

**RECOMMENDATION SYSTEM BASED ON ASSOCIATION RULES
FOR DISTRIBUTED E-LEARNING MANAGEMENT SYSTEMS***MIHAI GABROVEANU**Department of Computer Science, University of Craiova, Romania,
mihaiug@central.ucv.ro*

Abstract: *Traditional Learning Management Systems are installed on a single server where learning materials and user data are kept. To increase its performance, the Learning Management System can be installed on multiple servers; learning materials and user data could be distributed across these servers obtaining a Distributed Learning Management System. In this paper is proposed the prototype of a recommendation system based on association rules for Distributed Learning Management System. Information from LMS databases is analyzed using distributed data mining algorithms in order to extract the association rules. Then the extracted rules are used as inference rules to provide personalized recommendations. The quality of provided recommendations is improved because the rules used to make the inferences are more accurate, since these rules aggregate knowledge from all e-Learning systems included in Distributed Learning Management System.*

Key words: association rules, educational data mining, recommendation system, LMS, e-Learning

1. Introduction

Learning Management Systems (LMS) have become very popular in the last decade. A Learning Management System (or e-Learning Management System) is a software application used to deliver education content, providing functions for course administration and student registration, grade and track assignments, collaboration between users etc. The Learning Management Systems are used in colleges and universities to deliver online courses; also the corporations use these systems to provide training materials to their employees. Today there are many e-Learning systems available either open source software like Moodle (<https://moodle.org/>), .LRN (<http://www.dotlrn.org/>), ATutor (<http://www.atutor.ca/>), or commercial software such as BlackBoard (<http://www.blackboard.com/>).

Traditional learning management systems are installed on a single server (service provider) where learning materials and user data are kept. To increase its performance, the Learning Management System can be installed on multiple servers (multiple service providers) and learning materials and user data are distributed across these servers obtaining a Distributed Learning Management System (DLMS).

All these LMSs record information about courses, user's activities and academic performances etc. This information can be analyzed using data mining techniques in order to extract useful knowledge that can be used to improve the educational process.

This article describes the prototype of a recommendation system for Distributed Learning Management System able to provide personalized content and advise users to perform some actions. For example:

- a student may be advised to enrol in a particular course since he attended other courses or the skills he gained at the attended courses are similar with the ones students that already choose this course;
- a student can be advised to check some specific teaching materials in order to improve the necessary skills to pass a specific quiz.

This recommendation system is an extension of the system introduced in a previous work [1] that achieves the recommendations in the context of a centralized LMS. In this case, recommendations are made taking into account information from all LMS, which form the Distributed Learning Management System.

The system recommendations are made according authenticated users. For example may be useless to suggest enrolling to a specific course on selection courses panel, since the user already followed that course, but for others users this can be useful advice.

The rules used in the reasoning process by the recommendation system are obtained using a distributed data mining algorithm. This extracts association rules from DLMS activities logs, then rules are translated in Jena Rules, and finally these rules are used to infer additional information.

The remainder of this paper is organized as follows: section 2 shows the steps needed to extract association rules from LMS activity logs, section 3 describes the inference engine that will be used in the recommendation system. In section 4 is presented the architecture of our recommendation system and the paper ends with conclusions and future work.

2. Mining association rules from LMS activities logs

All e-Learning Management Systems store in database information about learner profiles, the courses they are enrolled, test results and log activities. Data mining provides efficient techniques and algorithms to automatically analyze the information collected in these databases in order to extract interesting, useful and new knowledge. Teachers can use the extracted knowledge to improve the courses quality or to build recommendations for learners.

Association rule mining is a process of discovering interesting associations and/or correlation relationships among large set of data expressed in the form of "if-then" statements. In our case, we mine the LMS activities logs in order to extract possible relations between courses, resources, student activities. The mining process can produce a large number of rules, and it is useful to perform a data selection in order to extract only relevant association rules for the inference process.

