MDM Tool: A Data Mining Framework Integrated Into Moodle

J. M. LUNA, C. CASTRO, C. ROMERO

Department of Computer Science and Numerical Analysis, University of Cordoba, Cordoba 14071, Spain

Received 17 July 2016; accepted 27 November 2016

ABSTRACT: The interest in developing Learning Analytics tools that can be integrated into the well-known Moodle course management systems is increasing nowadays. These tools generally provide some type of basic analytics and graphs about users' interaction in the course. However, they do not enable a varied set of Data Mining techniques to be applied, such as approaches for classification, clustering, or association. To address this issue, a new and freely available Moodle Data Mining tool, named MDM, has been proposed in this paper. The proposed tool eases the whole knowledge discovery process, including tasks such as selection, data pre-processing, and data mining from Moodle courses. The proposed MDM tool has been developed in PHP programming language, so it can be easily integrated into Moodle as a module for a specific course. Its main features and architecture are described in depth, and a tutorial is also provided as a practical way of using the MDM interface. Finally, some experimental results using a real-life sample dataset of mechanical engineering students are analyzed. © 2017 Wiley Periodicals, Inc. Comput Appl Eng Educ 25:90–102, 2017; View this article online at wileyonlinelibrary.com/journal/cae; DOI 10.1002/cae.21782

Keywords: data Mining; clustering; classification; association; moodle

INTRODUCTION

Moodle (Modular Object-Oriented Dynamic Learning Environment) is one of the most popular open source Course Management Systems (CMS). It is written in PHP programming language and distributed under the GNU General Public License. Moodle was created by Martin Dougiama to help educators to create online courses with a focus on interaction and collaborative construction of content, and it is in continual evolution [1]. Moodle is freely available and used in more than 225 countries around the world, translated into over 100 languages. Universities, K-12 schools, businesses, and individual instructors use it to deliver online courses and to supply traditional face-to-face courses. The modular design of Moodle makes it easier to create new courses by adding content that will engage learners. It provides a powerful set of web-based tools for an array of activities and resources, such as forums, messaging, tests, assignments, wiki, blogs, glossaries, etc. [2]. It also has a very large active community of people who are using the system and developing new features and enhancements. In fact, more than 1,000 complementary blocks, modules, and plug-ins can be found, extending the features of Moodle's core functionality.

Moodle gathers large amounts of information about students' interactions. This information gives an idea of how and when students perform their assignments, tasks, course engagements, etc. Moodle's default reporting tool offers information and filtering capabilities, but the data they provide is considered raw data and does not provide meaningful information about the course [3]. Extracting this type of useful data and transforming such information into actionable knowledge is a difficult task. New emerging disciplines such as Educational Data Mining (EDM) and Learning Analytics (LA) offer both different and convergent perspectives, methodologies, techniques, and tools aiming to facilitate the process of discovering knowledge from educational data. EDM is defined as the development of methods for exploring the unique types of data that come from educational settings, and using those methods to understand students and what and how they learn [4]. On the contrary, LA is defined as the measurement, collection, analysis, and reporting of data about learners and their contexts. The purpose of LA is to understand and optimise the learning process and the environments in which it occurs [5]. Although some authors make distinctions between these approaches due to the fact that their focus and routes come from potentially

Correspondence to C. Romero (cromero@uco.es).

different sources/places [6], both LA and EDM communities have the same goal of improving the quality of educational data analysis. EDM and LA have similar roots in digital learning environments and generally these communities deal with the same educational problems. Nowadays, a wide variety of tools are available to facilitate the extraction and analysis of valuable knowledge from educational data, including adhoc analysis of specific issues and more general purpose cross-platforms [7]. Nevertheless, only few of these tools were designed to be used on Moodle data [3]. Furthermore, currently, existing tools only provide a certain level of basic analytics and graphs about users' interaction with the platform but they do not provide any Data Mining (DM) techniques. Hence, although existing tools are very useful to obtain a general view of the status of the course, they are unable to discover new, interesting, and useful knowledge. In order to overcome this drawback, we have developed a new specific tool, named MDM[1], which is able to extract interesting and previously unknown knowledge hidden in Moodle data by means of different DM techniques. Moreover, given that it is integrated in Moodle, any of the DM techniques included in this tool use the same interface than Moodle. Finally, in order to extend the objective of the proposed LA/EDM tool, the following questions arise:

- Is it recommendable a LA/EDM tool for an online learning environment, for Moodle? Yes, Moodle does not provide tools to run LA/EDM but it relies on simple statistics to provide feedback [8]. Moodle collects a vast amount of student's usage information as raw data. Therefore, it offers very detailed information, but it is not able to provide meaningfull information about the course [9]. On the contrary, LA/EDM tools provide algorithms/methods for discovering meaningfull and unknown knowledge from data, that is they transform information into actionable knowledge [10].
- Which are the advantages? And why the LA/EDM tools are good for online learning environments, for Moodle? LA/EDM tools allow to explore, visualize, interpret, and analyze e-learning data in order to help instructors in a better understanding and improvement in their e-learning practice [7]. These tools act on the whole process of analyzing data (from data selection and data pre-processing to DM) in the same learning environment. These tools are more specific and easier to be used by non-expert users in DM than general or traditional DM tools [11]. In this way, instructors are able to discover, in an easy way, hidden information about students' behavior, and how students learn in the courses in a single and user-friendly interface [12].

This paper is organized as follows: Background and related tools briefly reviews existing Learning Analytics and Data Mining tools to be used with Moodle data; Architecture and implementation describes the main features of MDM and its architecture; Description of the tool illustrates the interface of the proposed tool in a practical way and by using a tutorial based on real data. Finally, conclusion and future work provides some concluding remarks and suggests future lines of research.

[1]MDM tool is freely available for donwload http://www.uco.es/grupos/kdis/index.php?option = com_content&view = article&id = 23&Itemid = 60&lang = en

BACKGROUND AND RELATED TOOLS

Up to now, EDM (Educational Data Mining) and LA (Learning Analytics) communities have developed a great number of specific LA and Data Mining (DM) tools to solve educational problems [7]. New tools are costantly emerging in different educational environments: Learning and Management Systems (LMS), Intelligent Tutoring Systems (ITS), Adaptive and intelligent hypermedia systems (AIHS), MOOCs (Massive Open Online Course), test and quiz systems, learning object repositories, educational games, etc. Thus, it is impossible to review or compare our tool versus all of them. For further reading, there are some reviews about tools used for data mining/analytics in the area of education [7,13]. On the other hand, only a few of those tools are specifically oriented to be used with Moodle data. Hereunder is a short description of those specific Moodle tools, which are divided into two subtypes: desktop and web-based tools.

Desktop tools are applications that run locally on a computer device. The oldest EDM and LA tools to use with Moodle data were desktop tools. CoSyLMSAnalytics [14] is an example of these tools, which allows a lecturer and an evaluator to assess and monitor individual or collective progress, so as to provide informative feedback to students. The aim of this tool is to easily evaluate the learners' progress and produce evaluation reports based on learners' behavior. It gathers information concerning learners' access patterns and extracts correlations and association rules among their learning paths. ViMoodle or Visual analytical tool for Moodle [15] provides the additional benefit of understanding educational relationships in Moodle. Its main objective is to extract specific knowledge related to the complex process of education and learning. It uses log activities, quiz, forums, resources, etc., from the database and it returns social network and snail graphs using clustering and association rule mining algorithms. Another sample tool is CIECoF or Continuous Improvement of E-learning Course Framework [16], which provides recommendations to courseware authors about how to improve Moodle courses. Its objective is to find, share, and suggest the most appropriate modifications to improve the effectiveness of the course. It uses association rule mining and collaborative filtering through student's usage data. Meerkat-ED [17] is another tool that enables students' participation in Moodle discussion forums to be analyzed. Its objective is to provide instructor with better means to assess participation in online discussions. It uses social network analysis techniques over forum discussion data. MMT or Moodle-Mining Tool [18] is a different tool that makes executing all the steps in the data mining process. Its objective is to provide DM techniques not only for expert but also for newcomers. It applies pre-processing, classification, and association rule mining algorithms over data files created from Moodle database. An additional desktop tool is DRAL [19], which reveals relevant activities for learners of Moodle. Its objective is to detect the activities that a student needs to pass a course based on features extracted from logged data. It uses multi-objective, grammar-guided genetic programming for classification tasks.

