stored in the descriptor. The system computes a score for each student that ranges from not similar (0) to very similar (1), according to the formula:

$$Score\ (d_j) = 1 - \prod_{kji} (Noise\ (t_p))$$

where $Score(d_j)$ is the final score of the descriptor d_j; $Noise(t_p)$ is the value of the noise parameter of term t_p, a concept used in noisy-OR probability models (Pradhan et al., 1994) and computed as $1 - P(d_j \mid t_p)$. The individual with the highest score is selected to assist the student needing assistance.

That expression contains an assumption of independence of the various t_p - which the designer of a practical system should be trying to achieve in the choice of terms. Ultimately the test of the assumption is in the users' perception of the quality of a system's recommendations: if the perception is that the outputs are fully satisfactory, this is circumstantial evidence for the soundness of the underlying design choices. The situation here is the same as in numerical taxonomy [21], where distances between topics id in a multidimensional space of attributes are given by metric functions where the choice of distinct dimensions should obviously aim to avoid terms that have mutual dependences. If the aim fails, the metric cannot - except occasionally by accident - produce taxonomic clusters C (analogous to sets of topics offered by a recommender system once a user has selected one member of C) that satisfy the users. This method is based on the assumption that any term matching the user's terms should increase the confidence that the descriptor holds the most appropriate recommendation. In a real-life example, let us suppose that we have a certain degree of confidence that a student who has shown a good ability in answering factorial exercises is our best bet to help another student who is having problem with the subject. Knowing that that same student is friendly and is in a good mood should increase the total confidence on his recommendation as a tutor, subject to not exceeding the maximum value of 1.

The Virtual Character is the interface element that delivers to student the result of recommendation process in natural language (Fig.1).

The knowledge base of the Virtual Character stores knowledge about Algorithms, enabling the character to assist students mainly in theoretical questions. The Artificial Intelligence Markup Language (AIML) is used to represent the character's conversational knowledge [30], employing a mechanism of stimulus-response. The stimuli (sentences and fragments which may be used to question the agent) are stored and used to search for pre-defined replies. The most important AIML tags are:

- **<aiml>**: indicates the beginning of a document.
- **<category>**: the simplest knowledge unit in AIML. Each category consists of an input question, an output answer and an optional context. The question, or stimulus, is called the pattern, while the answer is called the template.
- **<pattern>:** keeps a set of words which is searched for in sentences which the user may enter to communicate with the virtual character. The language that may be used to form the patterns includes words, spaces, and the wildcard symbols _ and *;
- **<template>**: when a given pattern is found in the input sentence, the corresponding template is returned and presented to the user. In its simplest form, a pattern is a word and the template consists of plain text. However, the tags may also force the conversion of the reply into a procedure which may activate other programs and recursively call the

pattern matcher to insert the responses from other categories.

The optional context of a category enables the character to remember a previous statement. This feature, together with the possibility of launching particular programs when a certain pattern is found, makes the AIML communication mechanism very distinct from a simple retrieval of questions and answers from a database.

Fig. 1. Recommendation example.

The user's affective state is also considered in order to choose the type of language the character uses to talk at a given moment. The affective state is entered as a pattern which has to be matched for the selection of a given sentence. For instance, the pattern *RECURSION* is modified into *RECURSION CHEERFUL* if the user is in a cheerful mood.

In addition to the existing AIML tags, new ones were created to manage the agents' emotional appearance. For instance, we created the tag <*humor*> to control the image changes reflecting different moods of the virtual character (happy, receptive, annoyed, etc).

Therefore, when the user poses a question (stimulus), the character starts the AIML Retrieval Mechanism in order to build an appropriate reply using the information, patterns and templates from the AIML database. A suitable picture of the character is picked from the Image Database to match the sentence retrieved according to the humor tag.

In addition to being able to answer questions in natural language, our character is also able to monitor the actions of each student and notice, for instance, that a particular topic is related to a given exercise. Such a behavior is achieved through the use of the template tag to launch the recommender system, which looks for appropriate activities and contents to each student.