2.1 Association rule basic concepts

In the following is introduced the formal definition for association rules [2]: Let $I = \{I_1, \dots, I_n\}$ a set of items. Given a database $DB = \{T_1, \dots, T_n\}$ containing a set of transactions, $T_i \subseteq I$, an *association rule* is an implication of the form $A \Rightarrow B$, where A and B are sets of items and $A \cap B = \emptyset$. The meaning of rule $A \Rightarrow B$ is the following: *when A is contained in a transaction, B is likely to be contained as well with a high probability.*

The *support* of rule $A \Rightarrow B$ is defined as *the percentage of transactions in DB that contain both A and B* and *confidence* of rule is defined as *the percentage of transactions in DB containing A which also contain B*. The formal definitions of these measures are [3]:

$$supp(A \Rightarrow B) = \frac{|\{T_i \mid A \cup B \subseteq T_i, T_i \in DB\}|}{|DB|}$$

$$conf(A \Rightarrow B) = \frac{|\{T_i \mid A \cup B \subseteq T_i, T_i \in DB\}|}{|\{T_i \mid A \subseteq T_i, T_i \in DB\}|}$$

The problem of mining association rules consist in identifying all rules having support and confidence greater than or equal to the user-specified minimum support and a minimum confidence. These rules are called *strong association rules*.

Table 1 shows the list of courses followed by students of a LMS obtained after a pre-processing task [1]. We can observe that student 1 followed, *Web Applications* (WA), *Web Documents* (WD) and *Web Technologies* (WT) courses, the student 2 followed *E-Business Technologies* (EBT) and *Web Applications* (WA) courses etc.

Table 1: Student Courses List [1]

Student ID	List of courses
1	WA, WD, WT
2	EBT, WA
3	EBT, WD, WT
4	EBT, WA, WD
5	WD, WT
6	EBT, WT

Let us consider the following association rule $WT \Rightarrow WD$. The meaning of this rule is "some of the students who were enrolled to *Web Technologies* (WT) course, also followed the *Web Documents* (WD) course". The *support* of $WT \Rightarrow WD$ rule is computed as:

$$supp(WT \Rightarrow WD) = \frac{|\{1,3,5\}|}{|DB|} = \frac{3}{6} = 0.50$$

expressing that 50% of students followed both the Web Technologies course and the Web Documents course.

The confidence of rule can be calculated as:

$$\text{conf}(WT \Rightarrow WD) = \frac{|\{1,3,5\}|}{|\{1,3,4,5\}|} = \frac{3}{4} = 0.75$$

and it express that: from all students who follow the Web Technologies course, 75% of them also followed the Web Documents course.

2.3 Mining association rules from DLMS

Since the introduction of association rules, many algorithms to extract strong association rules have been proposed. The most popular algorithms are: Apriori [4], DHP [5], PARTITION [6], DIC [7]. These algorithms require local (central) storage of data. But, in our case, the databases containing the activities data are distributed on each node of DLMS. The Distributed Data Mining systems offer us various techniques of analysis of these data.

In the following, we consider that the Distributed e-Learning Management System (DLMS) consists of n traditional systems and the learning content (courses, tests, resources etc.) are the same on all systems.

Let DB_1, \dots, DB_n the activities databases of the learning systems LMS_1, \dots, LMS_n included in the DLMS. Because we want to extract *global association rules* that involve a distributed database we use a distributed association rules algorithm like *CD (Count Distribution)* [8] algorithm for mining boolean distributed associations rules. An improvement of *CD* algorithm for mining boolean association rules is *DMA (Distributed Mining of Association Rules)* algorithm [9]. During the last years, some algorithms for mining boolean association rules based on MapReduce programming model was proposed [10], [11], [12].

2.2 Mining Logs to extract useful data

Like any KDD [13] process, mining association rules from a Learning Management Systems [14] database involves the following steps:

- *Collecting data.* The LMS keep in a relational database detailed logs with user activities, user profiles, academic results etc.
- *Data pre-processing.* After data selection (identifying data used in the mining process), we can perform tasks like discretization, derivation of new attributes, create summarization tables, and transform the data into a format required by the used data mining algorithms.
- *Applying the mining algorithms.* In this phase a specific implementation of association rules algorithm is applied on the data prepared on the previous step with the right parameters (like support and confidence threshold). Also, we can specify some other restrictions, such as the maximum number of items and what specific attributes can be present in the antecedent or consequent of the discovered rules. By example, we can add restrictions to produce rules such as:
 - 85% of the students, who followed WT course, also followed WA course.
 - 70% of the students that solve home-works from WT course pass the WA exam.
- *Data post-processing.* The extracted rules are represented in a comprehensible format like PMML[15].