Web-based tools are delivered to a local device over the Internet from a remote server. Currently, almost all new EDM and LA tools are web-based and some examples are described next. GISMO [20] uses graphical interactive student monitoring to extract tracking data from an online course maintained with Moodle. Its objective is to provide visualizations of behavioral, cognitive, and social data from the course, allowing constant monitoring of students' activities, engagement, and learning outcomes. GISMO is integrated as a block in Moodle. SNAPP or Social Networks Adapting Pedagogical Practice [21] is another tool that visualizes the evolution of participant relationships within discussions forums. Its aim is to analyze emergent interaction patterns to allow interventions to be undertaken as required. It is integrated into the Moodle interface as Bookmarklet, but only shows real-time visualization of discussion forum activity. AAT or Academic Analytics Tool [12] accesses and analyses students' behavior data in learning systems. Its objective is to allow the identification of difficult or inappropriate learning material in order to significantly contribute to the design of improved student support activities and resources. Although it is integrated into Moodle as a block, it generates graphical representations that can only be explored by course instructors. MOClog [11] is a tool for the analysis and presentation of log data on a Moodle server. Its goal is to measure the status of activities in the online-course as much as possible by relying exclusively log file data. It reuses components available from GISMO for counting and visualizing the number of pots and clicks. E-learningWebMiner [10] reveals students' behavior profiles and models how they work on virtual courses. Its aim is to provide useful information that instructors may use to improve their courses. It is a web-service that provides visualization graphs, clustering, and association algorithms. CVLA [8] integrates analytics techniques in a custom Moodle report. Its objective is to use multiple data sets and analytics techniques in a single interface for presenting data to learners and educators. It is integrated into Moodle as a module and provides social network analysis and classification algorithms for predicting assignment submission. IntelliBoard.net [22] extracts the statistical data gathered and available in Moodle and presents it on a dashboard in the form of printable charts, graphs, and multiple-format reports. Its goal is to deliver educational data analytics to single dashboard instantly. It is a commercial analytic and reporting Web service and plug-in. SmartKlass [23] measures and analyses the learning process at any time throughout Moodle courses. Its objective is to empower teachers to manage the evolution of the students in an online course. It is an Open source and multi-platform learning analytics dashboard plug-in. MEAP or Moodle Engagement Analytics Plug-in [9] provides feedback on the level of engagement of a student on a Moodle course. Its aim is to identify activities, that have an impact on student success. It is integrated as a block in Moodle. Analytics Graphs [24] graphically summaries students' access in a Moodle unit by providing three graphs that may facilitate pedagogical decisions. Its objective is to facilitate the identification of student profiles and to send messages to users according to their behavior inside a course. It is integrated as a block in Moodle. VeLA or Visual e-Learning Analytics [25] provides different representations of the information and displays it interactively. Its aim is to empower teachers to understand of student learning process. It is a framework that uses web services to extract information from different LMSs such as Moodle.

Table 1 illustrates a comparison of the LA/EDM desktop and web-based tools previously described. In this comparison, we include information such as whether the tool freely available for download; whether the tool is integrated (Integr.) into the Moodle interface; and the programming language used. Additionally, this comparison includes information about the visualization technique (Vis), for example, histograms, pie charts, etc.; pre-processing techniques (Preproc.) example, partition, normalisation, discretisation, etc.; supervised learning techniques (Supervi.) example, classification, regression, etc., and unsupervised learning techniques (Unsupervi.) example, clustering, association, etc.

As it is summarized in Table 1, the number of desktop tools is smaller than web-based tools. It is noteworthy to mention that desktop tools are not freely available for download; they are not integrated into Moodle; and they predominantly use Java as the programming language. Additionally, desktop tools usually include a single LA/DM technique except for MMT, which provides plenty of them. Focusing on web-based tools, it is interesting to highlight that they are more and more used and the number of them is increasing. From the summary shown in Table 1, it is obtained that more than half of existing tools are freely available for download; almost all of them are integrated into Moodle; and PHP is the most widely-used programming language. Besides, although all of them provide visualization techniques, none provides pre-processing techniques, only E-learningWebMiner offers unsupervised learning techniques, and only CVLA provides supervised techniques. Hence, current

Table 1 Comparison of LA and EDM Tools Specifically Oriented for Use With Moodle Data

Tool	Free	Integr.	Language	Vis.	Preproc.	Supervi.	Unsupervi.
CoSyLMSAnalytics	No	No	Visualbasic	No	No	No	Yes
ViMoodle	No	No	Java	Yes	No	No	No
CIECoF	No	No	Java	No	No	No	Yes
Meerkat-ED	No	No	Java	Yes	No	No	No
MMT	No	No	Java	No	Yes	Yes	Yes
DRAL	No	No	Java	No	No	Yes	No
GISMO	Yes	Yes	PHP	Yes	No	No	No
SNAPP	Yes	Yes	Javascript	Yes	No	No	No
AAT	No	No	PHP	Yes	No	No	No
MOClog	Yes	Yes	PHP	Yes	No	No	No
E-learningWebMiner	No	No	Java	Yes	No	No	Yes
CVLA	No	Yes	Phyton	Yes	No	Yes	No
IntelliBoard.net	No	Yes	PHP	Yes	No	No	No
SmartKlass	Yes	Yes	PHP	Yes	No	No	No
MEAP	Yes	Yes	PHP	Yes	No	No	No
Analytics graphs	Yes	Yes	PHP	Yes	No	No	No
VeLA	No	Yes	PHP	Yes	No	No	No

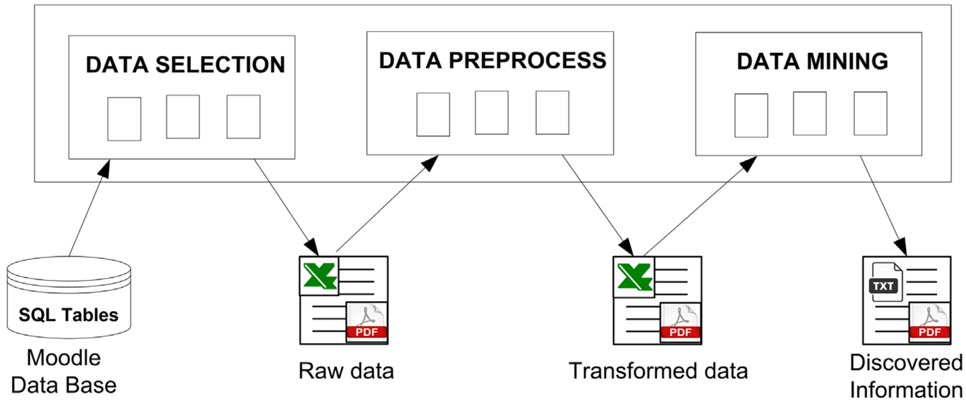


Figure 1 MDM architecture.

web-based tools can be considered as Visual Analytics tools that use different statistics and visualization techniques. In order to overcome this lack of web-based tools for DM purposes, the aim of this paper is to propose and describe a new tool named MDM (Moodle Data Mining tool), developed in PHP for integration into Moodle as a block. MDM is added into Moodle course to provide the instructor with traditional DM techniques such as pre-processing, supervised, and unsupervised learning tasks.

ARCHITECTURE AND IMPLEMENTATION

As previously stated, the proposed MDM tool has been developed in PHP language and integrated into Moodle as a block. The main reason to integrate the proposed tool as a Block is due to any Moodle Block provides the desired functionality requested to our tool. Moodle has available a range of standard blocks and it also enables to extend it by developing your own block. Blocks are items which may be added to the left, right, or centre column of any page in Moodle. They contain information or new applications oriented to students or instructors. In our case, MDM tool is only accessible by instructors which can add a new block to their course by only selecting it from the Block "Add a block" on the course homepage. Then, the new block is automatically added to page. In this way, any instructor can easily install our MDM tool into their Moodle courses.

