

Symposium on Data Mining Applications, SDMA2016, 30 March 2016, Riyadh, Saudi Arabia

Predicting Students' Performance in University Courses: A Case Study and Tool in KSU Mathematics Department

Ghada Badr^{a,b,*}, Afnan Algobail^a, Hanadi Almutairi^a, Manal Almutery^a

^aKing Saud University, College of Computer and Information Sciences, Computer Science Department, Riyadh, Kingdom of Saudi Arabia

^bIRI-The City of Scientific Research and Technological Applications, University and Research District, P.O. 21934 New Borg Alarab, Alex, Egypt

Abstract

Educational data mining is a growing field that uses the data obtained from educational information systems to discover knowledge and find answers to questions and problems concerning the education system. High dropout rates and poor academic performance among students are examples of the most common issues that affect the reputation of an educational institution. Students' academic records can be analyzed to explore the factors behind these phenomena. This paper discusses the building of a model to predict the performance of students in a programming course based on their grades in courses in other subjects. A classification based on an association rules algorithm is used to build a classifier to help evaluate the student's performance in the programming course. This model aims to reduce dropout levels by helping student predict their likelihood of success in a course before they enroll in it. In addition, course instructors will be able to enhance student performance in the course by better estimating their abilities to learn the subject matter and adjusting their teaching strategies and methods.

© 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the Organizing Committee of SDMA2016

Keywords: Data mining; classification; prediction; higher education; association rules; CBA algorithm.

1. Introduction

Every day, we generate a huge amount of data from different sources such as social networks, business transactions, and clinical records. These data are stored in databases as row data, and we do not benefit from the potentially useful information that we could extract from them. However, various data mining (DM) techniques and

* Corresponding author. Tel.:011-80-51941; fax: +0-000-000-0000 .

E-mail address: ghbadr@ksu.edu.sa

tools have been developed to turn this growing volume of data into valuable information. DM, or Knowledge Discovery in Database (KDD), is defined as “the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner [1].” In other words, DM can be useful and effective in extracting important, relevant information that has not previously been discovered. DM methods and techniques provide significant potential to organizations and researchers to discover implicit information in various areas, including bioinformatics, genetics, and education.

In the past few years, DM has been successfully deployed to enhance the quality of learning and teaching in higher educational institutions. In addition, DM techniques and tools have been found capable of explaining the causes of longstanding issues faced by the higher education sector, including rates of course failure and dropout and poor academic performance. To improve these rates, various data mining methodologies, including classification, clustering, regression, and association rules, can be used to predict students’ future grades. In fact, predicting students’ future performance based on their past academic performance and that of previous students is a common data mining task. The ultimate aim of such predictions is to help students find how well they will do in a particular course before they register in it so as to avoid having to drop out. In addition, it will help the course instructor identifying students at risk and take appropriate actions and adopt new strategies to improve student success.

This paper presents a data mining model for predicting student performance in a programming course based on their performance in English and mathematics courses. The Classification Based on Association rules (CBA) algorithm was used to build a classification model for predicting students’ performance. As its subject, the study used the academic records of mathematics students who graduated from King Saud University (KSU) between 2008 and 2014.

The rest of this paper is organized as follows: Section 2 contains background information on DM in higher education and classification association rules for DM. Section 3 introduces some related research. Section 4 proposes overall work that can be used for educational DM research. Section 5 presents the main idea behind the project, a case study proposing an application to predict students’ performance in a programming course based on their performance in mathematics and English courses. Section 6 summarizes the results. Section 7 presents discussion and conclusions.

2. Background

2.1. Data mining in higher education

Data mining can be applied in various fields to enhance the overall performance of a system. This can be done by extracting from a stored dataset important, relevant information that has not previously been discovered. The knowledge thus obtained can contribute to resolving many issues and improving the current system.

There is increasing interest in using DM in the educational field. In fact, applying traditional DM techniques to educational data such as student academic records is referred to as Educational Data Mining (EDM). EDM is defined as “the process used for transforming raw data compiled by education systems into useful information that could be used by lecturers to take corrective actions and answer research questions [2].” In other words, EDM can help institutions examine and improve the student learning process.

