Chapter 13
Association Rule Mining in Higher Education: A Case Study of Computer Science Students

Njoud Alangari and Raad Alturki

13.1 Introduction

The volumes of data that the world is producing every day are extremely huge. Data can be produced by humans directly through our interactions with social networks or by feeding of data into electronic systems. Also, they can be produced solely by machines, such as those logging the activities of humans or machines. Data production and storage have become the norm in every aspect of our lives. They are used in healthcare, banking, education, and even in homes. With adaptation of new technologies such as the internet of things (IoT) and advancements in technologies to produce and store data, it is expected that the world will produce more and more data.

Data can be stored in many forms and in different ways. They can be stored in relational databases and spreadsheets, and are called structured data. Alternatively, data can be stored in a nontraditional row–column database, such as text and multimedia content (e.g., videos, photographs, and audio) and are called unstructured data. Furthermore, data that contain semantic tags (such as e-mail messages, XML, and HTML) and are not stored in relational databases are called semistructured data. Because of this diversity, there have been many methods and techniques used to mine each type. Data differ in the scope and the field that they belong to, and data-mining (DM) experts deal with them differently. For instance, the objective used to mine educational data can be different from the objective used to mine healthcare data. It could be acceptable to build a model that predicts student performance with

N. Alangari · R. Alturki (✉)
Department of Computer Science, Al-Imam Mohammad Ibn Saudi Islamic University,
Riyadh, Saudi Arabia
e-mail: naalanqari@sm.imamu.edu.sa; ralturki@imamu.edu.sa

R. Mehmood et al. (eds.), *Smart Infrastructure and Applications*,
EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-13705-2_13

85% accuracy, whereas it is not acceptable to have accuracy of only 85% to predict the success rate of drug treatment.

DM techniques can be predictive or descriptive. Predictive methods use variables to predict unknown or future values of one or more variables. Examples of predictive methods are classification, regression, and deviation detection [1]. Descriptive methods such as clustering, association rule discovery, and sequential pattern discovery find human-interpretable patterns that describe data [1].

Regression techniques aim to predict the value of a continuous attribute on the basis of the values of other attributes [1]. The simplest form of regression is linear regression, where the class is a linear combination of attributes with predetermined weights. On the other hand, classification methods aim to predict the value of a discrete attribute on the basis of the values of other attributes. Classification has many approaches, including decision tree (DT), Bayesian classifier or network, neural network (NN), support vector machine (SVM), and logistic regression approaches. Logistic regression allows us to use regression for classification.

Clustering measures the similarity between data points or variables; in other words, it groups similar data points on the basis of their attributes, and each group is called a cluster [1]. Clustering methods are divided, in general, into two groups: partitioning methods and hierarchical methods. Association rules produce dependency rules based on strongly associated different attribute values [2]. There are many association rule–mining algorithms such as Apriori, Equivalence Class Transformation (Eclat), and frequent pattern growth (FPGrowth) [3].

Many researchers have used DM techniques such as regression, classification, clustering, and association rule mining in the field of education. They have applied these analyses in traditional education, web-based education (e-learning), and learning management systems [4]. Their studies have been conducted to accomplish many tasks related to students' learning processes, including providing feedback, predicting student performance, and detecting student behavior [4]. In traditional education, there are hidden patterns in data that are difficult for instructors and administrators to notice without the help of DM techniques. Such knowledge could be useful to students' learning processes in many ways. Processes such as students registering in courses, and instructors making or changing major plans or advising students, take time and effort during every semester. With the use of students' historical data that universities collect over the years, DM can help to enhance these processes. As a result, instructors and students can make better-informed decisions.

In this chapter, we report our work in finding an association between courses in a Computer Science (CS) program on the basis of students' grades. We report our work in discovering interesting rules by using different parameters such as support, confidence, lift, Kulczynski (Kulc), and the imbalance ratio (IR). We used the Apriori algorithm to mine data on undergraduate students majoring in CS at our university. The rest of this chapter is structured as follows: we review research in DM and association mining in Sect. 13.2. In Sect. 13.3 we give a detailed description of our experiments in using association rule mining to find interesting rules. We give our conclusion in Sect. 13.4.

13.2 Related Work

In this section, we review some related work that has used DM in education. We start by discussing previous work in terms of the classification methods used and the challenges associated with them. Then, we give a brief background of association rule mining and how we measure rules' interestingness. After that, we review related work that has used association rule mining in education.

13.2.1 Classification in Education

Classification is one of the most popular DM techniques that have been used in research to analyze educational data. Several studies have been conducted to review the literature in this area (e.g., a study by Shahiri et al. [5]) and to compare classification methods that have been used [5, 6]. For instance, Hämäläinen and Vinni [6] compared classification methods used on educational data on the basis of eight general criteria.

Many attributes have been used to predict students' performance, such as previous courses, standardized examination scores, or preuniversity scores. Different studies have used different attributes in their analyses. Shahiri et al. [5] reviewed important attributes used in prediction, such as the Cumulative Grade Point Average (CGPA) [7, 8], internal assessment [8, 9], students' demographic data [10, 11], external assessment, and psychometric factors. According to Shahiri et al. [5], the most popular task is classification, for which many algorithms have been used, such as DT [7–10, 12], artificial neural networks (ANNs) [7, 8, 12], naïve Bayes [9, 10, 12], K-nearest neighbor [10, 11], and SVM. Shahiri et al. [5] mentioned that most researchers have used CGPA and internal assessment as data attributes, and ANN and DT as classification techniques.