From a high level viewpoint, MDM architecture comprises three main modules that communicate between each other by means of different files (see Fig. 1).

The proposed MDM tool consists of three consecutive components or steps:

- Data pre-processing enables the data selected in the first step to be suitably prepared so different DM techniques can be applied. Firstly (see Fig. 1), an Excel data must be loaded, then different pre-processing tasks can be applied, and finally the transformed data are saved as Microsoft XLS Excel files (for use in the next step), or as Adobe PDF files (simply for visualization purposes). MDM tool provides a set of basic pre-processing tasks to allow the instructor to transform raw data into a suitable data format ready to be used by different data mining algorithms. In this regard, the following data pre-processing tasks have been considered: edit, anonymize, discretize, and split.
- Data selection enables the specific data that the instructor wants to use to be chosen from a Moodle course. Firstly (see Fig. 1), data about users and resources are selected from the Moodle Database, then the instructor chooses between different types of information and, finally, the raw dataset is saved as Microsoft XLS Excel files (to be used in the next step), or as Adobe PDF files (simply for visualization). Moodle uses a relational database with around 200 tables (all their names begin with a really descriptive word) and the relationships between them. From all these tables, MDM tool is able to extract the following data sources: summary (overall review about the general interaction between students and Moodle), grades (report of marks obtained by all the students in all graded items in the course), logs (historical record or low level student's activity record), forums (report of messages written and read in the forums of the course), quizzes (report of all student's quiz attempts), and resources (report of resources accessed by the student in the course).
- Subsequently, Data mining is used to run knowledge discovery algorithms on the previously pre-processed data. Firstly (see Fig. 1), a pre-processed Excel data file must be loaded, then different data mining techniques can be applied and, finally, the discovered models are saved into TXT files (to be used by external programs), or as Adobe PDF files (to ease the visualization). MDM tool provides a set of basic data mining tasks to allow the instructor to discover knowledge from data. In this regard, the following data mining tasks have been considered: clustering, classification, and association rule mining.

Finally, it is noteworthy to mention that all the code included in the proposed tool was written by ourselves and using PHP scripting language. Hence, we have not used proprietary software neither code from other DM software tools such as Weka or R.

DESCRIPTION OF THE TOOL

The proposed MDM tool is defined as a new block for Moodle that is freely available for download[2]. Moodle courses come with a set of default blocks. However, the user can add additional blocks, remove blocks, or rearrange the blocks in the sidebars. Installing

[2]http://www.uco.es/grupos/kdis/index.php?option = com_content&view = article&id = 23&Itemid = 60

Figure 2 MDM tool block in an example course.

the proposed MDM is similar to any other standard Moodle blocks: (1) the user has to to unzip the downloaded .zip file into the Moodle blocks directory (Moodle_DOCUMENT_ROOT/blocks); (2) the user has to log-on to Moodle as the administrator and go to the notifications page; (3) finally, the user needs to log-on to the course in order to activate the Edit mode and to add the new MDM block. Figure 2 shows the MDM tool block already included into Moodle (on the left side).

As it is illustrated in Figure 2, the MDM interface opens once the instructor clicks on the logo icon or on the blue "MDM Tool" text. All the text/content in the MDM tool can be configured in both Spanish or English language (before opening) by means of an option in the Moodle's top bar (see top left side in Fig. 2). The interface of MDM tool consists of three main tabs or components: data selection, data pre-processing, and data mining. Each one is described in the following sub-sections.

Data Selection

This component/tab enables instructors to select and visualize information about resource usage or the students enrolled on a Moodle course. It provides basic statistics and graphs about the students registered on the course and the resources provided in it. Instructors first select a source (statistics on students or resources) and then one, several, or all the elements (from the available students or resources) by using the "Add" or "Delete" buttons and the "Select All" checkbox (see Fig. 3). Finally, instructors need to select the time period, the type of information they want to recover from the set of statistics available (overall summary, grading, historical record, forums, quizzes, and files/resources) and, then, press the "Show statistics" button.

Finally, it should be noted that the statistics provided by MDM can be saved/exported to an Excel file (for pre-processing or mining purposes) or to a PDF file (for reporting purposes). The Instructor only needs to introduce the file name and the file extension.

Data Pre-Processing

The data pre-processing component/tab allows the instructor to transform a raw data file into a suitable format ready to be used by a data mining algorithm, to solve a specific educational problem. Currently, it provides the following data pre-processing tasks: edit, anonymize, discretize, and split. These specific pre-processing tasks are those required by any instructor to correctly prepare his/her data, in order to be able to run the three DM algorithms available in the tool as we describe next in each corresponding subsection. There is no restriction to apply several of these pre-processing techniques and all of them might be used together. In fact, in many cases, the four pre-processing tasks are required to be applied together for example in order to obtain a classification model.

Edit

Data edition allows the data to be modified by the user. MDM provides an editable table in which instructors can alter both the name of the attributes and their value in any cell. They can also delete a specific student (a full row) from a dataset, or even add the final mark obtained by students in the course (this additional attribute will be used by classification and association algorithms). In fact, the main reason of including this pre-processing task is due to the fact that the students' final mark in the course is not provided by Moodle, and it has to be added by the instructor. Additionally, some students can have all the values to 0, as they did not use Moodle so, they should not be used by any DM technique. These students can be easily deleted by the instructor.

The Instructor needs to open a file (e.g., the previously generated summary.xls), select the option edit from the drop-down menu/list, change a value, select the row or students to be removed, write the final mark, and click on the run button. Then a new Excel file with the edited data is automatically generated.

Anonymize

Data anonymization allows to preserve the privacy of the students, so different items of information are linked to the specific student without explicitly identifying or showing his/her name. A common solution consists of using a number or user ID instead of someone's real name. The MDM tool provides data anonymization by substituting the first two attributes of the student's name and surname with a new numerical attribute or ID. This number is incrementally generated from one to the total number of students on the course. The main reason of including this pre-processing task in MDM tool is because it allows to maintain the students' privacy if the datasets are share with other instructors or researchers.

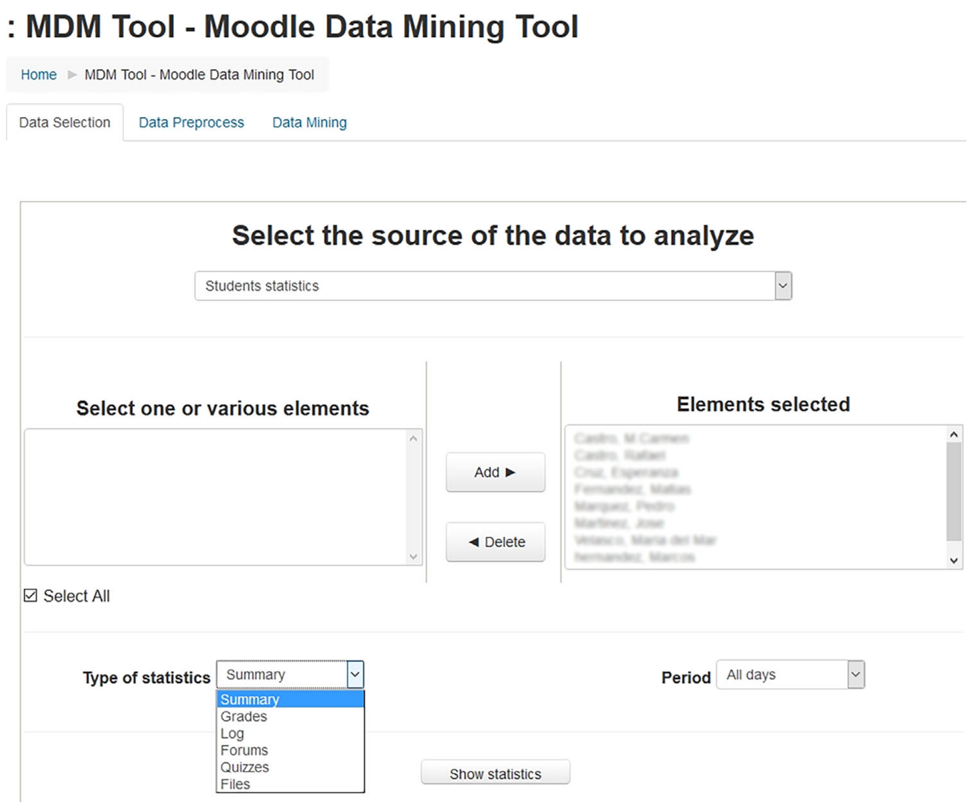


Figure 3 Interface for selecting data.