V. Validation and Discussion

An Environment for the Learning of Algorithms (A3), Fig. 1, has been developed at the Department of Computer Science of the University of Caxias do Sul with the main goal of making the courses more dynamic, increasing the interest and participation of the students and providing an environment where students may interact in order to improve their knowledge. The environment presents students with the regular contents of algorithms (central area of Fig.2), it proposes exercises, provides a forum for discussion and a tool for the testing and running of algorithms. All website functions can be accessed by the left menu on the detail 3 of Fig.2. Having been developed as a dynamic website, the system enables teachers and administrators to modify contents easily. Online users are shown in the interface (detail 2 of Fig.2). And most importantly, the system promotes the communication among students by suggesting individuals that may help others showing difficulty in learning a given topic. The recommendation is present in the detail 4 of Fig.2, below the image of Virtual Character.

Fig. 2. Environment for the Learning of Algorithms (A3).

The Affective States of students describe social-affective data which is used to recommend students tutors. The system does not try to infer social-affective states, but the user deliberately informs it about how he/she feels at login time (detail 1 in Fig.2). This information is used to define the type of language and stimuli that our Virtual Character has to show in order to communicate better with the user.

The A3 environment started to be tested in 2 courses at the Department. Descriptors were built manually in order to get the system to recommend contents and tutors. The data collected so far has not been sufficient for us to carry out conclusive experiments as to whether the system is making tutoring recommendations appropriately. However, initial experiments carried out and reported in Reategui [19] show that the item descriptors have a good performance in terms of

processing time and accuracy, when compared with collaborative filtering, one of the most popular approaches in recommender systems.

For the MovieLens database[1], for example, storing anonymous ratings of 3900 movies assigned by 6040 users, the item descriptors show an accuracy rate that is 6 points higher than that of the k-nearest neighbor algorithm. The Table 2 summarizes the results obtained.

The experiments were carried out considering neighborhoods with sizes 1, 20 and 40 (we did not observe any significant improvement in accuracy for the nearest-neighbor algorithm with neighborhoods larger than 40). The topic descriptors performed better than the k-nearest-neighbor algorithm, no matter what size of the neighborhoods was chosen.

Sarwar [20] have carried out a series of experiments with the same data set, employing the Mean Absolute Error (MAE) method to measure the accuracy of item-based recommendation algorithms. The results reported could not be compared directly with our own as the authors computed their system's accuracy using the MAE and considering integer ratings ranging from 1 to 5 (reaching values around 75%). In our experiment, we only took into account whether a user rated (1) or did not rate (0) a topic.

TABLE II
SCORING RESULTS FOR THE MOVIELENS DATA SET

Method	Scoring
Item Descriptors	65,7
k-nearest-neighbor (k=1)	39,3
k-nearest-neighbor (k=20)	54,9
k-nearest-neighbor (k=40)	59,7

In order to evaluate the system's performance, we monitored how much time was spent by the system in order to recommend the 2114 topics in the test data set[2]. For k=1, the nearest-neighbor approach needed less time than the topic descriptors to perform the tests, though showing a lower rate of accuracy. However, for larger values of k (or simply larger numbers of users) the performance of the nearest-neighbor algorithm degrades, while that of the topic descriptors remains stable. Table 3 summarizes the results of the experiment.

In more realistic situations where the nearest-neighbor algorithm may have to access a database containing actual users' transactions, the nearest-neighbor approach may become impractical. For the same experiment described above, we tested the nearest-neighbor through access to an actual database, using k=10. A few hours was needed for the system to make the whole set of recommendations. Further validation results may be found in Reategui [19].

Another popular approach applied to recommender systems is association rules [14] (Mombasher, 2001). This technique use well-known inductive learning algorithms, such as *a priori* [2], to extract knowledge and represent them in "if ... then ..." rules format. The main advantage of such learning method relies on the robustness and stability of the algorithms available. Although being successfully applied in innumerable application areas, association rules are hard to modify while keeping the rule base consistent (e.g. adding new rules without contradicting existing ones). Keeping track of and trying to understand the large number of generated rules for each topic is another difficulty of this approach.