Understanding the student learning process plays an important role in developing an institution’s educational process. Such understanding offers several advantages, including improving the outcomes of student learning and enabling planning to assist weaker students. Consequently, the number of students failing or dropping out of courses will decrease.

2.2. Classification Association Rules Mining (CARM)

There are many techniques for mining large amounts of data to explore and extract knowledge from it. The two most significant DM methods are classification rule mining and association rule mining. The former generates an accurate classifier by selecting a small group of rules from the dataset. The latter aims to discover and identify all rules within some minimum support and confidence constraints.

Liu *et al.* [3] proposed a new framework based on both association and classification rule mining. This method created a classifier using the generated classification association rules (CARs). A CAR is composed of two main steps: producing association rules and constructing the classifier. The results showed that CARM provides more accurate results than C4.5 algorithm [4].

3. Related work

V. Kumar and A. Chadha (2012) [4] applied the CARM techniques using Tanagra tool in order to enhance students' performances. Dataset was provided by renowned university in Haryana form Master of Computer Applications (MCA). They used the Apriori algorithm to compare students' performances in common courses at the undergraduate and post-graduate levels, discovered associations, and then identified factors that determined students' chances of success or failure like syllabus plan, student's interest, teaching and evaluation techniques.

B. Baradwaj and S. Pa (2011) [5] performed a study using Bayesian classification on bachelor of computer application students from Awadh University in India. Generally, they applied the method through choosing three hundred students from various levels, the final data set consisted of seventeen data objects. The authors found that the student's performance was strongly correlated with other elements other than students' effort such as family income, students' routine, and different factors were mentioned in the paper.

J. Kasih, M. Ayub, and S. Susanto (2013) [6] proposed a model to predict students final grade in the programming course. The prediction value was classified into three categories: Extraordinary, very satisfactory, and satisfactory. The authors demonstrated a solution based on the Apriori algorithm to discover the relationship between a programming course and other subjects students took during the first four-semester of their study period. As a result, they found a high correlation between computer courses and math courses.

Z. Abdullah, T. Herawan, N. Ahmad and M. Deris (2011) [7] presented a model to determine a suitable program for students based on their interest in the program rather than on availability. The proposed solution is based on using SLP-Tree and lift measurement to discover highly correlated association rules. The study, which was conducted using Computer Science students from Malaysia University, found many students were offered computer science programs but were not within their program's field interests.

4. Proposed workflow

This section discusses the flow of work that can be involved in EDM studies. Important steps in the process include data collection, preprocessing, integration of classification, and association rule mining. Fig. 1. shows a flowchart of the proposed workflow.

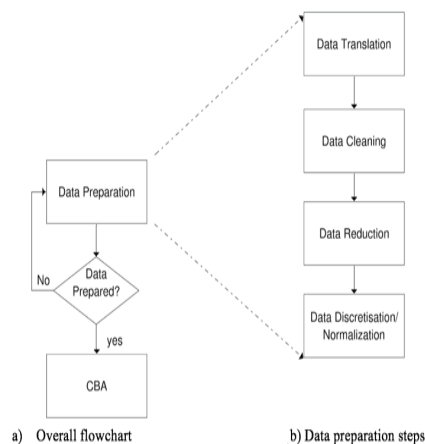


Fig. 1. (a) the flowchart shows the proposed workflow: First is the data preparation stage, then the Classification Based on Associations (CBA) stage, which generates all class association rules (CARs) to construct the classifier.; (b) the flow chart shows the data preparation stages, including data translation, data cleaning, data reduction, and data discretization/normalization.

4.1. Data preparation

This step is very important for any DM process, since it specifies the data to be mined, including collecting data from the source, preparing it, and preprocessing it so that it is in the format necessary for data mining analysis. This step should be done before applying DM because it significantly affects the accuracy and interpretability of the DM results. In the case of most educational data, preprocessing should include the following steps.

4.1.1. Data translation

Institutions usually store students' records using their original language. Therefore, records may need to be translated, either manually or by machine translation, to the language of the study, which was English in this case. This step improves understandability of the data and simplifies the process of using and integrating it.