3. Recommendation Inference Engine

In order to perform reasoning we use Jena [16] open source framework that allows reasoning over RDF(S) ([17], [18]). Typically rules used in the reasoning process are obtained from the knowledge of experts and operators. As a result, the rules are limited, subjective and inaccurate. In our system, these rules are automatically induced as boolean association rules starting from LMS activities database.

The strong association rules, obtained in an offline mining process, are translated in Jena rules and then, based on these, the reasoning can be performed. If PMML language is used to represents the association rules then an XSLT transformation can be used to translate them in Jena rules. The translation process of association rules in Jena rules is described in a previous work [1]. For example, if we consider an association rule like “85% of the students who followed Web Technologies (WT) course, also followed Web Application (WA) course” obtained above in the mining process, then we can recommend to the

students, *who already followed the WT course and have not yet followed the WA course*, to choose this course.

Let us consider that the user named Thomas Miller logs in a Moodle e-Learning system and accesses courses selection page. In this case, the Jena rule used to advise user to consider the WA course is the following [1]:

```
[R:
(?x rdf:type moodle:Student)
(?x moodle:username moodle:Tom)
(?x moodle:takenCourse moodle:WT)
noValue(?x moodle:takenCourse moodle:WA)
->
(?x moodle:followCourse moodle:WA)]
```

This rule is dynamically generated when the user accesses the page, and the triple `(?x moodle:username moodle:Tom)` of rule isn't part of the association rule. The subject *Tom* of the triple is the login *username* of current authenticated user.

During the inference process additional information can be required from the LMS database. The Jena framework has a dedicated module that allows extracting models needed in the reasoning process directly from the database. Based on models extracted from LMS database, RDF triples are created and stored in the working memory. Those triples represent the initial facts base. In the following, Jena rules induced from association rules are loaded into a *RuleStore* object and starts the reasoning. Next, when of inference process is done, the working memory contains new facts obtained by applying rules over initial facts base. Finally, the new information (facts) is used to build useful recommendations.

4. Recommendation System Architecture

In Figure 2 is presented the architecture of the proposed recommendation system. The system contains the following components:

1. *Local Mining Module* – deployed on each node of DLMS. This module run a local mining process that analyze information stored in the local LMS database;
2. *Recommendation Server* – this is a dedicated server that provide recommendations as a Web Service and includes:
 - a. *Global Association Rules Base* – store the global association rules extracted from distributed LMS databases;
 - b. *Jena Rules Translator* – this module use a XSLT transformation in order to translate the association rules to Jena rules;
 - c. *Inference engine* – the reasoning module that provides information needed to build personalised recommendations.
3. *Recommendation Client* – deployed on each node of DLMS. This module is a web service client that communicates with the recommendation server in order to obtain the recommendations.

The *Local Mining module* connects to the LMS local database, retrieves activities logs, select and prepares data in order to compute the local frequent itemsets needed to extract the association rules. For the mining process, is started a *Local Mining Association Rule* sub-module that run a distributed algorithm for mining global association rules. These sub-modules that run on each node of DLMS collaborate in order to compute all global association rules and finally store in the *Global Association Rules Base* only the strong association rules.