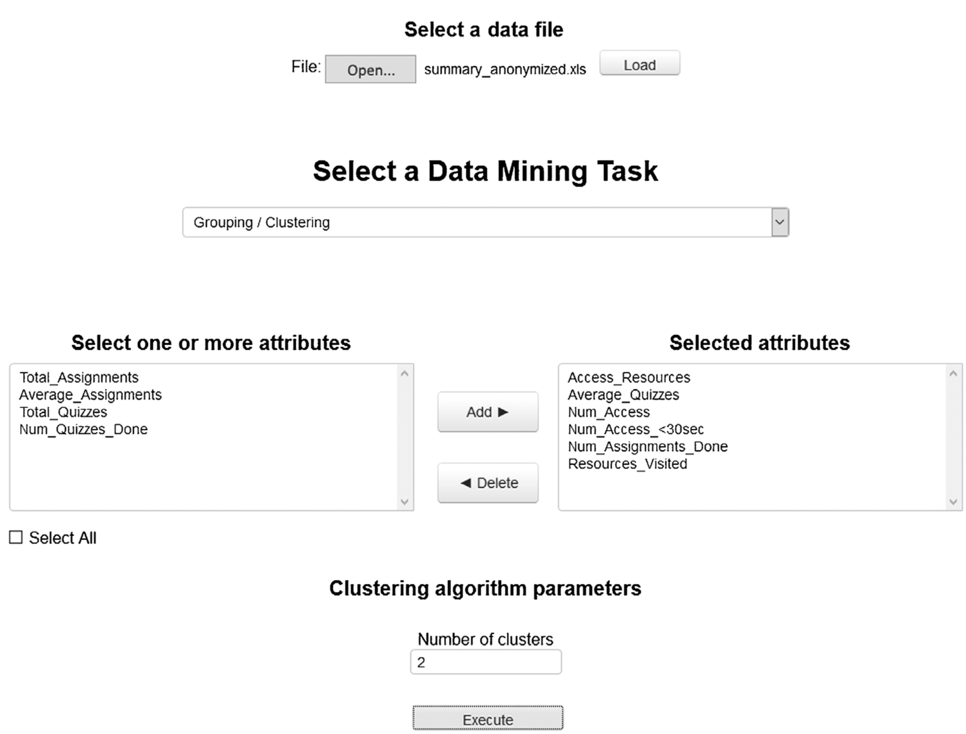


Figure 4 Interface for clustering data.

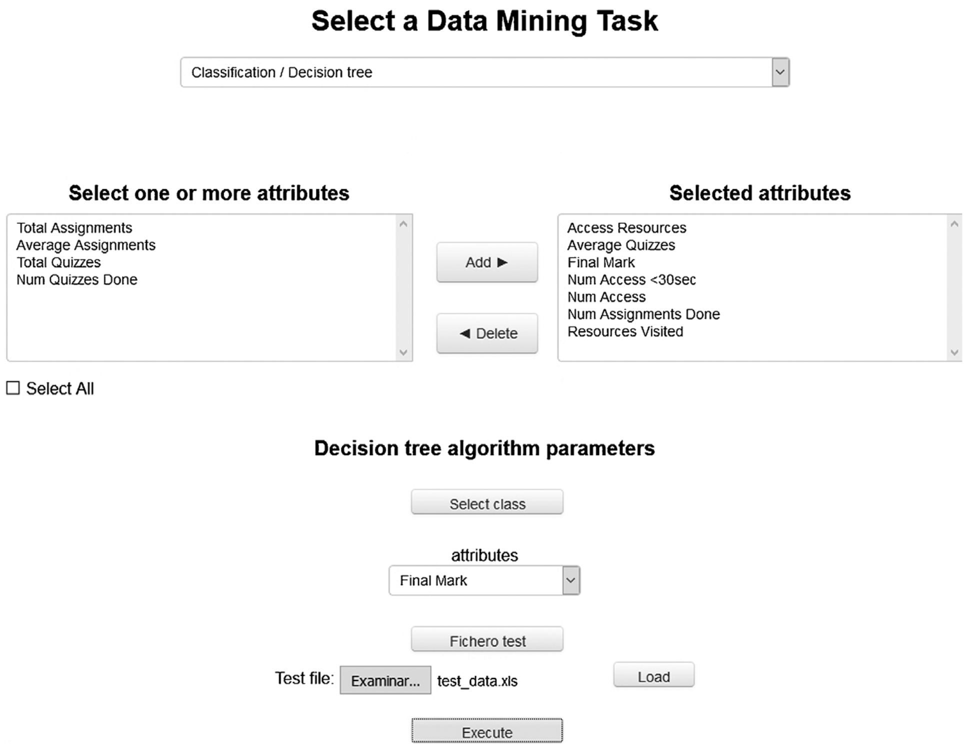


Figure 5 Interface for data classification.

The Instructor only has to open a data file (e.g., the previously edited summary data file), select the option anonymize from the drop-down menu/list, and click on the anonymize button. Finally, a new Excel file with the anonymized data is automatically generated.

Discretize

Data discretization transforms numerical data into categorical labels that are more user-friendly than precise magnitudes and ranges. It reduces the number of possible values for a continuous feature and provides a more understandable/clear view of the data. It can be included as a reduction method, when using DM algorithms that do not work well with continuous attributes. The main reason of including this task in MDM tool is because some classification algorithms and most association rule mining algorithms only work with categorical data. In fact, the implemented ID3 and Apriori algorithms only work with categorical data.

MDM provides three well-known discretization methods [26].

- Equal-width binning, which divides the range of feasible values into N sub-ranges of the same size in which: $bin_{width} = (\max value - \min value)/N$. For example, if values are all between 0 and 100, 5 bins could be created as follows: [0–20], [20–40], [40–60], [60–80], and [80–100].
- Equal-frequency or equal-height binning that divides the range of possible values into N bins, each of which holds the same number of instances. For example, in the event of the following 10 values: 5, 7, 12, 35, 65, 82, 84, 88, 90, and 95. In order to create five bins, the range of values could be divided so that each bin holds two values as follows: [5,7], [12,35], [65, 82], [84, 88], and [90, 95].
- Manual discretization, in which the user is responsible for specifying the cut-off points. For example, if values are all between 0 and 100, 2 bins with cut-off at 40 could be created, e.g., [0–40], [40–100].

The Instructor has to open a data file, select the option discretization from the drop-down menu/list, select the attributes to discretize, select a method, select the number of labels/bins (and cut-offs for manual discretization), and click on the "discretize" button. Then, a new Excel file with the discretized data is automatically generated.

Split

The option of splitting datasets enables a data file to be divided into several data files. The main reason of including this pre-processing task in MDM tool is because it is required by any supervised learning algorithms such as ID3. That is, in classification it is necessary to divide the original dataset into complementary sub-sets or partitions named Training and Test datasets. A Training set is a set of data used to discover potentially predictive relationships. A Test set is a set of data used to assess the strength and utility of a predictive relationship. MDM provides a simple and easy to use splitting method.