4.1.2. Data cleaning

In the real world, data may contain inconsistent or incomplete values. For example, integrating educational data may introduce inconsistent attributes because some institutions store grades as letters and others store them as numeric values. This presents a data quality problem that can obscure useful patterns. Therefore, to obtain accurate DM results, it is necessary to clean the data, a process that may involve replacing missing data, removing noisy values, and resolving inconsistencies. There are many strategies for handling missing values; one approach is to ignore the row when the class attribute is missing. Another is to fill in the missing data manually with the attribute mean or with a global constant value such as NULL.

4.1.3. Data reduction

Sometimes attributes, such as student name or ID that do not provide knowledge pertinent to the mining process must be eliminated. This can be accomplished by selecting only attributes relevant to DM. The final data should include the courses under study, saved in a file format suitable for the DM task.

4.1.4. Data discretization/normalization

Discretization is typically important to reduce values, which can be categorized as supervised and unsupervised. Supervised class labels should be indicated while the unsupervised class labels are unknown. However, supervised discretization has been shown to significantly influence the classification efficiency of the algorithms used. For this reason, the present study adopted the supervised class approach. In addition, normalization is very useful for classification algorithms because it helps avoid dealing with large value intervals.

4.2. Classification Based on Associations (CBA)

CBA [4] is an algorithm that combines both classification and association rule mining. It is based on generating all class association rules and then using these rules to construct the classifier. To date, there has been no real application of the CBA algorithm to educational data. So, because it uses the CBA algorithm, this study's results contribute significantly to the field of EDM.

5. Case study: Predicting students' grades in a programming course for KSU mathematics department

The objective of this study was to identify the association rules relating a particular course to other courses and then to use those rules to predict the students' grades in the course. The case study investigated the poor performance of students in a programming course, identifying the association rules relating the programming course to mathematics and other courses and then using these rules to predict the students' grades in the programming

course. The study followed the workflow proposed in the previous section by performing the following specific steps:

1. First Gathering and preparing the data.
2. Generating the classifier using the CARM algorithm, which was based on support and confidence values.
3. Using the rules generated to predict the students' performance in a programming course by implementing a small application in the JAVA programming language, with the help of LUCS-KDD CARM Discretization/Normalization (DN) software Version 2 [8] and LUCS-KDD implementation of CBA [9].

5.1. Data preparation

The dataset used for this study was obtained from the Mathematics Department in the College of Sciences. It included the records of mathematics graduate students from 2008–2014. The study received official approval to copy these records from the Dean of Admissions and Registration. The main problem with the dataset was that data were stored in a non-understandable format, were stored in multiple files, and were in Arabic. In addition, the data included many irrelevant courses and multiple unnecessary details such as the graduation year.

5.1.1. Data translation

To transform the data into a more understandable format, the data were translated into English and all the records were combined into a single table. This process was performed using JAVA code developed for this purpose. The resulting data file was an Excel spreadsheet with 203 records and 57 attributes, in which each row represented one student and the columns represented graduation year, student ID, course code, and GPA. Table 1 shows an example of the data set after the translation.

Table 1. Part of the original dataset after translation, with 203 records and 57 attributes.

Year	ID	ENG102	ENG101	ENG104	ENG121
29	1	D+	NULL	Drop	NULL
29	2	NULL	NULL	NULL	B
29	3	NULL	A+	NULL	A
29	4	NULL	NULL	NULL	C
29	5	NULL	A	NULL	A
29	6	NULL	NULL	NULL	B
29	7	NULL	F	NULL	D+
29	8	NULL	C+	NULL	B

5.1.2. Data cleaning

Then, data cleaning was performed on the file to address the values for some attributes that were missing because the courses selected were optional and not all the students registered for them. Three main methods were adopted to deal with these missing values. First, the average score in the course was calculated and used as the missing grade. Second, the missing value was replaced with the student's grade in an equivalent course. Finally, the rows that had missing values were eliminated if the percentage of unavailable data was high.