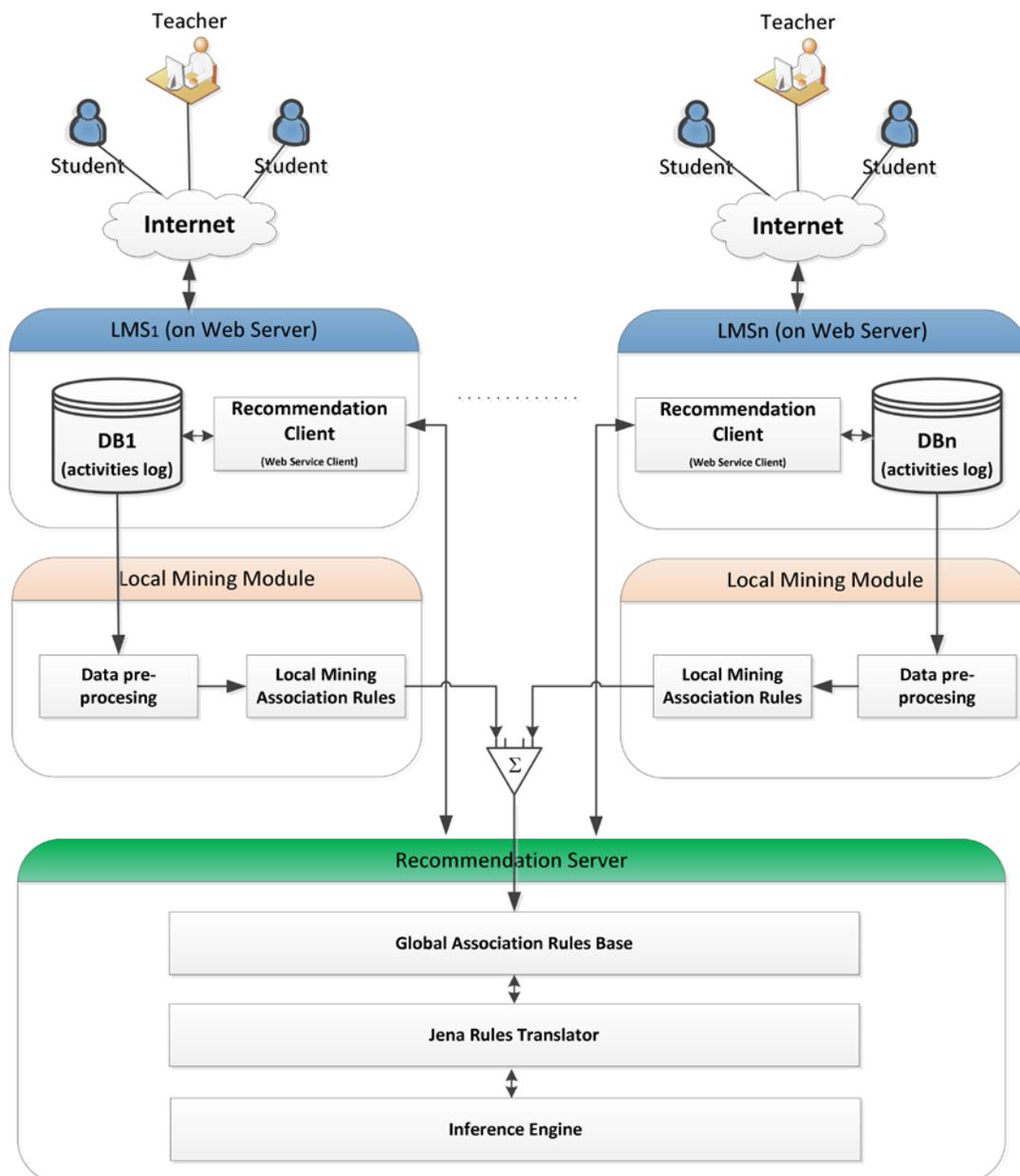


Figure 2: Recommendation System Architecture

Using the *Jena Rules Translator* module, association rules, which are obtained from the mining process, are then translated into Jena rules. The inference engine is configured to run as a servlet, it uses Jena API to connect to the database to extract the models and rules obtained before in order to obtain new information. Finally, on each LMS of DLMS the *Recommendation Client* connects to the *Recommendation Server* using a REST protocol and extracts the possible recommendations obtained after the inference process.

5. Conclusions and future work

The paper describes the architecture of a recommendations system able to provide personalized recommendations in a Distributed Learning Management System. In our system, rules used in the reasoning process are automatically induced as boolean association rules starting from information kept in all LMS databases. Beside traditional recommendation systems, which use only knowledge (rules) extracted from local database, the extracted rules aggregate in this case the knowledge from all nodes of the DLMS.

Future work includes using fuzzy association rules instead of boolean association rules and other data mining techniques like sequential patterns mining, clustering in order to provide recommendations.

