# RECMOODLE - AN EDUCATIONAL RECOMMENDER SYSTEM

Dhiego Carvalho[1], Carlos Tadeu Queiroz de Moraes[2], Dante Barone[1] and Leandro Krug Wives[1]
[1]*PGCC, UFRGS, Porto Alegre, RS, Brazil*
[2]*PGIE, UFRGS, Porto Alegre, RS, Brazil*

Keywords: Recommender Systems, Collaborative Filtering, Moodle.

Abstract: This paper proposes a recommender system for the Moodle platform that uses Apache Mahout as a recommendation engine. In this system, students receive personalized and adaptive recommendations based on forum discussions and learning materials available in the environment, according to their interactions. The recommendation method developed is based on the Moodle's activity log. In this paper, we compare four different ways to process user's preferences and present recommendations. Three of them exist in the literature, and one is proposed by us. The test set uses data from a real case scenario, i.e., the activity log from a graduation course that is given in our institution. We have found that the proposed approach produced the best set of recommendations.

## 1 INTRODUCTION

Recently, due to the increasing demand from the Industry and to the academic effort, new Recommender Systems' (RS) technologies were developed. The interest in this field is very high because of its many applications, including helping users cope with information overload and receiving personalized recommendations. Some examples include the recommendation of books, products, scientific papers, movies and so on (Amazon, 2011) (NetFlix, 2011).

In the educational context, recommender systems help students be more efficient. For instance, learners get recommendations about relevant topics, discussions, learning objects or other students with the same interest. RS help Virtual Learning Environments (VLE) like Moodle (Modular Object Oriented Developmental Learning Environment) to provide better interaction.

This paper proposes a recommender system for the Moodle platform. The RS uses as engine the Apache Mahout. In this system, students receive personalized and adaptive recommendations based on forum discussions and learning materials available in the environment, according to their interactions. The recommendation method developed is based on the Moodle's activity log, which is stored in a relational database, and is used as the basis to generate the computation of the recommendation utility, also based on users` preferences on items.

The paper is structured as follows. In the next section we discuss some related works. After, in section 3, we present the proposed recommendation module for Moodle, describing its components, algorithms and recommendation model. Then we provide some evaluations that are performed on synthetic data. Finally, we present some concluding remarks, stating current limitations and future work.

## 2 RELATED WORKS

The first recommender systems appeared in the decade of 1980 and 1990, (D., 1981), (Resnick et al., 1994) and (Hill et al., 1995). Later, many studies and articles related with VLE recommendation were written.

For instance, Olga (2008) describes a recommendation model that is applied to learning scenarios. Such model was conceived based on empirical results considering some usability and accessibility criteria. It not only presents recommendations, but also explains to the user why the recommendation was performed. This is an important feature in terms of systems focused on e-learning, since it helps learners to have a better understanding of what is going on and have a better learning performance. The system proposed here,

Carvalho D., Tadeu Queiroz de Moraes C., Barone D. and Krug Wives L..
RECMOODLE - AN EDUCATIONAL RECOMMENDER SYSTEM.
DOI: 10.5220/0003939802410246
In *Proceedings of the 4th International Conference on Computer Supported Education* (CSEDU-2012), pages 241-246
ISBN: 978-989-8565-06-8
Copyright © 2012 SCITEPRESS (Science and Technology Publications, Lda.)