The procedure implemented to divide a dataset into training and test sets is as follow. Starting from a dataset and a specific percentage of split, the procedure randomly selects some index numbers (instances from the whole dataset) and then, it uses the set of selected indices to divide the dataset into training and test sets. That is, a random number of instances (students' records) are selected to be included into the training dataset (according to the percentage

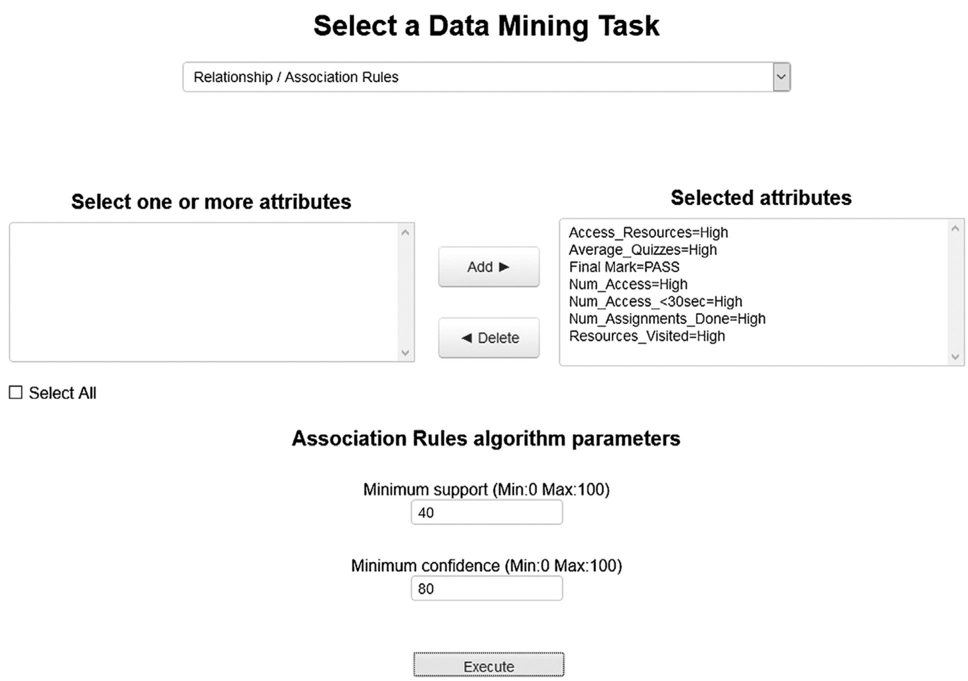


Figure 6 Interface for data association.

provided) and the remaining instances (those not selected for the training dataset) are then added to the test dataset. It should be noted that this procedure has been implemented in such a way that each fold contains roughly the same proportions of the class labels.

Instructors only have to specify the ratio or percentage of randomly selected instances/students to be included in the test file (the remaining instances/students are automatically included in the training file) and click on "run" button. Then, the two Excel data files (test and training) are automatically generated.

Data Mining

This component/tab enables the instructor to apply some typical DM techniques to the previously pre-processed data. Currently, the MDM tool enables three different types of DM methods/tasks to be performed: clustering/grouping, classification/decision tree, and association/relationships.

Clustering

Clustering is a process of grouping objects into classes of similar objects. It is an unsupervised task or partitioning of patterns (observations, data items, or feature vectors) into groups or subsets (clusters). This technique groups a given collection of unlabeled patterns into meaningful clusters. Cluster analysis helps construct meaningful partitions of large sets of objects based on a divide and conquer methodology. The principle of clustering is to maximise similarity within an object group and minimise similarity between object groups.

The MDM tool includes KMeans [27] for clustering [28]. The main reasons for selecting this algorithm are: (1) it is one of the most popular and simplest algorithms for clustering; (2) it is easy to implement; (3) results are easily interpretable by any user (non-expert user); (4) it does not require so many parameters (only the number of clusters to be obtained).

In order to use it, instructors only have to select a data file and Data Mining Task as the Grouping/Clustering option, then select what attributes they want to use to for the clusters, select the number of desired clusters, and finally click the *Run* button (see Fig. 4).

Once the Kmeans algorithm has been run, the Centroid Cluster model obtained is shown and can be saved as a PDF or plain text file. This model has information regarding the clustering performed. It shows which items are part of which cluster. It also has information regarding the centroids of each cluster.

Classification

Classification or discriminant analysis predicts class labels. It is a supervised learning method that provides a collection of labeled (pre-classified) patterns; and the problem is to label a newly encountered, still unlabeled, pattern. Typically, the given labeled (training) patterns are used to learn the descriptions of classes which in turn are used to label new patterns (test).

The MDM tool incorporates the ID3 algorithm [29], which is the precursor to the well-known C4.5 algorithm, and they are typically used to generate a decision tree. The main reasons for selecting this algorithm are: (1) it is a simple classification algorithm; (2) it produces understandable models; (3) it builds short decision trees in a fast way; (4) it does not require any additional parameter.

To use it, instructors only have to select a training data file and the Classification/Decision Tree option as the Data Mining Task; select which attributes they want to segregate by classification; select the class attribute; select training and test files; and, finally, click the *Run* button (see Fig. 5).

Once the ID3 algorithm has been run, the resulting Decision Tree model is displayed and can be saved as a PDF or plain text file. A decision tree is a flowchart-like structure in which each internal node represents a condition on an attribute, each branch represents the outcome of the condition, and each leaf node represents a class label (decision taken after computing all

Table 2 Attributes Included in the Statistics Summary

Name	Description
Lastname and firstname	Name and surname of the student.
Total assignments	Total number of assignments in the course.
Num assignments done	Number of assignments done by the student.
Average assignments	Average score obtained in the assignments by the student.
Total teststests	Total number of tests in the course.
Num Teststests done	Number of tests done by the student.
Average teststests	Average score obtained in the tests by the student.
Num access	Number of student sessions
Num access <30 s	Number of student sessions of less than 30 s.
Resources visited	Number of resources visited by the student.
Access resources	Number of access to resources by the student.

attributes). The paths from root to leaf represent classification rules. MDM tool provides the decision tree in text mode together with information about classification accuracy (number of correctly and incorrectly classified instances). However, the same decision tree can also be visualized in a graph mode by pressing the "Tree View" link.

Association

The discovery of patterns of interest [30–33] within a specific domain plays a truly important role in the knowledge discovery process. A pattern is considered to define any type of homogeneity and regularity in data, and it represents intrinsic and important properties of datasets. Nevertheless, the knowledge extracted by a single pattern might be meaningless, and a more descriptive analysis may be required. In this sense, the concept of association rules was proposed [28], as a way of describing correlations among sets of items within a pattern of potential interest. The original problem of association rule mining was the market basket analysis [34]. The aim is to find all the interesting relations between the bought products. Such rules associate one or more attributes of a dataset with another attribute, producing IF-THEN statements concerning attribute-values. An association rule is a rule of the form IF X THEN Y, where X and Y are non-intersecting sets of items.

MDM Tool includes the Apriori algorithm [28] for mining association rules. The main reasons for selecting this algorithm are: (1) it is the most well-known algorithm in the pattern mining field; (2) it has served as baseline of any existing association rule mining algorithm in literature; (3) it is easy to implement; (4) it requires a low number of parameter.

In order to run it on Moodle data, instructors only need to select a binary data file and click on the Relationship/Association Rules option as Data Mining Task. Then, they need to select which attributes are appropriate to be used in the mining process. Finally, the instructor is required to choose the minimum values for two quality measures: support and confidence. The support quality measure is defined as the frequency of occurrence of a specific rule, whereas the confidence of a rule measures the reliability of the rule, that is the probability of satisfying the consequent knowing that the antecedent is already satisfied. The final step to run the algorithm once all the aforementioned parameters are properly selected is to click on the Run button (see Fig. 6).