Several studies [7–11, 13] have focused on predicting students' performance by using DM techniques alone. The prediction of overall performance or performance in a specific course could be based on different attributes such as students' preuniversity data [7, 10], average of course attendance [11], grades in courses [13], marks in the course [8, 9], and behavioral features in an e-learning education system [12]. Most of the related studies that we reviewed used classification algorithms for prediction, including a nearest neighbor algorithm (IBk) [10], rule learners (OneR and JRip) [9, 10], classification based on an association rules algorithm [13], linear regression models [7], a classification and regression tree (C&RT), and chi-squared automatic interaction detection (CHAID) [8]. To compare the models, the studies used different evaluation measures such as accuracy (which was used by most of the studies), precision, recall, and F-measures [12].

13.2.2 Background to Association Rule Mining

Association rule mining is a DM technique that discovers interesting relations between attributes and then generates rules that represent these relations. These rules do not imply a causal relationship; for example, rule ($A \Rightarrow B$) does not mean that A causes B. However, the rules imply an association relationship between attributes (that is, those attributes go together) [14]. As we have mentioned above, the popular algorithms in association rule mining are Apriori, Eclat, and FPGrowth [3].

The Eclat algorithm uses a vertical data format method where each item is stored together with its list of TIDs (that is, the IDs of transactions that contain this item) [3, 15]. It is used only at the Apriori join step to generate the candidate itemsets [15]. To compute the support for an itemset, Eclat uses an intersection-based approach [15]. The intersection operation on TIDs is fast frequency counting, and it is advantageous for the vertical format that Eclat uses [16]. However, it has a drawback when the intermediate results of vertical TID lists become too large for the memory [16].

The FPGrowth algorithm was developed on the basis of a new data structure called a frequent pattern tree (FP-tree) [17]. It stores the database in the main memory using a combination of vertical and horizontal database layouts [17]; the transactions are stored in a tree structure, and each item has a linked list going through to all transactions that contain that item [17]. It avoids the cost of the candidate generation that Apriori does, by focusing on frequent pattern growth [3]. However, with long pattern data sets, it consumes more memory and performs badly [17, 18].

Apriori is a basic algorithm to find frequent item sets [3]; it is the one that we used in this work. Sometimes, Apriori produces a huge number of rules, making it difficult to use all of them. Because of that, and for other reasons that will be mentioned later, we must use interestingness measures. The measures of interestingness evaluate how interesting each rule is. This makes it easier to distinguish between the rules.

Measures of Interestingness In Apriori, patterns can be represented in the form of association rules. If the rule is ($A \Rightarrow B$), then, to measure its interestingness, we have many measures such as support, confidence, lift, Kulczynski, and the IR. Such measures are useful to distinguish between rules especially when there is a huge number of resulting rules. In addition, we should distinguish between uninteresting rules, which present an *obvious fact*, and new rules that could be interesting and useful. It is common in association rule mining to get a large number of rules that present facts we already know [19]. More details about the measures are as follows:

- The *absolute support* (also known as support count, count, or occurrence frequency) for itemsets A and B is the number of transactions that contain the itemset, and this is the probability: $P(A \cup B)$ [3]. The support for the rule ($A \Rightarrow B$) (sometimes referred to as relative support) is the count of transactions containing A and B to the number of transactions in the database [20].

$$\text{Support}\,(A \Rightarrow B) = \frac{\text{count}\,(A \cup B)}{n} \tag{13.1}$$

- The *confidence* of a rule ($A \Rightarrow B$) is the number (or count) of transactions that contain A and B to the number of transactions that contain A, and this is the conditional probability: $P(B|A)$ [3, 20]. The confidence value is calculated as shown below [20]:

$$\text{Confidence}\,(A \Rightarrow B) = \frac{\text{count}\,(A \cup B)}{\text{count}(A)} \tag{13.2}$$

- The *lift* of a rule measures how many more times A and B occur together in transactions than would be expected if A and B were statistically independent (not correlated). The lift value could be equal to 1, which would mean that A and B are independent and there is no correlation between them; or it could be less than 1, which would mean that the occurrence of A is negatively correlated with the occurrence of B; or it could be greater than 1, which would mean that A and B are positively correlated, and the occurrence of one implies the occurrence of the other. For example, if the lift of rule ($A \Rightarrow B$) is greater than 1, then we could say that A occurrence increases (or "lifts") the likelihood of B occurrence by a factor of the lift's value [3]. The lift of a rule can be calculated as follows [20]:

$$\text{Lift}\,(A, B) = \frac{P\,(A \cup B)}{P(A)P(B)}, \text{OR} \tag{13.3}$$

$$\text{lift}\,(A \Rightarrow B) = \frac{\text{confidence}\,(A \Rightarrow B)}{\text{support}(B)} \tag{13.4}$$

- The *Kulczynski* measure of A and B (abbreviated as Kulc) is an average of two confidence measures where the two confidence measures mean the conditional probabilities: the probability of itemset B given itemset A, and the probability of itemset A given itemset B. Its range is from 0 to 1, and the higher the values are, the closer the relationship between A and B is [3]. Kulc $= 0.5$ signifies neutral or balanced skewness, whereas the further the value is from 0.5, the closer the relationship is between the two item sets [21]. Kulc is defined by Eq. (13.3).