6. Acknowledgment

This work was supported by the strategic grant POSDRU/159/1.5/S/133255, Project ID 133255 (2014), co-financed by the European Social Fund within the Sectorial Operational Program Human Resources Development 2007-2013.

7. References

1. Gabrovanu, M., Diaconescu, I. M., Extracting Semantic Annotations from Moodle Data, *Proceedings of the 2nd East European Workshop on Rule-Based Applications (RuleApps 2008)* at the 18th European Conference on Artificial Intelligence (ECAI 2008), pp. 1-5, Patras, Greece, (2008).
2. Agrawal, R., Imielinski, T., Swami, A. N., Mining association rules between sets of items in large databases, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, Buneman, P., Jajodia, S. eds., ACM Press, pp. 207-216, Washington, D.C., May 26-28, (1993).
3. Tan, P.-N., Steinbach, M., Kumar, V., *Introduction to Data Mining*, Addison-Wesley, ISBN 0321321367, (2005).
4. Agrawal, R., Srikant, R., Fast algorithms for mining association rules, *Proceedings of the 20th International Conference Very Large Data Bases, (VLDB)*, Morgan Kaufmann, pp. 487-499, (1994).
5. Park, J. S., Chen, M. S., Yu, P. S., An effective hash-based algorithm for mining association rules, *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, (SIGMOD '95)*, pp. 175-186, San Jose, Canada, (1995).
6. Savasere, A., Omiecinski, E., Navathe, S. B., An efficient algorithm for mining association rules in large databases, *Proceedings of 21th International Conference on Very Large Data Bases (VLDB'95)*, Dayal, U. Gray, P. M. D. , Nishio, S. eds., Morgan Kaufmann, pp. 432-444, (1995).
7. Brin, S., Motwani, R., Ullman, J. D., Tsur, S., Dynamic itemset counting and implication rules for market basket data, *Proceedings ACM SIGMOD International Conference on Management of Data*, ACM Press, pp. 255-264, (1997).
8. Agrawal, R., Shafer, J. C., Parallel mining of association rules, *IEEE Transactions on Knowledge And Data Engineering*, vol. 8, pp. 962-969, (1996).
9. Cheung, D. W., Han, J., Ng, V. T., Fu, A. W., Fu, Y., A fast distributed algorithm for mining association rules, *Proceedings of the 4th International Conference on Parallel and Distributed Information Systems (PDIS '96)*, IEEE Computer Society Technical Committee on Data Engineering, and ACM SIGMOD, pp. 31-43, (1996).
10. Li, L., Zhang, M., The strategy of mining association rule based on cloud computing, *Proceedings of the International Conference on Business Computing and Global Informatization, (BCGIN '11)*, IEEE Computer Society, pp. 475-478, Shanghai, (2011).
11. Lin, J., Dyer, C., *Data-Intensive Text Processing with MapReduce*, Morgan and Claypool Publishers, (2010).
12. Lin, M. Y., Lee, P. Y., Hsueh, S. C., Apriori-based frequent itemset mining algorithms on MapReduce, *Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication (ICUIMC '12)*, ACM , pp. 76:1-76:8, New York, NY, USA (2012).
13. Han, J., *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, (2005).
14. Garcia, E., Romero, C., Ventura, S., Calders, T., Drawbacks and solutions of applying association rule mining in learning management systems, *Proceedings of the International Workshop on Applying Data Mining in e-Learning (ADML'07)*, pp. 13-22, Crete, Greece, (2007).
15. Predictive Model Markup Language (PMML) v.4.2.1 Specification, DMG, <http://www.dmg.org/pmml-v4-2-1.html>, (2014)
16. Apache Jena, <http://jena.apache.org/>, (2015)
17. Brickley, D., Guha, R.V., *RDF Schema 1.1, W3C Recommendation February 2014*, <http://www.w3.org/TR/rdf-schema/>, (2014).
18. Cyganiak, R., Wood, D., Lanthaler, M., *RDF 1.1 Concepts and Abstract Syntax, W3C Recommendation, 25 February 2014*, <http://www.w3.org/TR/rdf-concepts/>, (2014).