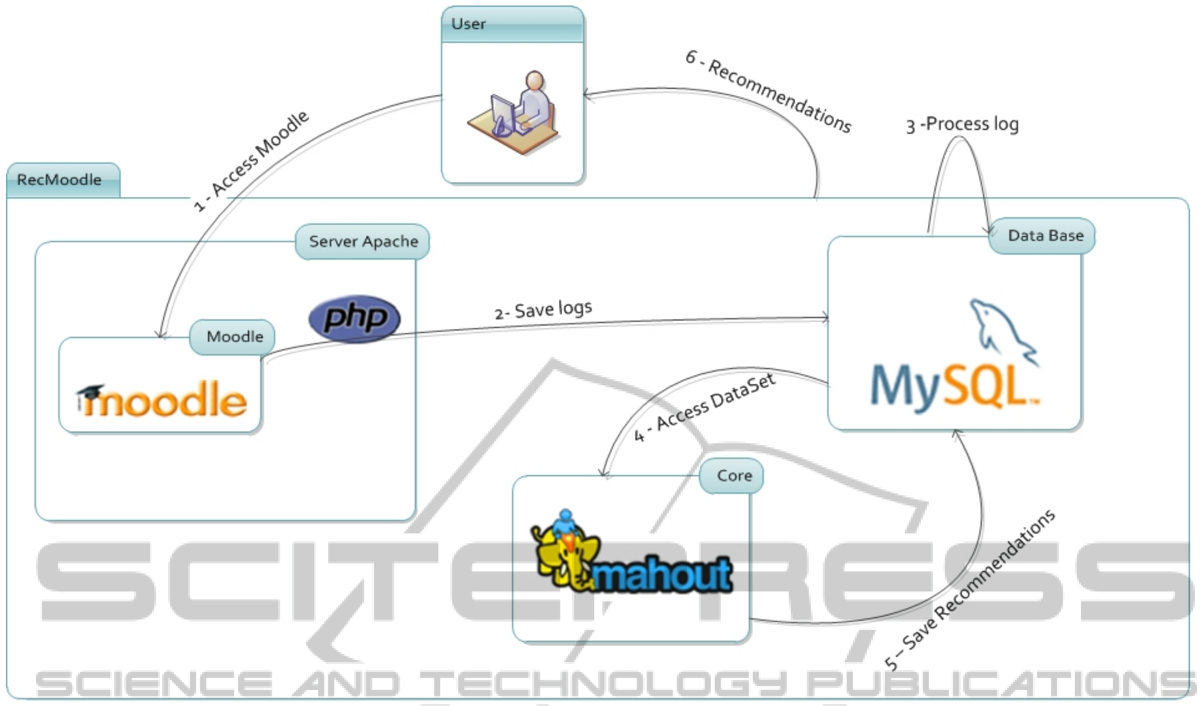


Figure 1: RecMoodle´s general view.

i.e., RecMoodle, uses statistical metrics to generate recommendations and can generate recommendations for users and items.

Another interesting tool is described by Fischer (2005). He developed a tool that can recommend postings in a discussion forum. The idea is that users in a forum sometimes post questions that were already answered or discussed. Then, before posting the question again, the system is able to detect and recommend messages with similar content. If the user is satisfied with the recommendations, the message is discarded. If the user is not satisfied, then the message is posted. The tool considers only the contents of the messages and not the user profile (i.e., interests). In this sense, RecMoole provides recommendations that are based on each user`s interaction with the system.

Barcellos (2009) presents a system that is able to identify users` interests implicitly, using data mining techniques. Our system makes recommendations based on the user profile, but also according to their interactions.

Finally, Cazella (2009) presents a learning object recommender system that is based either on contents and user competences. Its underlying recommender model allows a user to receive recommendations about learning objects that are related to their interests and competences that must be better developed. These competences are stated in a course plan. The main problem of this approach is that the user should explicitly mention their interests.

However, any intelligent recommender system must be able to adapt to the users´ needs in a collaborative manner (Tang & Mccalla, 2005). That means that the user opinion has great value. According to McSherry (2004), the research on RS has a growing conscience that the recommendation process must be more transparent to its users.

RecMoodle not only analyzes activity logs, but also makes use of the Apache Mahout as a recommender engine, thus generating relationships of students and disciplines by the means of adaptive suggestions using techniques that are based on a utility function.

## 3 RECMOODLE

E-learning systems like Moodle store different information about the activities performed by the learners in the environment. Specifically on Moodle, any activity can be logged, e.g., the page being displayed, a message in a forum, the activity being accomplished, etc. The system stores this information in a database containing user information organized by access and activity. Based on this, it is possible to recover the sequence of steps performed by any given user in the environment.

Due to the amount of data daily generated by the system, teachers need specific tools to aid them manage courses. Moodle provides a report tool that shows all the user activities. Such data can be used to understand user interaction but as there is too much information, especially considering the amount of students, it is complicated to manually analyze it.

Data Mining (DM) and Recommender Systems techniques aid the teacher in this process. DM is a pattern identification process that can be applied on large datasets (Maimon & Rokach, 2005) , and can be used by instructors to understand students´ patterns and also to evaluate their activities. A RS generates suggestions to learners about the activities, resources, paths or other users who may be relevant to them.

The proposed system was developed as an extension to Moodle. It is composed of the Moodle itself, a database and Apache Mahout (Figure 1). The database, as already stated, is used by Moodle to store information about the users. Apache Mahout is a powerful framework to perform mining techniques such as clustering and classification, and is used by the recommender module as explained in section 3.2.