$$\text{Kulc}\,(A, B) = \frac{1}{2}\,(P\,(A|B) + P\,(B|A)) \tag{13.5}$$

- The *imbalance ratio* (IR) assesses the imbalance of two item sets (A and B) in rule implications [3]. Its range is from 0 to 1; IR $= 0$ means that the two directional implications between A and B ($A \Rightarrow B$ and $B \Rightarrow A$) are the same, which means it is not an interesting rule, whereas IR $= 1$ means it is a highly skewed or very interesting rule [21]. IR is calculated by Eq. (13.3):

$$\mathrm{IR}\,(A, B) = \frac{|\sup(A) - \sup(B)|}{\sup(A) + \sup(B) - \sup\,(A \cup B)} \tag{13.6}$$

Mining of association rules is a two-step process: first, we must find all frequent item sets that satisfy the minimum support threshold (Min_sup) specified by the user. Second, from these item sets, we must generate association rules that satisfy the minimum confidence threshold (Min_conf), and these rules are called strong [3]. However, when the items that we are interested in have support that is below (or far below) a user-specified minimum support threshold, they are called infrequent (or rare) items [22]. Rare items are caused by an imbalance in the data set where some items have a very high frequency, count, or support, while other items have very low support, and the resulting rules mostly cover only those items with high support [3, 22]. To mine the rare items, there are several methods that can be applied, such as balancing techniques or rare association rule–mining algorithms [23]. Another way is to set the minimum support threshold to a low value, which counts as the simplest way to mine rare items [22].

If we choose to set the minimum support threshold at a low value, that will produce a huge number of strong rules. When the resulting strong rules are huge in number, then use of a lift will help to rank or filter these rules [20]. Another reason to use a lift is to avoid misleading "strong" rules, because not all strong rules are interesting [3]. However, the lift is influenced by a null transaction—a transaction that does not contain any of the itemsets being examined. For example, a transaction that does not contain item sets *A* and *B* is a null transaction. If the value of a measure (of the interestingness of a rule) is not influenced by null transactions, it is called a *null-invariant measure*.

Null invariance is an important property for measuring association patterns in large transaction databases. So, a lift is not a null-invariant measure, whereas Kulc and IR are null-invariant measures because they are not influenced by a null transaction [3, 21]. Because of that, the lift has difficulty distinguishing interesting pattern association relationships in comparison with Kulc and IR. As far as we know (from reviewing the studies by Jiawei [3], Gupta and Arora [21], Wu et al. [24], and Gopalakrishnan [25]), we could use the three measures lift, Kulc and IR together as follows:

- If the Kulc value is close to 1, then the left-hand side (LHS) and the right-hand side (RHS) are positively correlated [3, 24].
- If the Kulc value is close to 0, then the LHS and RHS are negatively correlated [3, 24].
- If Kluc = 0.5 (that is, neutral) [3, 21, 24], then check the IR value.
  - If IR = 0, then it is not an interesting rule [21].
  - If the IR value is close to 1, then the rule might be worth looking at [21].

13.2.3 Association Rule Mining in Education

Association rule mining has been applied in education, mostly using the Apriori algorithm. The relevant studies have used Apriori for several objectives. Some of them have used it to predict students' performance [14, 26, 27] and to provide a good placement for a student by matching the organization's requirement with the student's profile [26]. In a study by Kasih et al. [14], the prediction of the students' final results was based on their performance in eight courses in the first four semesters, while in a study by Borkar and Rajeswari [27], the prediction was achieved by finding the association between attributes such as attendance and assignments. In a study by Ahmed et al. [28], the authors used students' academic and personal data to discover their impact on the students' performance; they extracted the association rules related to the impacts of sex, residence, retention, etc.

Other studies have used students' admission data [29, 30]. In a study by Mashat et al. [29], the data represented applicant student information and their status of being rejected or accepted for enrollment at the university. The researchers applied Apriori to the whole data set and then to the accepted applicants and the rejected applicants separately. The resulting rules were presented and interpreted with respect to the admissions office perspective [29]. Abdullah et al. [30] applied the SLPGrowth (Significant Least Pattern Growth) algorithm and two measures—lift and critical relative support (CRS)—to a student admission data set.

Damaševičius [31] aimed to improve the content of an informatics course. He used association rules and ranked course topics on the basis of their importance to the final course marks. He also proposed a novel metric called "cumulative interestingness" for assessing the strength of an association rule. Vranic et al. [19] used data on an electrical engineering fundamentals course with general data to predict the success of the next year's students in this course.

Upendran et al. [32] proposed a course recommendation system that suggested courses for new students. They used the Apriori algorithm to generate rules using previous students' marks in core courses and focused on rules with success as a consequence. These rules were used to suggest courses for new students where they had a high probability of success.

Some studies have associated courses with students' grades; for example, Buldu and Üçgün [33] were interested in the relation between courses in which students failed. In the study by Ahmed et al. [28], the resulting association rules showed that the grade in one course might depend on prerequisite courses. In the study by Upendran et al. [32], marks in core subjects such as Mathematics, Physics, Chemistry, Biology, Computer Science, and English were considered as attributes. Table 13.1 summarizes some of the studies' experimental settings and the Apriori parameters that they applied.