TABLE III
PERFORMANCE RESULTS FOR THE MOVIELENS DATA SET

Method	**Time spent in secs.**
Topic Descriptors	32
k-nearest-neighbor (k=1)	14
k-nearest-neighbor (k=20)	43
k-nearest-neighbor (k=40)	86

The item descriptor approach is different in that it represents knowledge in the form of descriptors and correlation factors. When compared with the other approaches in this respect, descriptors are interesting because they make it easy for users to understand as well as modify the knowledge represented. This is particularly important when the user wants to make the system respond in a certain way in given circumstances, e.g. if the teacher wants the system to recommend a certain reading when the student is viewing a particular topic.

The learning mechanism used on the item descriptors also exploits well-known methods to compute correlation factors and define the strength of the relationships among features and topics. The option to use term confidence instead of conditional probability to describe the model comes from the fact that other correlation factors that are not supported by probability theory are computed by the system, such as interest and conviction [4]. However, at present these are provided only to let the user analyze and validate the knowledge extracted from the database. We are currently testing different variations on the combination of these factors in the reasoning process.

Although the system learns and updates its descriptors in an offline process (therefore not critical for the application to recommend topics in real time), our learning algorithm is fairly simple and fast. Above all, it is faster than algorithms that group evidence and try to compute the relevance of each topic and then of each group of evidence.

Our model may also be compared with Hidden Markov Models (HMM), employed in tasks such as the inference of grammars of simple language [10], or the discovery of patterns in DNA sequences [3]. The two models are similar in that both

1 MovieLens is a project developed in the Department of Computer Science and Engineering at the University of Minnesota (http://movielens.umn.edu).

2 The tests were performed on a PIII 500MHZ PC with 128Mb of RAM.

use probability theory to determine the likelihood that a given event takes place. However, the actual methods used to compute probabilities of events are different: while HMM considers the product of the probabilities of individual events, we consider the product of noise parameters. Both models are based on the assumption that an output is statistically independent of previous outputs. This assumption may be limiting in given circumstances, but for the type of application we have chosen, we do not believe this to be a serious problem (e.g. as we have remarked above in our comments on independence). To take one practical example, the probability that a user studies topic C is very rarely dependent on the order in which users have read other topics (e.g. B before A, or A before B).

The recommendation method we use has the peculiarity of computing the correlation of individual terms initially, and then combining them in real time. This is analogous to finding first a set of rules with only one left-side term, followed at run time by finding associations between the rules. This is a good technique to avoid computing the relevance of all possible associations among terms in the learning phase.

Gomes [11] proposes a different recommendation strategy to identify tutors based on the computation of a utility function. Their strategy combines features in a mathematical expression to determine how effective a student can be for a given tutoring task. Compared to this approach, our mining and recommendation mechanism is more interesting in that it uses learning algorithms to learn a model from the available data automatically, identifying the importance of each utility function variable.

VI. Conclusion

One important contribution of this work has been the definition of the types of data to be used in the mining and in the recommendation process of student tutors. Using the descriptors to calculate the relevance of terms individually, and then combining them at recommendation time through the use of the noisy-OR is also a novel approach. A similar use of the function can be found in research on expert systems [9], but not in applications for recommender systems. Initial results have shown that the approach can be very effective in large-scale practice for personalization purposes.

The use of social-affective information to promote the communication and collaborative learning among students is starting to be tested in the environment A3. The results obtained so far show that the use of Social Profile, Mood State, Performance Acceptance, Sociability and Tutorial Degree in tutor recommendation, is a promising alternative.

Although the data collected from students' interactions so far are not sufficient for us to draw assertive conclusions about the use of item descriptors to recommend tutors, other experiments have shown the adequacy of the approach in item recommendation.