5.1.3. Data reduction

Correlation can be used as an important preprocessing step for feature selection and reduction. We used data reduction to identify which attributes depended on the target attribute. We used the correlation coefficient [10] to compute the correlation between attributes of type numeric where a_i and b_i represent values of A and B in row i, \bar{A}

and \bar{B} represent the sample mean of attributes A and B, σ_A and σ_B represent standard deviations of A and B respectively, and N represent the number of instance.

$$r_{A,B} = \frac{\sum_{i=1}^N (a_i - \bar{A})(b_i - \bar{B})}{N\sigma_A\sigma_B} \quad (1)$$

When we applied this equation to our data, we found that there was a high positive correlation between some courses (English and Math) and the program course, as shown in Table 2.

Table 2. Correlation coefficient values of sample attributes with the programing attribute.

	CSC206	Eng.121	Eng.122	Math102	Math201	Math202	Math253
	3.50	0.00	0.00	3.00	4.00	3.00	2.00
	5.00	4.00	2.50	2.50	2.00	2.00	3.00
	4.75	4.75	4.75	2.00	3.00	2.50	4.00
	5.00	3.00	3.00	2.50	2.00	2.00	3.50
	3.50	4.75	4.75	2.00	3.00	2.00	2.00
	4.75	4.75	4.00	3.00	2.00	2.00	4.00
Correl. value	1.00	0.42	0.50	-0.19	0.04	-0.17	0.00

We reduced the number of attributes by selecting courses that had high correlation with the programming course CSC206. For English courses, we selected ENG121 and ENG122 for two main reasons. First, most students enrolled in these courses, meaning that we had fewer NULL values. Second, they had a high correlation rate compared to other English courses. We took the student's grade from another English course or the average if a grade for one of these courses was not available. In contrast, all math courses had the same correlation value, so we selected MATH201, which was taught in Arabic, and MATH253, which was taught in English and had a high correlation rate.

5.1.4. Data discretization/normalization

In this step, we determined the class label and attribute. In this case, the class attribute was the programming course for all students who took any programming course, and since the data set we got had a small number of records, we used two categories, binary class labels, which were good/bad: good for A+, A, B+, and B values and bad for C+, C, D, and D+. Table 3 shows an example of the dataset used.

Table 3. This table shows part of the dataset used, which consisted of two courses of English and math, respectively, with a specified class attribute (programming course).

ENG121	ENG122	MATH201	MATH253	Programming
D+	D+	B	D	Bad
B	D+	D	C	Good
A	A	C	B	Good
B	B	D+	D+	Bad
D+	D+	D	D	Good
B	C+	D+	C	Bad

The dataset needed to be discretized and normalized into a binary format suitable for LUCS-KDD software [9]. To accomplish this, the user should input the schema of the dataset to be converted. This file has three lines; the first represents the type of each field, the second gives the names of the fields, and the third contains possible legal values separated by white space.

When the application is run, the *Preprocessing* tab is shown. This tab includes instructions for accomplishing preprocessing. First, the user loads the schema file and then the dataset file, either space or comma separated. Lastly, the user sets the division value. Fig. 2. shows a simple data file after preprocessing.