Table 13.1 Summary of studies that have used the Apriori algorithm

Study	Number of students/records	Min_supp	Confidence	Other parameters	Number of rules
Angeline [26]	21 students	0.33	0.75	Max rule length = 4, lift filtering = 1.1	127
Borkar and Rajeswari [27]	60 students	0.01	0.9, 0.87, and 0.7		2, 2, and 11
Kasih et al. [14]	146 records	0.2	0.8	–	6
Mashat et al. [29]	83K records (accepted and rejected applications)	0.4	0.75		4
	38K records (accepted applications)	0.6	Not mentioned	–	5
	45K records (rejected applications)	0.6	Not mentioned	–	3
Vranic et al. [19]	952 records	0.4	0.8	–	>500

Min_supp minimum support threshold

13.3 Experimental Settings and Results

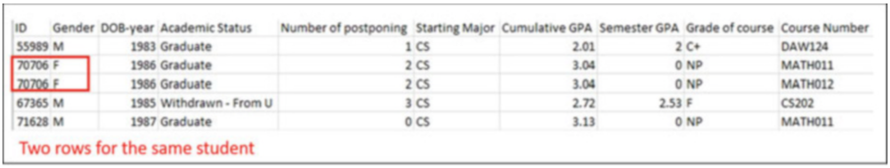
In this section, we present some of the experiments that we did to mine association rules by using the Apriori algorithm to find associations between CS courses based on the students' grades. We present the settings of the experiments, including the data preprocessing, and the results of these experiments.

13.3.1 Experimental Settings

In this subsection, we show how we preprocessed the data, then we show how we selected the values of the Apriori parameters.

Data Preprocessing Our aim in this study was to find an association between CS courses based on students' grades. Therefore, the items in the data set should be in the form (course = G) where course is the course code and G is the grade, G ∈ {A, B, C, D, F}. For example, CS140 = A, CS140 = C, CS322 = A, and CS322 = F.

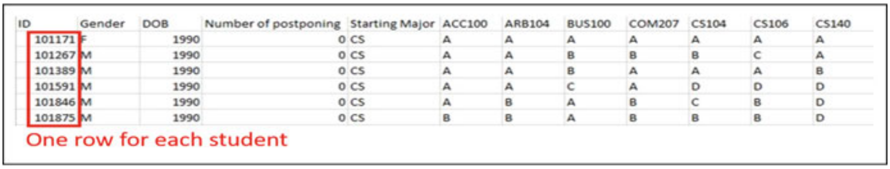
We started by translating the raw data set from Arabic into the English language, and we selected CS students' data only. Then we transformed the "date of birth" attribute from "dd/mm/yyyy" to "yyyy"; the results are presented as a screenshot (Fig. 13.1). Notice that a student is represented by multiple rows. For instance, if a student studied 50 courses, then he or she would have 50 rows: one row for each course.



ID	Gender	DOB-year	Academic Status	Number of postponing	Starting Major	Cumulative GPA	Semester GPA	Grade of course	Course Number
55989	M	1983	Graduate	1	CS	2.01	2	C+	DAW124
70706	F	1986	Graduate	2	CS	3.04	0	NP	MATH011
70706	F	1986	Graduate	2	CS	3.04	0	NP	MATH012
67365	M	1985	Withdrawn - From U	3	CS	2.72	2.53	F	CS202
71628	M	1987	Graduate	0	CS	3.13	0	NP	MATH011

Two rows for the same student

Fig. 13.1 Data set after deletion of some attributes



ID	Gender	DOB	Number of postponing	Starting Major	ACC100	ARB104	BUS100	COM207	CS104	CS106	CS140
101171	F	1990	0	CS	A	A	A	A	A	A	A
101267	M	1990	0	CS	A	A	B	B	B	C	A
101389	M	1990	0	CS	A	A	B	A	A	A	B
101591	M	1990	0	CS	A	A	C	A	D	D	D
101846	M	1990	0	CS	A	B	A	B	C	B	D
101875	M	1990	0	CS	B	B	A	B	B	B	D

One row for each student

Fig. 13.2 Data set after transformation of rows to columns

After that, the grades were merged into five categories (A, B, C, D, F): A+ was merged with A, B+ was merged with B, etc. Any grades that did not belong to any of the five categories were deleted. For instance, the grades in course CS480 (Practical Training) had two values—NP signified success without a degree and NF signified failure without a degree—and most of the students got NP, so we deleted them. We also deleted the attributes that did not seem useful, especially when they had the same value for all students, such as major and residential area attributes.

Then, to represent each student by one row, we transformed rows into columns; the results are presented as a screenshot (Fig. 13.2), where we present some of the attributes. With this transformation, we kept only the first trail (which is the first grade that a student got when he or she studied the course for the first time). To handle missing data, we selected the graduated students, since their records had fewer missing data/grades. We had 833 CS graduated students.

At the end, to prepare the data set for association rule mining, and to find the association between courses, we selected only the attributes that represented CS courses—general courses and computer specialized (mandatory and elective) courses—which meant that we deleted the rest of the attributes. In the CS study plan, we had 60 courses; for each course, student could get one of five grades (A, B, C, D, F), which meant we had ($60 \times 5 = 300$ items) in the form (course = G).

Setting of Apriori Parameters

After conducting many experiments and reading related works, we set the minimum support at 0.01 and confidence at 0.8. This minimum support meant that approximately 94% of the items would be included in the mining process, while items that had support of less than 0.01 would not be included. It also meant 1% of students, which was either 8 students out of 833 students ($1\% \times 833 = 8.33$) or 4 students out of 483 students [29].