The possibility to represent different types of information (demographic or behavioral) in a similar way seems to be advantageous when it comes to practical implementation issues. Previous work in the field has shown the importance of dealing with and combining such types of knowledge in recommender systems [17]. Current research on the identification of implicit user information also shows that recommender systems will have to manipulate different sorts of data in order to infer users' preferences [6].

One of our biggest challenges now concerns the automatic inference of students' affective states. At present we are using questionnaires and graphic interface controls to let the users indicate such states. Thus, little is done to automatically infer the social-affective information necessary for tutor recommendation. This will be one of our main research efforts in the near future.

This project should also be integrated with the JADE/MAIDE platform [11] [22] and have its knowledge used in the MACE platform [1].

References

[1] Andrade, A.; Jaques, P.; Viccari, R.; Bordini, R. and Jung, J. 2001. A Computational Model of Distance Learning Based on Vygotsky's Socio-Cultural Approach. In Proceedings of Multi-Agent Based Learning Environments Workshop. International Conference on Artificial Intelligence on Education. Antonio, Texas, May 19-23.

[2] Agrawal, R. and Srikant, R. 1994. Fast Algorithms for Mining Association Rules. In Proceedings of the 20th International Conference on Very Large Databases, Santiago, Chile.

[3] Allison, L.; Stern, L.; Edgoose, T. and Dix, T. I. 2000. Sequence Complexity for Biological Sequence Analysis. Computers and Chemistry 24(1):43-55.

[4] Baker, R. S. J. D., Yacef, K. 2009. The State of Educational Data Mining in 2009: A Review and Future Visions, Journal of Educational Data Mining, 1(1): 3-17.

[5] Brin, S.; Motwani, R.; Ullman, J. D.; and Tsur, S. 1997. Dynamic topicset counting and implication rules for market basket data. SIGMOD Record (ACM Special Interest Group on Management of Data), 26(2):255.

[6] Buchner A. and Mulvenna, M. 1998. Discovering Behavioural Patterns in internet files. Proceedings of Intelligent Tutoring Systems '98, San Antonio, Texas, USA. 16-19. Lecture Notes in Computer Science, 1452.

[7] Claypool, M.; Brown, D.; Le, P.; and Waseda, M. 2001. Inferring User Interest. IEEE Internet Computing, 5(6):32-39.

[8] Echeita, G., Martin, E. 1995. Interação Social e Aprendizagem. Porto Alegre: Artes Médicas.

[9] Finkelstein, A., Lim, S. L. 2012. Using Social Networks and Collaborative Filtering for Large-Scale Requirements Elicitation. IEEE Transactions on Software Engineering, 38(3): 707-735.

[10] Gallant, S. 1988. Connectionist Expert Systems. Communications of the ACM, Vol. 31, Issue 2.

[11] Georgeff, M. P. and Wallace, C. S. 1984. A General Selection Criterion for Inductive Inference. European Conference on Artificial Intelligence – ECAI 84, Pisa, 473-482.

[12] Gomes, E. R; Boff, E. and Viccari, R. 2004. Social, Affective and Pedagogical Agents for the Recommendation of Student Tutors. Proceedings of The Seventh Intelligent Tutoring Systems, Workshop on Social and Emotional Intelligence on Learning Environments, Maceió, Brazil, August, 31.

[13] Goldberg, D., Nichols, D., Oki, B. M. & Terry, D. 1992. Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35(12).

[14] Gomes, E. R.; Silveira, R. A. and Viccari, R. M. 2003. Utilização de agentes FIPA em ambientes para Ensino a Distância. In: XI Congresso Iberoamericano de Educação Superior em Computação (CIESC), Bolívia.

[15] Herlocker, J., Konstan, J. & Riedl, J. 2000. Explaining Collaborative Filtering Recommendations. In Proceedings of ACM Conference on

Computer Supported Cooperative Work. Philadelphia, Pennsylvania, USA.