Once the Apriori algorithm has been run, the resulting set of frequent item sets and association rules that satisfy the minimum support and confidence thresholds is obtained. It should be noted that this resulting set can be saved as a PDF or plain text file. First, frequent item sets of different sizes are shown together with their support (number of students that satisfy the item-set). Then, any association rules that can be produced from the previous frequent item sets are obtained. It is worth noting that all these association rules should satisfy the confidence threshold defined by the user.

```
ALGORITHM KMEANS
(9,6,70,18,71,31)[0]
(7,0,20,1,29,19)[0]
(9,5,72,16,65,23)[1]
(10,5,71,19,95,36)[1]
(0,0,1,1,0,1)[0]
(7,3,53,5,22,14)[0]
(10,6,33,5,27,21)[1]
(10,8,81,11,104,41)[1]
...
(10,5,90,20,35,15)[1]
(10,7,71,7,76,26)[1]
(10,8,104,5,76,41)[1]

CENTROIDS
(7,2.8,35.9,3.8,29.4,16.8)[0]
(9.8,6,82.9,8.8,77.6,31.6)[1]
```

Figure 7 Result of the clustering algorithm.

```
DECISION TREE
Resources_Visited=Low
-------Average_Quizzes=High
--------------Num_Assignments_Done=High
---------------------Num_Access=Low
---------------------------=> (Final_Mark=PASS)
---------------------Num_Access=High
---------------------------=> (Final_Mark=FAIL)
--------------Num_Assignments_Done=Low
---------------------=> (Final_Mark=FAIL)
-------Average_Quizzes=Low
--------------=> (Final_Mark=FAIL)
Resources_Visited=High
-------=> (Final_Mark=PASS)
Number of instances to test: 10
Number of instances classified correctly: 7
```

Figure 8 Result of the classification algorithm.

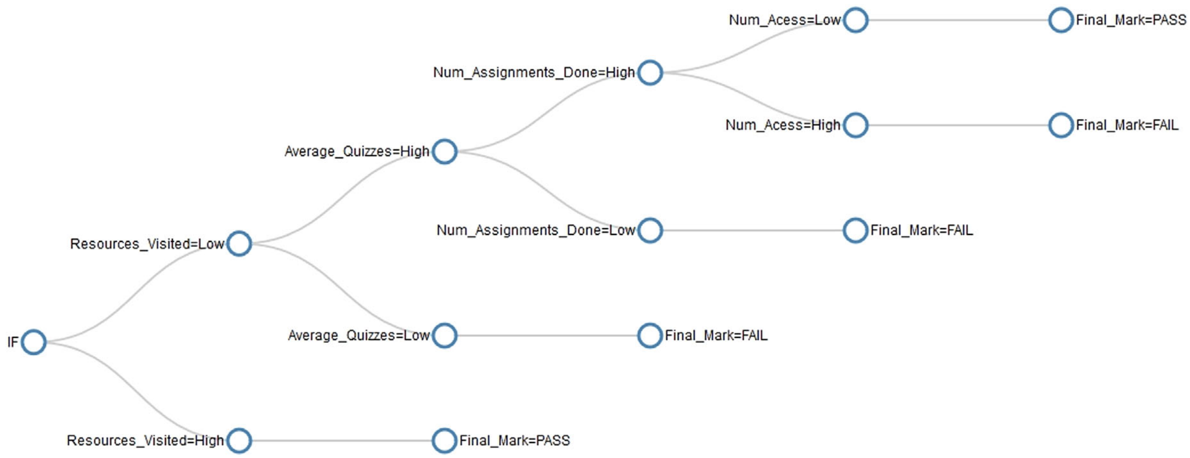


Figure 9 Decision tree in graph mode.

CASE STUDY

For the purpose of better understanding the functionality of the proposed MDM tool, a case study is properly described in this section. The data used was gathered from a real course (all datasets used in the case study are freely available to be donwloaded) on Introduction to Programming (IP) with 1st year undergraduates reading Mechanical Engineering at University of Cordoba (Spain) in 2014. Data comprise information about 31 students and their mark obtained in the final exam to complete the course. The Moodle course consists of 10 subjects ("Temas" in Spanish) and one final test/exam at the end of the course. Each subject includes several resources (PowerPoint presentations and some additional complementary resources) and one assignment that is not used to obtain the final mark.

In our case study, we have considered the statistics Summary [35] that displays a general overview of any activity carried out by the selected students on the Moodle course (see Table 2).

Next, there is a description about how this summary dataset has been pre-processed and how different DM techniques (clustering, classification, and association rule mining) have been applied on data.

Preprocessing

In our case study, we have applied four pre-processing tasks following a specific order: manual data edition/modification, data anonymization, data discretization, and data partition.

Table 3 Rules Extracted From the Decision Tree

#	Rule
1	If resources_visited = low and average_tests = high and num_assignments_done = high and num_access = low THEN final_mark = PASS
2	If resources_visited = low and average_tests = high and num_assignments_done = high and num_access = high THEN final_mark = FAIL
3	If resources_visited = low and average_tests = high and num_assignments_done = low THEN final_mark = FAIL
4	If resources_visited = low and average_tests = low THEN final_mark = FAIL
5	If resources_visited = high THEN final_mark = PASS

Firstly, the following two manual modifications were performed by directly editing the data in the summary dataset:

- To replace percentages by specific numbers. For instance, *Average_Assignments* and *Average_Tests* are shown/stored in MOODLE as the percentage obtained by the division (score/maximum score). Thus, 5/10 indicates that 5 is the average score obtained on a scale whose maximum value is 10. Then, these values are transformed to single numerical values by deleting the backslash (/) and the maximum score (the previous example results in the number 5).
- To add the final mark obtained by each student. The Instructor includes the final mark obtained by students in the course by added a column in the summary table. In this regard, two different labels were considered: *PASS* (students who passed the course) or *FAIL* (students who failed the course).

Next, we have anonymized the data, that is, the name and surname of each student is replaced by a numerical ID (Identificator). Beginning with the previously anonymized summary dataset (*summary_anonymized.xls*), two different discretizations have been considered:

- On the one hand, a dataset (*classification_data.xls*) was created to be used by a classification algorithm. The following attributes were selected: manual method, two bins (labels: *LOW* and *HIGH*), and the average or mean value of each attribute as cut-off: *Num_Assignments_Done* (mean = 8), *Average_Tests* (mean = 4), *Num_Access* (mean = 65), *Num_Access_ (mean = 7), Access_Resources (means = 62), Resources_Visited (means = 26). Thus, if the numerical value is lower than the average, then the label LOW is assigned. Otherwise, label HIGH is considered.*
- On the other hand, two binary datasets (*relationship1_data.xls* and *relationship2_data.xls*) were created to be used by an association rule mining algorithm. Discretization is carried out in a similar way to the previously described.

Finally, the *classification_data.xls* file is split to obtain train and test datasets. In this regard, percentage ratio of 33% is considered to botain the test datasets, whereas a 66% is used to obtain the train dataset. The two resulting datasets are named as *training_data.xls* and *test_data.xls*.

```
Algorithm Apriori

***Frequent Itemsets***

Access_Resources=Low , sup: 0.52
Final Mark=FAIL , sup: 0.42
Num_Access=Low , sup: 0.48
Num_Access_<31sec=Low , sup: 0.55
-
Access_Resources=Low Final Mark=FAIL , sup: 0.42
Num_Access=Low Num_Access_<31sec=Low , sup: 0.42
-
***Rules****

[ Access_Resources=Low -> Final Mark=FAIL ] (sup= 0.4 , conf= 1.00)
[ Final Mark=FAIL -> Access_Resources=Low ] (sup= 0.4 , conf= 1.00)
[ Num_Access=Low -> Num_Access_<31sec=Low ] (sup= 0.4 , conf= 1.00)
[ Num_Access_<31sec=Low -> Num_Access=Low ] (sup= 0.4 , conf= 1.00)
```

Figure 10 Result of the association algorithm mining algorithm.

Clustering

In this section, we include a description about how to use clustering techniques by the proposed MDM tool. The aim of this technique will be to obtain two different sets/groups of students by using information about student's interaction (*Access_Resources, Average_Quizzes, Num_Access, Num_Access <30 s,* Num_Assignments_Done and *Resources_Visited*). Figure 7 shows the result obtained when using the summary_anonymized.xls file with two clusters. Results show the instances (students) assigned to each cluster/group and the centroids for each cluster/group.