The processing steps of our approach are the following:

1. Data cleansing: information from the database is transformed into a format appropriated to be processed by Mahout using an automated task that is triggered when there is any inclusion of an activity in the log table of Moodle;

2. Mining: once the data is transformed, the recommendation module activates Mahout, which analyses the data and generates recommendations that are stored again in the database;

3. Recommendation: according to the action that the user is performing on Moodle, the recommendation module use the data generated by Mahout to show recommendations to the user.

## 3.1 Preprocessing

Moodle follows a structure based on topics for the courses. When organizing a course, one is able to add personalized modules, static material (e.g., web pages, hyperlinks), interactive material (e.g., tasks, blog, questionnaires) and cooperative activities (e.g., chat, forum, glossary, wiki). Each user access to these activities is logged into the database. In our case, MySQL was used.

RecMoodle does not use the Moodle database directly. Instead, it uses specific tables that are prepared in a format that Mahout understands. Thus, a preprocessing must be performed. This is conducted by specific triggers programmed by us to collect data and store in a table named Tab_rec (Figure 2).

In this table, course_id identifies the course accessed by the user. User_id is the user identification. Item_id represents the element accessed by the user inside Moodle. Preference represents the user preference on this item, and timestamp corresponds to the date and the time the user has accessed this item.

By this table we are able to know the order of access or sequence of items the user has followed in a give curse. We are also able to know the most accessed items and the items the user has not accessed.

Preference means how much the user liked or not a specific item or element in the course. It is based in a five points' Likert scale, in which the least significant is 1 and the most is 5.

In Mahout, the values of de preferences can be omitted (e.g., like or dislike, accept or not accept, access or not access or even without a score), or he value's preference can be an implicit preference (when the user simply states that he likes something given a score).

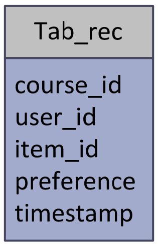


Figure 2: Pre-processing table.

In the left size of Figure 3 we show a situation in which the user has single visit to an item. In this case, it may be a Boolean value. In the right of the figure, the user scored that his preference for the item clicked.

In the case of RecMoodle, four types of preference representation were taken into consideration. None of them is implicit. The first one is based on the order of item access followed by each student and the resulting performance of this student on the course. Table 1 gives an example of this kind of representation.

Each time a user access one item, this item is put on this table, in order of access and do not recount the times they were accessed. Based on this table, it is possible to infer the similarity among users. For instance, in Table 1, users A1 and A3 have

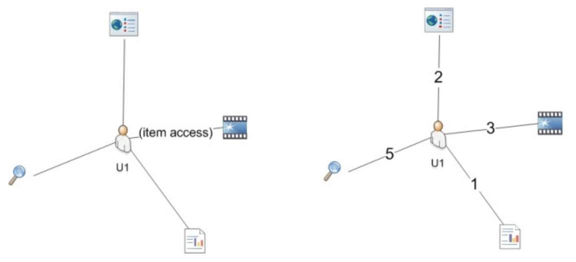


Figure 3: User values omitted left and values' associations on the right.

accessed similar items, but at different orders. Then, these users have no similarity. Comparing these sequences is a difficult task because there may be many differences between the logs. An example of this kind of problem can be seen while comparing U1 and U2 in Table 1. They access the same items, but they are different because the sequence of accesses is distinct. However, it allows us to, for instance, recommend a successful path (order of activities) to new users or to users that are following a path that is known to lead to a bad performance.

The second model consists on stating the items that were visited by the students, independently of their order of access or how many times they were accessed (see Table 2).

Table 1: Users vs sequence of items accessed.

| | Seq_1 | Seq_2 | Seq_3 | Seq_4 |
|---|---|---|---|---|
| U1 | Item1 | Item1 | Item2 | Item3 |
| U2 | Item1 | - | - | - |
| U3 | Item3 | Item2 | Item1 | Item1 |

Table 2: Items accessed by each user.

| | I1 | I2 | I3 | I4 | I5 | I6 | I7 |
|---|---|---|---|---|---|---|---|
| U1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| U2 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| U3 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |

The problem of the representation used in Table 2 is that it is impossible to know the most visited item by each user neither how many times a specific user visited any given item. So it's not possible, for example, know what the five most accessed by the user or which items are most accessed. Another problem is the accuracy of the recommendations, which, in our case, was only 47%.

The third representation model takes into consideration the number of items accessed by each user and normalizes the number of access of each item using Equation 1.

$$\frac{x_i}{\sum_{i=0}^{n} P_n} \tag{1}$$

In this equation, for each user, $x_i$ represents the number of times item $i$ was accessed. P is the frequency of each item, and $n$ is the number of items. For instance, given the values presented in Table 3, Table 4 shows the result of the application of Equation 1.