[16] Krulwich, B. 1997. Lifestyle Finder: Intelligent User Profiling Using Large-Scale Demographic Data, Artificial Intelligence Magazine 18(2). 37-45.

[17] Lin, W.; Alvarez, S. A.; and Ruiz, C. 2000. Collaborative Recommendation via Adaptive Association Rule Mining. KDD-2000 Workshop on Web Mining for E-Commerce, Boston, MA, USA.

[18] Maturana, H. 1998. Emoções e Linguagem na Educação e na Política. Belo Horizonte: Ed. UFMG.

[19] Mobasher, B., Dai, H., Luo, T. and Nakagawa, M. 2001. Effective Personalization Based on Association Rule Discovery from Web Usage Data. Proceedings of the ACM Workshop on Web Information and Data Management. Atlanta, Georgia, USA.

[20] Nitzke, J., Carneiro, M. L., Franco, S. 2002. Ambientes de Aprendizagem Cooperativa Apoiada pelo Computador e sua Epistemologia. Informatica na educação: Teoria & Prática. Porto Alegre, v.5, n.1, p.13-23.

[21] Noel, J., Sanner, S., Khoi-Nguyen, T., Christen, P., Xie, L. Bonilla, E., Abbasnejad, E., Penna, N. D. 2012. New Objective Functions for Social Collaborative Filtering. ACM International World Wide Web, Lyon, France, 859-868.

[22] Pazzani, M. 1999. A Framework for Collaborative, Content-Based and Demographic Filtering. Artificial Intelligence Review. 13(5-6): 393-408.

[23] Piaget, J. 1973. Estudos Sociológicos. Rio de Janeiro: Companhia Editora Forense.

[24] Pradhan, M., Provan, G. M., Middleton, B., and Henrion, M. 1994. Knowledge engineering for large belief networks. In Proceedings of Uncertainty in Artificial Intelligence, Seattle, Washington. Morgan Kaufmann.

[25] Reategui, E.; Campbell, J. A. and Torres, R. 2004. AAAI - The Nineteenth National Conference on Artificial Intelligence, Workshop on Semantic Web Personalization, San Jose, USA, July.

[26] Sarwar, B. M.; Karypis, G.; Konstan, J. A. and Riedl, J. 2001. Topic-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International World Wide Web Conference - WWW10, Hong Kong.

[27] Sarwar, B., Konstan, J., Borchers, A., Herlocker, J., Miller, B. & Riedl, J. 1998. Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System. Proceedings of the Conference on Computer Supported Cooperative Work. Seattle, USA, 345-354.

[28] Shardanand, U. & Maes, P. 1995. Social information filtering: Algorithms for automating "word of mouth". In Proceedings of Human Factors in Computing Systems - CHI, Denver, Colorado, USA. 210-217.

[29] Sneath, P. A. and Sokal, R. R. 1973. Numerical Taxonomy: The Theory and Practice of Numerical Classification. San Francisco, CA: Freeman.

[30] Wallace, R. The Elements of AIML Style, ALICE A. I. Foundation. 2003.

Elisa Boff has her Ph.D. in Computer Science at the Federal University of Rio Grande do Sul (UFRGS), 2008. Currently is full professor and researcher at the Caxias do Sul University (UCS). She has experience in Computer Science with emphasis in the following topics: Artificial Intelligence applied to Education, Computers in Education, Recommender Systems and Distance Education. Dr. Boff is member of Brazilian Computer Society and she has publications at IEEE Journals and Proceedings, such as ITEE, ICTAI, ICALT, and Computers and Education.

Eliseo Berni Reategui has a PhD degree in Computer Science from the University of London, England, and MS in Computer Science from Universidade Federal do Rio Grande do Sul, Brazil. After finishing his PhD and working in the industry for 5 years, Dr Reategui held a lecturer position at the University of Caxias do Sul, Brazil, for a few more years. Nowadays, he works as a lecturer and researcher at the Federal University of Rio Grande do Sul. His research interests are related to the use of computers in education, involving areas such as artificial intelligence and human-computer interaction.