Table 13.2 Kulczynski (*Kulc*) and imbalance ratio (*IR*) values and their meanings

Relationship between LHS and RHS	Kulc/IR	Value
Positive correlation	Kulc close to 1	Kulc $\geq$ 0.7
Negative correlation	Kulc close to 0	Kulc $\leq$ 0.3
Neutral	Kulc close to 0.5	0.7 > Kulc > 0.3
Very imbalanced	IR close to 1	IR $\geq$ 0.8
Imbalanced	IR relatively close to 1	IR $\geq$ 0.6
Balanced	IR close to 0	IR $\leq$ 0.3
Neutral	Kulc and IR close to 0.5	0.6 < IR < 0.3 and 0.7 > kulc > 0.3

LHS left-hand side, *RHS* right-hand side

Besides using support and confidence to measure the rules' interestingness, we decided to use lift, Kulc, and IR for the reasons mentioned in Sect. 13.2. In addition, on the basis of the relation between the three measures, we wrote the pseudocode below to evaluate the rules. As in the studies by Bramer [20] and Angeline [26], we used lift to rank or filter the rules, then we applied the pseudocode. Table 13.2 lists the values that we chose for Kulc and IR.

The itemset number was set at (2, 3, 4, and 5) to specify the length of the rules. This helped us to focus on each subset of the rules; for example, when we analyzed 2-itemset rules, we focused on the relationship between two courses only, so if the relation was positive, then getting a high grade in one course would be associated with getting a high grade in the other one, and vice versa.

```
Pseudocode Used for Rule Filtering
  Sort according to lift and lift > 1
  If Kulc close to 1 then
       LHS and RHS are positively correlated
  Else if Kulc close to 0 then
       LHS and RHS are negatively correlated
  Else if Kulc close to 0.5 then
       // Neutral, use IR to help find the imbalance
       If IR very close to 1 then
       // Imbalanced
         The rule might be worth looking at: very imbalanced case
       Else if IR relatively close to 1 then
       // Imbalanced
            The rule might be worth looking at: imbalanced case
       Else if IR close to 0 then
       // Balanced
            Not interesting rule
       Else "neutral"
       // Kulc close to 0.5 and IR between 0.3 and 0.6
```

13.3.2 Results of the Experiments

In the experiments, we used two data sets. In the first data set, the instances were CS graduated students (833 students). In the second data set, we kept students who failed at least one course (483 students or rows) and we kept items that contained failing grades only (CS140 = F, CS322 = F, etc.). We present the two experiments as follows:

- The first experiment focused on the association of courses based on success and failure, and it used data set 1.
- The second experiment focused on association of courses based on failure, and it used data set 2.

Also, for each experiment, we generated rules with itemset numbers from 2 to 5. Because of space limitations, we present and discuss the resulting rules only for 2- and 3-itemsets, presenting 2-itemset rules for the first experiment and 3-itemset rules for the second experiment.

Association of Courses Based on Success and Failure In this experiment, we set Apriori parameters as listed in Table 13.3.

The results of the 2-itemset were 1138 rules, and we were interested in rules where the LHS and RHS were positively correlated (103 rules) or worth looking at, especially in a very imbalanced case (453 rules). We first explore and present the positive correlation rules, then those worth looking at. After that, we present the rules that contained grade F.

Positive Correlation Rules In Table 13.4, we present the top five rules with the highest lift values where LHS and RHS were positively correlated. As can be seen, these rules had high confidence and the support values were relatively low to high, being in the range of 0.3–0.8, which meant that 30–80% of the students achieved these grades in these courses. The confidence values were in the range of 0.8–0.9, indicating that these rules were found to be true 80–90% of the time; in other words, in 80–90% of instances where a student achieved LHS, he or she would achieve RHS too.

As an example of the rules presented in Table 13.4, **rule (1043) {ARB104=A} ⇒ {IDE133=A} (sup = 0.384 and conf = 0.821)** means that 38% of the students achieved an A in both course ARB104 and course IDE133, and the rule was found

Table 13.3 Apriori parameter values

Parameter	Value
Min_supp	0.01
Confidence	0.8
Itemset number	2, 3, 4, and 5

Min_supp minimum support threshold

Table 13.4 Top five 2-itemset rules where the left-hand side (*LHS*) and right-hand side (*RHS*) are positively correlated

Rule no.	Rules	Support	Confidence	Lift	Kulc	IR	LHS and RHS correlation
1043	{ARB104 = A} ⇒ {IDE133 = A}	0.384	0.821	1.312	0.717	0.222	Positive
1053	{ECO100 = A} ⇒ {QUR401 = A}	0.415	0.824	1.241	0.725	0.212	Positive
1062	{BUS100 = A} ⇒ {COM207 = A}	0.403	0.806	1.188	0.700	0.229	Positive
1063	{BUS100 = A} ⇒ {QUR451 = A}	0.419	0.837	1.186	0.715	0.261	Positive
1055	{ECO100 = A} ⇒ {QUR451 = A}	0.421	0.836	1.184	0.716	0.256	Positive

IR imbalance ratio, *Kulc* Kulczynski

Table 13.5 Number of 2-timeset rules containing one of the Holy Quran (*QUR*) courses

QUR on RHS	Number of rules
{QUR101 = A}	139
{QUR201 = A}	127
{QUR251 = A}	105
{QUR351 = A}	103
{QUR301 = A}	89
{QUR151 = A}	79
{QUR451 = A}	75
{QUR401 = A}	41

RHS right-hand side

to be true for 82% of those students; in other words, in 82% of instances where a student got an A in ARB104, he or she would get an A in IDE133 too.