As it is illustrated in Figure 7, data are grouped into two different groups (cluster 0 and cluster 1), and each row/instance represents a different student. Results also show the values of each attribute and each student (depicted in brackets and separated by commas). The cluster assigned to each student also appears in square brackets. Finally, the information about the centroids of each cluster is also illustrated. It should be noted that the values of the centroids are the mean/average of each attribute for all students belonging to each cluster. In the example studied in this section, centroids number 1 represents a group of students that are more active and successful than those belonging to centroid number 0. Indeed, on average, they have accessed to more than 9 resources (7 the other group); they have obtained an average score of 6 (3 the other group), they have logged on to the course more than 82 times (35 the other group), they have finished more than 77 assignments (29 the other group), and they have visited more than 31 resources (16 the other group).

Classification

As an example of classification, we are going to classify or predict students who will pass/fail the course according to the *Final_Mark* and using the information about student's interaction (*Access_-Resources, Average_Quizzes, Num_Access, Num_Access <30 s,* Num_Assignments_Done and *Resources_Visited*). Figure 8 shows the final decision tree obtained in text mode when using the *training_data.xls* and *test_data.xls* files. The class to be predicted is defined as the *Final_Mark* attribute.

As it is illustrated in Figure 8, the resulting decision tree has four internal nodes (attributes *Resources_Visited*, *Average_Tests, Num_Assignments_Done* and *Num_Access*) and five leaf nodes (class *Final_Mark*). It obtains a classification accuracy of 70%, i.e., 7 out of 10 students in the test file were correctly classified. Finally, the MDM tool provides an option to show the decision tree in graph mode, which is easier to be understood and more user-friendly. Results in graph mode are illustrated in Figure 9.

Finally, taking the resulting decision tree, we obtain that five *IF-THEN* classification rules can be extracted by following their branches from the root to each leaf node (see Table 3).

Table 3 shows that students who are predicted to *PASS* the course visited a high number of resources (rule no.5); or they visited a low number of resources but, however, they obtained a high average score in tests, they carried out a large number of assignments and they also had a low number of accesses (rule no.1). On the other hand, students who are predicted to *FAIL* the course visited a low number of resources and they obtained a low average score in tests (rule no.4); or they visited a low number of resources, obtained a high average score in tests, finished a low number of assignments (rule no.3); or they visited a low number of resources but they obtained a high average score in tests, finished a large number of assignments, and had a high number of accesses (rule no.2).

Association

As an example of association rule mining, we plan to discover any existing relationship between different actions or attributes about student's interaction with course (*Access_Resources, Average_Quizzes, Num_Access, Num_Access <30 s,* Num_Assignments_Done and *Resources_Visited*). Figure 10 shows the set of item-sets and association rules discovered when using the *relationship1_data.xls* file, and considering 0.40 and 0.80 as the minimum support and confidence threshold values, respectively.

Analysis of Figure 10 reveals a set of frequent single items such as *Accesss_Resources = Low, Final_Mark = FAIL, Num_Access = Low*, etc., together with their support value, which is the percentage of students who matched it. From the resulting set of frequent item-sets, the Apriori algorithm included in the MDM tool extracts a set of association rules, denoting their support and confidence. It should be noted that from an item-set comprising two single items, two association rules can be obtained and their support values are the same since they describe the same item-set. For instance, the results illustrated in Figure 9 describe two association rules obtained from the item-set *Access_Resources = Low Final_Mark = FAIL*. The first rule determines that those students that meet Access_resources = Low will also satisfy

```
***Frequent Itemsets***

Access_Resources=High , sup: 0.48
Average_Quizzes=High , sup: 0.68
Final Mark=PASS , sup: 0.58
Num_Access=High , sup: 0.52
Num_Access_<30sec=High , sup: 0.45
Num_Assignments_Done=High , sup: 0.77
-
...

***Rules****

[ Access_Resources=High -> Average_Quizzes=High ] (sup= 0.6 , conf= 1.00)
[ Average_Quizzes=High -> Access_Resources=High ] (sup= 0.6 , conf= 1.00)
[ Access_Resources=High -> Final Mark=PASS ] (sup= 0.6 , conf= 1.00)
[ Final Mark=PASS -> Access_Resources=High ] (sup= 0.6 , conf= 1.00)
[ Average_Quizzes=High -> Final Mark=PASS ] (sup= 0.6 , conf= 1.00)
[ Final Mark=PASS -> Average_Quizzes=High ] (sup= 0.6 , conf= 1.00)
[ Average_Quizzes=High -> Num_Access=High ] (sup= 0.6 , conf= 1.00)
[ Num_Access=High -> Average_Quizzes=High ] (sup= 0.6 , conf= 1.00)
[ Final Mark=PASS -> Num_Access=High ] (sup= 0.6 , conf= 1.00)
[ Num_Access=High -> Final Mark=PASS ] (sup= 0.6 , conf= 1.00)
...
```

Figure 11 Results obtained by the association rule mining algorithm.

Final_Mark = *FAIL*, and this rule has a maximum reliability, that is a confidence value equal to 1.

Let us now consider the results obtained by Apriori for a different data file: relationship2_data.xls and with the same thresholds for support and confidence, that is 0.40 and 0.80, respectively. Figure 11 illustrates the set of association rules discovered by the algorithm. By analyzing the resulting set of rules, truly interesting association rules can be found. For instance, the rule *IF Access_Resources* = *High THEN Average_Quizzes* = *High*. This rule is met by 60% of the students, denoting that this percentage of students are very active in terms of their use of Moodle. Additionally, the confidence of this association rule states that any student that has a high number of accesses to the Moodle platform uses it to access quizzes obtaining a high average. Thus, it suggests that, it is hard to find a student that accesses Moodle simply to have a look, without using resources to obtain good average in quizzes. Another interesting rule is *IF Num_Acess* = *High THEN Final_Mark* = *PASS*, since it indicates that those that pass the course are highly active students, that is students with a high interest in the course. According to the support quality measure, it defines that 60% of the students in this dataset will have a high number of accesses to Moodle and will pass the course.

CONCLUSIONS AND FUTURE WORK

Moodle management systems include a growing number of Learning Analytics tools to provide some type of basic analytics and graphs about users' interaction. However, it is hard to find any integrated tool that enables a varied set of Data Mining techniques to be applied in the course. In this way, the aim of this paper is to propose a new data mining tool, called MDM, for the discovery of knowledge in Moodle data. The major feature of the proposed tool is that all its algorithms were implemented in PHP and integrated in Moodle platform. The proposed data mining tool may be considered as a complement for all the existing Learning Analytics plug-ins, providing new, and varied functionalities related to the knowledge discovery process: pre-processing and data mining techniques, including supervised and unsupervised learning tasks.

In order to demonstrate the usefulness of the proposed tool, we have described its general features, illustrating its architecture, and interface. Finally, the utility of the proposed tool is described with a case study using a sample real-life dataset of engineering students. This case study illustrates how the tool enables input data to be transformed as the user requires. Three different data mining techniques have been applied in this case study. The models and results obtained were analyzed and described in order to show its usefulness for providing instructors with feedback about how students learn within Moodle courses.

In future work, we intend to add new data mining algorithms in order to provide instructors with more advanced algorithms. Currently, we are working on an updated version of the Apriori algorithm that works with discretized data and another classifier that works both with nominal and discretized data. We are also working in developing a visual interface to present the results obtained in clustering and association rule mining tasks. Additionally, we would also like to add parameter-free data mining algorithms to simplify their configuration and running.

ACKNOWLEDGMENTS

This research is supported by the Spanish Ministry of Science and Technology under project no.TIN2014-55252-P.

REFERENCES

[1] J. Cole and H. Foster, Using moodle, 2nd ed, O'Really Community Press, Sebastopol, CA, 2005.

[2] W. H. Rice, Moodle E-learning course development. A complete guide to succeful learning using moodle, 3rd ed, Packt Publishing, Birmingham, UK, 2015.