Table 3: Users versus number of accesses on each item.

| | I1 | I2 | I3 | I4 |
|---|---|---|---|---|
| U1 | 3 | 1 | 0 | 4 |
| U2 | 3 | 1 | 0 | 1 |
| U3 | 2 | 0 | 0 | 0 |

Table 4: Results of the application of equation 1.

| | I1 | I2 | I3 | I4 |
|---|---|---|---|---|
| U1 | 0,375 | 0,125 | 0,000 | 0,500 |
| U2 | 0,600 | 0,200 | 0,000 | 0,200 |
| U3 | 1,000 | 0,000 | 0,000 | 0,000 |

We associate items to users by computing their similarity using Equation 1, which is applied to the data in Table 3, and generating Table 4.

Based on this information, it is possible to assess the similarity among users and perform recommendations. The more data we have, the better will be the assessment. This assessment is explained in the next section.

## 3.2 Users' Similarity Assessment

RecMoodle can be seen as a utility-based recommender system, following (Burke, 2007) definition.

As already stated, Apache Mahout is the core of our system. It is used to analyze the user versus item matrix explained in the previous section, and, thus, generate recommendations based on users or items.

Mahout is a complete framework for the generation of recommendations. It accesses the data, processes and generates recommendations based on the algorithms chosen by the application developer.

In order to generate recommendations with Mahout, four steps are needed: Mahout must access the DataModel containing users' preferences about the different items (i.e., the matrixes explained in the last section). Then, item or user similarity is assessed based on this DataModel. The items or users are clustered based on its neighborhood. Finally, recommendation is performed based on similar items or users (Owen et al., 2011).

The system access the DataModel table containing the users' preferences by a JDBC connection.

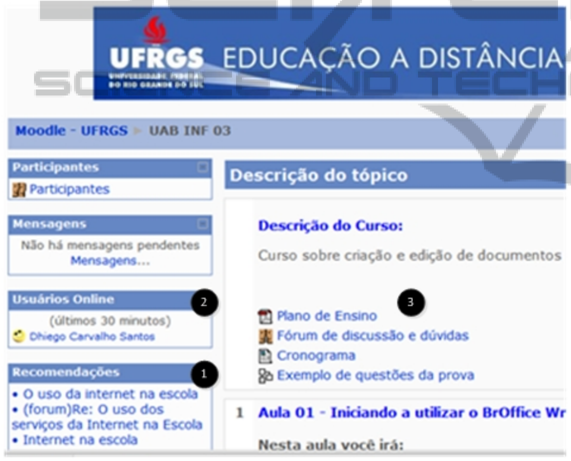
Mahout provides different similarity and recommendation algorithms. We have used the one named *Generic User Based Recommender* (GUBR), based on the *Euclidean Distance Similarity*. The result of this process is as follows:

*user_id [item_id:score, item_id:score, ...].*

In this case, user_id is user identification, item_id:score are the ids of the items accessed by this user and its corresponding scores of preference (by the specific user).

### 3.3 Frontend Application on Moodle

While the user interacts with Moodle, recommendations are generated based on his/her navigation. A frontend module is responsible by showing recommendations to the user. It shows the Top 5 (the five best) recommendations made by Mahout.



Participantes
Não há mensagens pendentes
Mensagens...
(últimos 30 minutos)
Dhiego Carvalho Santos
• O uso da internet na escola
• (forum)Re: O uso dos serviços da Internet na Escola
• Internet na escola
Descrição do Curso:
Curso sobre criação e edição de documentos
Plano de Ensino
Fórum de discussão e dúvidas
Cronograma
Exemplo de questões da prova
1 Aula 01 - Iniciando a utilizar o BrOffice Wr
Nesta aula você irá:

Figure 4: Example of recommendation in RecMoodle.

Figure 4 shows a screenshot of a recommendation made by the RecMoodle module. The black circle at the bottom left is (1) a list of personalized recommendations as messages on the discussion forum of Moodle or links to materials. The circle (2) shows what users are logged on the course and circle (3) the materials that students can access.

## 4 EVALUATION

Evaluations were performed based on the precision assessment that is provided by a Mahout's tool that is known as Recommender Evaluator. The data set used in this evaluation was built based on a real scenario in which the students' interaction logs from a course given at University Federal of Rio Grande do Sul (UFRGS), Brazil, in the period of March 2009 to October 2009. We have recorded the logs from five different classes, summing up 24,986 log lines, 282 items, 158 users and a table of users versus item interaction, as described in Table 4, with 3,422 lines.