In addition, we noted the following about the positive correlation rules:

- All rules that had a positive correlation associated getting grade A in LHS courses with getting grade A in RHS courses.
- Examples of the courses that appeared in the rules are general courses and two mandatory courses, CS492 and CS493 (Senior Project in Computer Science 1 and Senior Project in Computer Science 2). However, grade A was the only grade that appeared.
- Most of these rules were in the form ({X = A} ⇒ {QURxxx = A}), where "X" represented a course and "xxx" represented the code of the QUR (Holy Quran) courses. Table 13.5 shows the number of rules (in this form) for each QUR course.

The high support indicates that these item sets {course1 = A, course2 = A} appeared frequently in the data set; most of the students got a grade of A in these courses (the general courses and two mandatory courses (CS492 and CS493)). For example, Fig. 13.3 shows the frequency or support of the QUR courses with grades

(A, B, C, D, F) and, as we said, (QURxxx = A) items were the most frequent. As a conclusion, these rules may not be interesting, since getting an A in these courses could count as an obvious fact, and any rule that supports an obvious fact is not an interesting rule.

Rules Worth Looking At In Table 13.6 we present the top five rules that were worth looking at (very imbalanced cases). These rules had high confidence values and low to relatively low support values; the support values were in the range of 0.01–0.1, meaning that 1–10% of the students achieved grades (A, B, C, D, F) in the LHS courses and grade A in the RHS courses. The confidence values are in the range of 0.8–1, which indicates that these rules were found to be true 80–100% of the time; in other words, 80–90% of instances where a student achieved in LHS courses, he

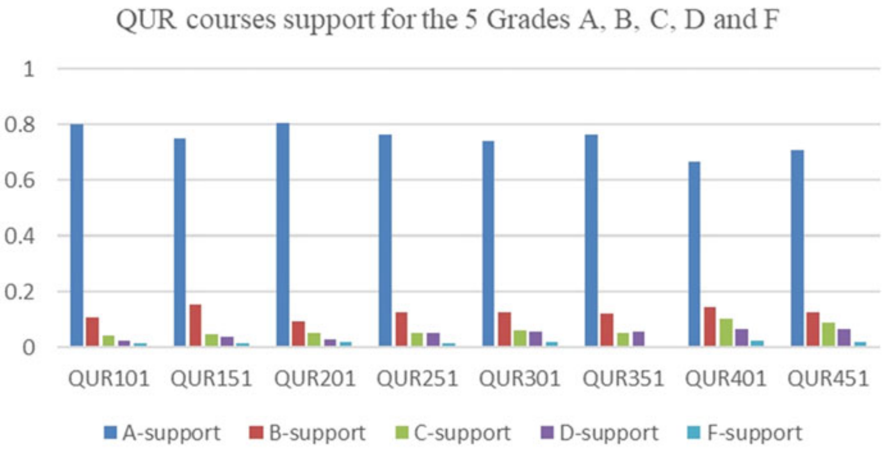


Fig. 13.3 Support of Holy Quran (*QUR*) courses with grades (A, B, C, D, and F)

Table 13.6 Top five “worth looking at” 2-itemset rules

Rule no.	Rules	Support	Confidence	Lift	Kulc	IR	LHS and RHS correlation
45	{CS439 = A} ⇒ {CS322 = A}	0.014	0.800	4.660	0.442	0.877	Very imbalanced case
46	{CS439 = A} ⇒ {CS370 = A}	0.014	0.800	4.191	0.438	0.889	Very imbalanced case
47	{CS439 = A} ⇒ {MATH227 = A}	0.014	0.800	3.471	0.431	0.908	Very imbalanced case
106	{CS438 = A} ⇒ {CS391 = A}	0.025	0.955	2.432	0.509	0.930	Very imbalanced case
49	{CS439 = A} ⇒ {ENG208 = A}	0.018	1.000	2.394	0.522	0.957	Very imbalanced case

IR imbalance ratio, *Kulc* Kulczynski, *LHS* left-hand side, *RHS* right-hand side or she would achieve in RHS courses too. A "worth looking at" rule may or may not imply an interesting relationship.

We noticed the following about the "worth looking at" rules:

- All rules that had a positive correlation associated getting grade (A, B, C, D, F) in LHS courses with getting grade A in RHS courses.
- As we saw in the (2-timeset rules with positive correlation), most of the rules were in the form (X = G ⇒ {QURxxx = A}), where G represented one of the grades (A, B, C, D, F), and we said that these rules were mostly not interesting, since they could count as obvious facts because of the high support of (QURxxx = A) items.

As an example of the rules presented in Table 13.6, **rule (45) {CS439 = A} ⇒ {CS322 = A}** implied that maybe there was an association between getting an A in both course CS439 (Cloud Computing) and course CS322 (Operating Systems), and we know from the CS study plan that one of the prerequisite courses to register in CS439 (Cloud Computing) was success in CS322 (Operating Systems). So, an association between getting an A in both courses is logical, since it confirms what we already know. The same applies to **rule (46) {CS439 = A} ⇒ {CS370 = A}**, where there might an association between getting an A in both course CS439 (Cloud Computing) and course CS370 (Introduction to Databases), and since the CS study plan did not link these two courses, that could mean that this rule is a new rule and therefore could be an *interesting rule*.