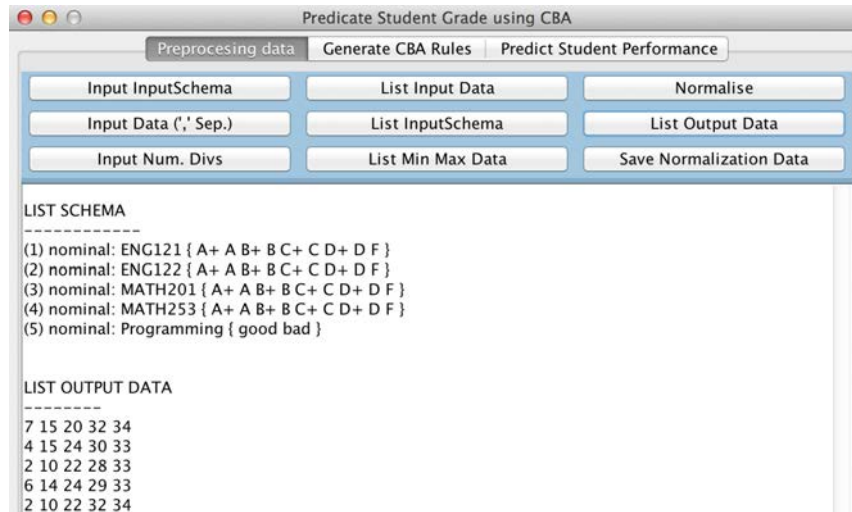


Fig. 2. Data discretized and normalized using the implemented application with the help of LUCS-KDD CARM Discretization/Normalization (DN) software [8]. This step is important for the use of Classification Association Rule Mining (CARM).

5.2. Classification Association Rule Mining (CARM)

To build a classifier, the system uses the file generated by the data discretization/normalization step as an input file. Then, the system applies multiple rules to the dataset to produce the CBA rules, which are based on three parameters: number of classes, support threshold, and confidence threshold.

The process of generating the rules using the application is as follows. First, the user clicks the *Generate CBA Rules* tab. Then, the user fills three input boxes with the number of classes, support, and confidence threshold. Then, the user clicks the *Generate Rules* button to display the results, which are shown in the *Results* field and include the time taken to generate the rules, their percentage of accuracy, and the classification rules.

The difficult part of this process is identifying the best values for support and confidence. This is very important because the values for support and confidence directly affect the values for the generated rules and accuracy rate. Various numbers were tested to find the most appropriate values for these parameters.

Our case study conducted two experiments with different datasets. The first used the student results in two English courses and two mathematics courses. The second one used the student results only in two English courses. This was done to confirm that the English courses alone have a direct effect on the student results in the programming course. Mathematics courses were found to have no effect on student performance in the programming course.

5.2.1. Classification generating rules using English and Math courses

Table 4 shows a set of different support and confidence values gave various accuracy rate. For (support = 6, confidence = 72) produced a higher accuracy rate of 62.75% compared to other proposed values.

Table 4. Values for support, confidence, and accuracy rate.

Support	Confidence	Accuracy
1	72	60.75%
2	72	56.79%
3	72	53.04%
6	72	62.75%
7	72	60.25%

The rules generated using the CBA algorithm are shown as below:

- (1) {ENG122 = A, ENG121 = A+} -> {Programming = Good} 91.66
- (2) {ENG121 = A+} -> {Programming = Good} 91.3
- (3) {ENG122 = A} -> {Programming = Good} 83.87
- (4) {ENG122 = D} -> {Programming = Bad} 78.57

These rules can be interpreted as follows:

1. If the student got an A in ENG122 and an A+ in ENG121, then she will get a good grade in the programming course, with confidence of 91.66%.
2. If the student got an A+ in ENG121, then she will get a good grade in programming course, with confidence of 91.3%.
3. If the student got an A in ENG122, then she will get a good grade in the programming course, with confidence of 83.87%.
4. If the student got a D in ENG122, then she will get a bad grade in the programming course, with confidence of 78.57%.

5.2.2. Generating rules using English courses

Table 5 shows a set of different support and confidence values gave various accuracy rate. For (support = 6, confidence = 72) produced a higher accuracy rate of 67.33% compared to other proposed values.

Table 5. Values for support, confidence, and accuracy rate.

Support	Confidence	Accuracy
1	72	61.71%
2	62	64.04%
3	82	52.46%
6	72	67.33%
7	62	61.5%

The rules generated using the CBA algorithm are shown as below:

- (1) {ENG122 = A, ENG121 = A+} -> {Programming = Good} 91.66
- (2) {ENG121 = A+} -> {Programming = Good} 91.3
- (3) {ENG122 = A} -> {Programming = Good} 83.87
- (4) {ENG122 = D} -> {Programming = Bad} 78.57

5.3. Predicting Student Performance

Using the rules generated in the previous section, the application can specify the students' future performance in the programming course, whether good or bad. The prediction process is as follows. First, the user clicks the *Predict Student Performance* tab. Then, the user chooses from a dropdown menu the grades earned in both English and mathematics courses. Then, the user clicks the *Predict* button to display the predicted performance. Fig. 3. shows an example showing that if the student got an A in ENG121, an A+ in ENG122, a C+ in MATH201, and a D+ in MATH253, she will probably get a good grade in the programming course.