For the evaluation of the recommendation algorithm, we have used data as specified in Table 4. We have analyzed some similarity metrics and the quality of the neighborhood algorithm. The *Average Absolute Difference* was used to assess the various user-based recommendation algorithms for *n* users, and using their neighborhood. The closer to zero, the better the evaluation.

Table 5: Average absolute difference to assess the types of similarity using different amounts of neighbourhood.

| Similarity | n=2 | n=5 | n=10 | n=20 |
|---|---|---|---|---|
| Euclidean | 0,0231 | **0,0204** | 0,2837 | 0,2891 |
| Pearson | 0,1880 | 0,2709 | 0,2897 | 0,2877 |
| LogLikel | 0,5301 | 0,2848 | 0,2741 | 0,2820 |
| Tanimoto | 0,4987 | 0,2438 | 0,2597 | 0,2763 |

We have also tested other kinds of neighbourhood assessment, i.e., *User Threshold Neighbourhood*. Its computation is based on all the similarities between users.

Table 6: Average absolute difference to assess different types of similarity threshold using threshold-based.

| Similarity | t=0.95 | t=0.80 | t=0.5 | t=0.20 |
|---|---|---|---|---|
| Euclidean | 0,0289 | 0,0289 | 0,0289 | 0,0289 |
| Pearson | 0,0292 | 0,0332 | 0,0288 | 0,0287 |
| LogLikel | 0,0323 | 0,0265 | 0,0281 | 0,0281 |

It was concluded that the best recommendations for this database are generated through the user-based recommender, using the Euclidean distance and the fifth-nearest neighbourhood. The recommendations accuracy considering this approach was around 98%.

Finally, it was found that the most frequent recommendations from users are:

- Item recommendation: Internet use in school;
- Forum recommendation: The use of Internet services in the School;
- Item recommendation: Registries and domain on the web;
- Item recommendation: Internet at school.

## 5 CONCLUSIONS

User preferences can be represented in different forms such as access, grades, etc. In proposed approach, it was possible to reach 98% of accuracy, while using the Boolean approach the accuracy was only 42%.

In the future, we will use different strategies to achieve better recommendations. Besides, we also want to take into consideration the courses' contents besides users' preferences.

## ACKNOWLEDGEMENTS

We'd like to thank the Institute for Informatics, Federal University of Rio Grande do Sul, for the support.

## REFERENCES

Amazon, 2011. *Amazon*. [Online] Available at: http:// www.amazon.com [Accessed 1 November 2011].

Barcellos, C. D., Musa, D. L., Brandã, o. A. L. & Warpechowski, M. e. a., 2007. Sistema de recomendação para apoio a aprendizagem. In *CINTED-UFRGS*. Porto Alegre/RS, 2007.

Burke, R. D., 2007. Hybrid Web Recommender Systems., 2007.

Cazella, S. C., Reategui, E. B., Machado, M. & Barbosa, J. L. V., 2009. Recomendação de Objetos de Aprendizagem Empregando Filtragem Colaborativa e Competência. *Anais do Simpósio Brasileiro de Informática na Educação SBIE*.

D., M. J., 1981. *Approximation Theory and Methods*. Cambridge University Press.

Fischer, C. G. & Wives, L. K., 2005. Recomendação de conteúdo em fóruns eletrônicos. *Fórum de Inteligência Artificial*.

Hill, W., Stead, L., Rosenstein, M. & Furnas, G., 1995. Recommending and evaluating choices in a virtual community of use., 1995. ACM Press/Addison-Wesley Publishing Co.

Maimon, O. & Rokach, L., eds., 2005. *The Data Mining and Knowledge Discovery Handbook*. Springer.

McSherry, D., 2005. Explanation in Recommender Systems. *Artif. Intell. Rev.*, 24(2), pp.179-97.

NetFlix, 2011. *Netflix*. [Online] Available at: http://www. netflixprize.com/ [Accessed 1 November 2011].

Owen, S., Anil, R., Dunning, T. & Friedman, E., 2011. *Mahout in Action*. 1st ed. Manning Publications.

Resnick, P. et al., 1994. GroupLens: an open architecture for collaborative filtering of netnews., 1994. ACM.

Santos, O. C. & Boticario, J. G., 2009. Building a knowledge-based recommender for inclusive eLearning scenarios. Amsterdam, The Netherlands, The Netherlands, 2009. IOS Press.

Tang, T. & Mccalla, G., 2005. Smart Recommendation for an Evolving e-Learning System: Architecture and Experiment. *International Journal on e-Learning*, 4(1), pp.105-29.