[3] M. A. Conde, A. Hernádez-García, F. J. García-Peñalvo, and M. L. Séin-Echaluce, Exploring student interactions: Learning analytics tools for student tracking, International Conference on Learning and Collaboration Technologies (2015), 50–61.

[4] C. Romero and S. Ventura, (2010). Educational Data Mining: A Review of the State-of-the-Art. IEEE Transactions on Systems, Man, and Cybernetics–Part C: Applications and Reviews. 40:601–618.

[5] G. Siemens, and P. Long, (2011). Preface of the 1st International Conference on Learning Analytics and Knowledge. Banff, Alberta Canada.

[6] G. Siemens and R. S. J. d. Baker, Learning Analytics and Educational Data Mining: Towards Communication and Collaboration. Proceedings of the 2nd International Conference on Learning Analytics and Knowledge. (2012) 1–3.

[7] C. Romero and S. Ventura, (2017) Educational data science in massive open online courses. WIREs Data Mining Knowledge Discovery, in press.

[8] B. Dragulescu, M. Bucos, and R. Vasiu, (2015). CVLA: Integrating Multiple Analytics Techniques in a Custom Moodle Report. International Conference ICIST, Druskininkai, Lithuania, 115–126.

[9] D.Y.-T. Liu, J. Froissard, D. Richards, and A. Atif, An enhanced learning analytics plugin for moodle: Student engagement and personalised intervention. In: Australasian society for computers in learning and tertiary education (ascilite2015), 29 November - 2 December 2015, Perth, Australia.

[10] D. García-Saiz and M. E. Zorrilla, A service oriented architecture to provide data mining services for non-expert data miners, Decis Supp Syst 55 (2013), 399–411.

[11] R. Mazza, M. Bettoni, M. Faré, and L. Mazzola, (2012) MOCLog—Monitoring Online Courses with log data. 1st Moodle Research Conference Proceedings, Heraklion, Greece, 132–139.

[12] S. Graf, C. Ives, N. Rahman, and A. Ferri, AAT-A tool for accessing and analysing student's behaviour data in learning systems. In: Proceedings of the 1st International Conference on Learning Analytics and Knowledge, Lak, Banff, AB, Canada, 2011, pp 174–179.

[13] S. Slater, S. Joksimovic, V. Kovanovic, R. Baker, and D. Gasevic, Tools for educational data mining, J Edu Behav Stat (2016); DOI: 10.3102/1076998616666808

[14] S. Retalis, A. Papasalouros, Y. Psaromiligkos, S. Siscos, and T. Kargidis, Towards networked learning Analytics—A concept and a tool. Fifth International conference on networked learning. Lancaster University, Lancaster, UK, 2006, pp 1–8.

[15] D. Gómez-Aguilar, R. Theron, and F. Garcia-Peñalvo, Understanding educational relationships in Moodle with ViMoodle, International Conference on Advance Learning Technologies, (2008), 954–956.

[16] E. García, C. Romero, S. Ventura, and C. de Castro, An architecture for making recommendations to courseware authors through association rule mining and collaborative filtering, UMUAI 19 (2009), 99–132.

[17] R. Rabbany, M. Takaffoli, and O. Zaïane, Analyzing participation of students in online courses using social network analysis techniques. In international conference on educational data mining, Eindhoven (2011), 21–30.

[18] R. Pedraza-Perez, C. Romero, and S. Ventura, A java desktop tool for mining moodle data. In international conference on educational data mining, Eindhoven (2011), 319–320.

[19] A. Zafra, C. Romero, and S. Ventura, DRAL: A tool for discovering relevant e-Activities for learners, Knowledge and Information Systems 36 (2013), 211–250.

[20] M. Luca, N. Mauro, M. Christian, and M. Riccardo, (2010). L'aggiornamento del tool di monitoraggio delle attività degli studenti: GISMO 2.0. In: Atti del convegno italiano MoodleMoot, Bari, Italia. 1–3.

[21] A Bakharia and S. Dawson, SNAPP: A bird's-eye view of temporal participant interaction. International conference on learning analytics and knowledge. ACM New York, NY, USA, 2011, pp 168–173.

[22] IntelliBoard, (2015), IntelliBoard.net Retrieved from http://www.intelliboard.net/

[23] Klassdata, SmartKlass: The Learning Analytics Plugin (2015), Retrieved from http://klassdata.com/smartklass-learning-analytics-plugin/

[24] J. Singh, New Block: Analytics Graphs (2015), Retrieved from http://www.Moodleworld.com/2015/03/31/new-block-analytics-graphs

[25] J. Cruz-Benito and F. Garcia, VeLA: A visual eLearning analytics tool, LASI, Bilbao (2015), 1–18.

[26] J. Han and M. Kamber, (2006) Data mining: Concepts and techniques, Morgan Kaufmann Publishers.

[27] J. MacQueen, Some methods for classification and analysis of multivariate observations. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. Vol. 1, University of California Press, Berkeley, CA, 1967, pp 281–297.

[28] C. Aggarwal and C. Reddy, Data clustering: Algorithms and applications, CRC Press, Boca Raton, FL, 2014.

[29] J. R. Quinlan, Induction of decision trees, Machine Learning 1 (1986), 81–106.

[30] J. M. Luna, Pattern mining: Current status and emerging topics. Progress in artificial intelligence, First Online (2016), 1–6. DOI: 10.1007/s13748-016- 0090-4

[31] S. Ventura and J. M. Luna, Pattern Mining with Evolutionary Algorithms, 1st edition, Springer International Publishing, Cham (ZG), Switzerland, 2016.

[32] R. Agrawal, T. Imielinski, and A. Swami, Mining association rules between sets of items in large databases. Proceedings of the ACM SIGMOD international conference on management of data. ACM New York, NY, 1993, pp 207–216.

[33] C. Romero, J. M. Luna, J. R. Romero, and S. Ventura, RM-Tool: A framework for discovering and evaluating association rules, Advances in Engineering Software 42 (2011), 566–576.

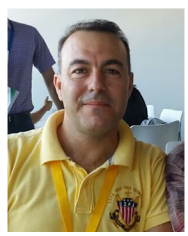
[34] R. C. Blattberg, B.-D. Kim, and S. A. Neslin, Maket basket analysis. Database marketing analyzing and managing customers. Springer-Verlag, New York, NY, 2008.

[35] C. Romero, J. R. Romero, and S. Ventura, A survey on pre-processing educational data. Educational data mining: Applications and trends. Springer International Publishing, Cham (ZG), Switzerland, 2013, pp 29–64.

BIOGRAPHIES

Jose Maria Luna received PhD degree in Computer Science from the University of Granada (Spain) in 2014. His research was firstly granted by the Ministry of Education with FPU AP2010-0041, and then, he was awarded with a JdC - training PostDoc. He is author of the book "Pattern Mining with Evolutionary Algorithms", published by Springer. He has published more than 30 papers in top ranked journals and international scientific conferences, and he is author of two book chapters. He has also been engaged in four national and regional research projects. Dr. Luna has contributed to three international projects. His research is performed as a member of the Knowledge Discovery and Intelligent Systems Laboratory and is focused on evolutionary computation, pattern mining, association rule mining and its applications.

Cristobal Castro received MSc degree in computer science from the University of Cordoba (Spain) in 2015. His research is performed as a member of the Knowledge Discovery and Intelligent Systems Laboratory.



Cristóbal Romero (http://www.uco.es/~in1romoc/) is an associate professor in the Department of Computer Science and Numerical Analysis at the University of Córdoba in Spain and member of the KDIS (Knowledge Discovery and Intelligent Systems) research group. He received his BSc and PhD degrees in computer science from the University of Granada, Spain, in 1996 and 2003, respectively. He has published more than 100 papers in books, journals, and conferences, 35 of which have been published in Thomson-Reuters Impact Factor (IF) journals and some of them are important EDM (Educational Data Mining) surveys and reviews. He is also the co-editor of several special issues and two books regarding EDM (Data Mining in e-learning and Handbook of Educational Data Mining.).