Fig. 3. Predicting student performance

6. Results

To extract useful knowledge, this study analyzed the rules generated in Section 5 and recognized the relationship between the programming course and the other courses. The mathematics courses had no effect on the students' performances in the programming course. Thus, only the English courses have a direct effect on the programming course. In addition, the study tested the accuracy of the classifier on the test dataset, as discussed below.

6.1. Testing the classifier using English and Math courses

Table 6 shows sample prediction results for the classifier built using English and mathematics courses. The study was able to predict 9 out of 17 correctly, for a percentage of 52.94%.

Table 6. A sample of English and mathematics class prediction results.

ENG121	ENG122	MATH201	MATH253	Programming	Results
B	C+	D	C	Bad	Bad
C	D+	D+	C+	Bad	Bad
C	C	D	C	Bad	Bad
A	D+	C	D	Bad	Bad
D+	C	D+	B+	Good	Bad
A+	A+	C	B+	Good	Good

6.2. Testing the classifier using English courses

Table 7 shows sample prediction results for the classifier built using English courses. The study was able to predict 9 out of 17 correctly, with a percentage of 52.94%.

Table 7. A sample of English class prediction results.

ENG121	ENG122	MATH253	Programming
B	C+	Bad	Bad
C	D+	Bad	Bad
C	C	Bad	Bad
D+	C	Good	Bad
A+	A+	Good	Good
C+	C+	Good	Bad

7. Conclusion

This study built an application to predict students' performance in a programming course based on their previous performances in specific mathematics and English courses. The main objective was to improve the quality of higher education institutions by enabling them to adopt a classification approach that helps identify those students at risk, thus allowing adjustment of their instructors' teaching strategies. In addition, this application can reduce dropout rates by helping students predict their performance in programming courses before registering for them. Two experiments were conducted using the CBA rule-generation algorithm. The first used students' grades in two English courses and two mathematics courses, which generated four rules with accuracy of 62.75%. The second used students' grades only in two English courses, generating four rules with accuracy of 67.33%. These results showed that students' performance in English courses has a significant predictive effect on their performance in the programming course. In future work, we will try to include more data records to generate more rules with a higher accuracy rate.

References

1. Hand, D., Mannila, H., and Smyth, P. 2001. *Principles of Data Mining*. MIT Press, Cambridge, MA.
2. Funatsu, K. 2011. *New Fundamental Technologies in Data Mining*. InTech, Croatia.
3. Liu, B., Hsu, W., and Ma, Y. 1998. Integrating classification and association rule mining. In *Proceedings of the fourth international conference on knowledge discovery and data mining (KDD-98, Plenary Presentation)*, New York, USA. <http://www.aaai.org/Papers/KDD/1998/KDD98-012.pdf>
4. Kumar V., and Chadha, A. 2012. Mining association rules in student's assessment data. *IJCSI International Journal of Computer Science* 9, 5, 211-216.
5. Bhardwaj, B. and Pal, S. Data Mining: A prediction for performance improvement using classification. *International Journal of Computer Science and Information Security (IJCSIS)* 9, 4 (2011), 136-140.
6. Kasih, J., Ayub, M., and Susanto, S. 2013. Predicting students' final passing results using the Apriori algorithm. *World Transactions on Engineering and Technology Education* 11, 4, 517-520.
7. Abdullah, Z., Herawan, T., Ahmad, N., and Deris, M. 2011. Extracting highly positive association rules from students' enrollment data. *Procedia - Social and Behavioral Sciences* 28, 107-111.
8. Coenen, F. *LUCS-KDD DN Software*. Department of Computer Science, The University of Liverpool, UK, 2003.
9. Coenen, F. *LUCS KDD implementation of CBA (Classification Based on Associations)*. Department of Computer Science, The University of Liverpool, UK, 2004.
10. Han, J. and Kamber, M., 2006. *Data mining*. Amsterdam: Elsevier.