Table 13.7 shows some of the rules that contained CS courses on the LHS, arranged by the support values from largest to smallest. For example, the first three rules applied to approximately 5% of the students (5% of 833 students = 41 students), and they were true 90% or 80% of the time. They showed the association between getting an A in CS401 (Computational Numerical Analysis) and getting an A in CS493 (Senior Project in Computer Science 2), as in **rule (262)**; getting

Table 13.7 "Worth looking at" 2-itemset rules where the left-hand side (*LHS*) items are Computer Science courses

Rule no.	Rules	Support	Confidence	Lift	Kulc	IR	LHS and RHS correlation
262	{CS401 = A} ⇒ {CS493 = A}	0.059	0.907	2.026	0.519	0.844	Very imbalanced case
258	{CS401 = A} ⇒ {CS391 = A}	0.058	0.889	2.264	0.518	0.820	Very imbalanced case
264	{CS401 = A} ⇒ {CS492 = A}	0.058	0.889	1.903	0.506	0.848	Very imbalanced case
235	{CS471 = A} ⇒ {CS492 = A}	0.052	0.878	1.879	0.494	0.861	Very imbalanced case
201	{CS451 = A} ⇒ {CS492 = A}	0.036	0.857	1.835	0.467	0.898	Very imbalanced case

IR imbalance ratio, *Kulc* Kulczynski, *RHS* right-hand side an A in CS391 (Seminar), as in **rule (258)**; and getting an A in CS392 (Senior Project in Computer Science 1), as in **rule (264)**.

"Worth Looking At" Rules Containing Failed Courses We present some of the 2-timeset rules that contain grade F in Table 13.8; they are worth looking at (a very imbalanced case) on the basis of both their Kulc value and their IR value. Note that they have high confidence in the range of 0.8–1 and low support in the range of 0.011–0.067; that is, 1.1–6.7% of the students who got grade F in LHS courses and got grade A in QUR courses (RHS). Also, these rules were found to be true 80–100% of the time. As we know, a "worth looking at" rule may or may not imply an interesting relationship. From our point of view, since most of the students got an A in QUR courses, those students would get different grades (A, B, C, D, F) in the rest of their courses. So, it is an obvious fact, which means these rules do not seem to be interesting. For example, the support value for **rule (13)** is equal to the support value for item (CS430 = F) on the LHS; the same students who got an F in CS430 (approximately 11 students) got an A in QUR courses.

Association of Courses Based on Failure In this experiment, to find the association between failure and getting an F grade in courses, we used the second data set, which contained CS graduated students who failed in at least one course (483 rows), with 60 courses, and each course where a student got an F grade (so, 60 items in form (course = F)). Table 13.9 presents the Apriori parameter values that we used. We present the 3-itemset rules because there was no resulting 2-itemset rule.

Table 13.8 2-Itemset rules that contain an F grade on the left-hand side (*LHS*), on the right-hand side (*RHS*), or both

Rule no.	Rules	Support	Confidence	Lift	Kulc	IR	LHS and RHS correlation
13	{CS430 = F} ⇒ {QUR251 = A}	0.014	1.000	1.312	0.509	0.981	Very imbalanced case
11	{CS412 = F} ⇒ {QUR201 = A}	0.011	1.000	1.241	0.507	0.987	Very imbalanced case
12	{CS430 = F} ⇒ {QUR151 = A}	0.013	0.917	1.228	0.467	0.979	Very imbalanced case
20	{CS451 = F} ⇒ {QUR151 = A}	0.014	0.857	1.148	0.438	0.974	Very imbalanced case
21	{CS451 = F} ⇒ {QUR351 = A}	0.014	0.857	1.121	0.438	0.975	Very imbalanced case

IR imbalance ratio, *Kulc* Kulczynski

Table 13.9 Apriori parameter values

Parameter	Value
Min_supp	0.01
Confidence	0.8
Itemset number	2, 3, 4, and 5

Min_supp minimum support threshold

Table 13.10 Top five "worth looking at" 3-itemset rules for the failure data set

Rule no.	Rules	Support	Confidence	Lift	Kulc	IR	LHS and RHS correlation
18	{CS340 = F, CS471 = F} ⇒ {CS401 = F}	0.012	0.857	8.809	0.492	0.833	Very imbalanced case
17	{CS403 = F, CS471 = F} ⇒ {CS401 = F}	0.010	0.833	8.564	0.470	0.854	Very imbalanced case
29	{CS215 = F, CS370 = F} ⇒ {CS401 = F}	0.010	0.833	8.564	0.470	0.854	Very imbalanced case
13	{QUR101 = F, STA111 = F} ⇒ {MATH227 = F}	0.010	1.000	7.318	0.538	0.924	Very imbalanced case
5	{MATH227 = F, QUR101 = F} ⇒ {CS344 = F}	0.010	1.000	7.103	0.537	0.926	Very imbalanced case

IR imbalance ratio, *Kulc* Kulczynski, *LHS* left-hand side, *RHS* right-hand side

The results of the 3-itemset were 43 rules; there was no rule where LHS and RHS were positively correlated. Table 13.10 shows the top five "worth looking at" rules (a very imbalanced case). These rules had high confidence values and low support values. The support values were in the range of 0.01–0.024, which meant that 1–2.4% of the students achieved grade F in these courses, while the confidence values in the range of 0.8–1 indicated that these rules were found to be true 80–100% of the time; in other words, 80–100% of the time that a student achieved LHS, he or she would achieve RHS too. For example: **rule (18)** {**CS340 = F, CS471 = F**} ⇒ {**CS401 = F**} indicates that 1.2% of the students got an F in all three courses of CS340 (Artificial Intelligence), CS471 (Database Management Systems), and CS401 (Computational Numerical Analysis), and 85% of the times that a student failed in CS340 and CS471, he or she would fail in CS401 too.

13.4 Conclusion

In this chapter, we have reviewed some of the literature that has used data-mining (DM) techniques (such as classification and association rule mining) in education for exploring patterns and extracting knowledge. We conducted experiments to extract and mine interesting rules from data on undergraduate Computer Science (CS) students, using the Apriori algorithm, and we presented the settings of these experiments and some of the results. We answered the questions of the research and we used lift, Kulczynski (Kulc), and the imbalance ratio (IR) to measure the interestingness of rules, along with their support and confidence. We explained how we preprocessed the data and how we set the values for minimum support, minimum confidence, Kulc, and IR. Our results showed correlation between some courses when students obtained A and F grades. Other grades (B, C, and D) did not show correlation between courses in the 2-itemset rules. In addition, they showed that most of the interesting rules had support higher than 1% and confidence higher than 80%. Therefore, we cannot confirm or deny previous opinions that have associated some courses with each other, such as associating success in mathematics with success in programming. We also plan to apply classification algorithms to the data set.

References

1. Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Pearson Addison Wesley, Boston (2005)
2. Frank, E., Witten, I.H.: Data Mining: Practical Machine Learning Tools and Techniques, vol. 54, no. 2. Morgan Kaufman, San Francisco (2011)
3. Jiawei, H., Kamber, M., Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. Elsevier, New York (2012)
4. Romero, C., Ventura, S.: Educational data mining: a review of the state of the art. IEEE Trans. Syst. Man Cybernet. Part C Appl. Rev. **40**(6), 601–618 (2010)
5. Shahiri, A.M., Husain, W., Rashid, N.A.: A review on predicting student's performance using data mining techniques. Procedia Comput. Sci. **72**, 414–422 (2015)
6. Hämäläinen, W., Vinni, M.: Classifiers for educational data mining. In: Handb. Educ. Data Mining, Data Min. Knowl. Discov. Ser., pp. 57–71 (2010)
7. Ibrahim, Z., Rusli, D.: Predicting students' academic performance: comparing artificial neural network, decision tree and linear regression. In: 21st Annual SAS Malaysia Forum, 5th September 2007
8. Ogor, E. N.: Student academic performance monitoring and evaluation using data mining techniques. In: Electronics, Robotics and Automotive Mechanics Conference (CERMA 2007), pp. 354–359 (2007)
9. Al-Barrak, M.A., Al-Razgan, M.S.: Predicting students' performance through classification: a case study. J. Theor. Appl. Inf. Technol. **75**(2), 167–175 (2015)
10. Verma, K., Singh, A., Verma, P.: A review on predicting student performance using data mining method. Int. J. Curr. Eng. Sci. Res. **3**(1), 127–132 (2016)
11. Harwati, Alfiani, A.P., Wulandari, F.A.: Mapping student's performance based on data mining approach (a case study). Agric. Agric. Sci. Procedia. **3**, 173–177 (2015)
12. Amrieh, E.A., Hamtini, T., Aljarah, I.: Mining educational data to predict student's academic performance using ensemble methods. Int. J. Database Theory Appl. **9**(8), 119–136 (2016)
13. Badr, G., Algobail, A., Almutairi, H., Almutery, M.: Predicting students' performance in university courses: a case study and tool in KSU Mathematics Department. Procedia Comput. Sci. **82**, 80–89 (2016)
14. Kasih, J., Ayub, M., Susanto, S.: Predicting students' final passing results using the Apriori algorithm. World Trans. Eng. Technol. Educ. **11**(4), 517–520 (2013)
15. Pramod, P.S., Vyas, O.P.: Survey on frequent itemset mining algorithms. Int. J. Comput. Appl. **1**(15), 94–100 (2010)
16. Zaki, M.J., Gouda, K.: Fast vertical mining using diffsets. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining—KDD'03, p. 326 (2003)

17. Solanki, S., Soni, N.: A survey on frequent pattern mining methods Apriori, Eclat, FP growth. Int. J. Comput. Tech. **X**(X), (2014)
18. Suman, M., Anuradha, K.T., Ramakrishna, A.: A frequent pattern mining algorithm based on Fp-tree structure and Apriori algorithm. Technicaljournals.Org. **2**(1), 114–116 (2012)
19. Vranic, M., Pintar, D., Skocir, Z.: The use of data mining in education environment. In: 2007 9th International Conference on Telecommunications, pp. 243–250 (2007)
20. Bramer, M.: Principles of Data Mining. Springer, New York (2016)
21. Gupta, D., Arora, H.: Market basket analysis using Apriori and correlation measures. Int. J. Innov. Res. Sci. **6**(6), 10282–10286 (2017)
22. Romero, C., Romero, J.: Mining rare association rules from e-learning data. In: 3rd International Conference on Educational Data Mining, pp. 171–180 (2010)
23. Koh, Y.S., Nathan, R.: Rare association rule mining: an overview. In: Rare Association Rule Mining and Knowledge Discovery: Technologies for Infrequent and Critical Event Detection, IGI Global, pp. 1–14 (2009)
24. Wu, T., Chen, Y., Han, J.: Re-examination of interestingness measures in pattern mining: a unified framework. Data Min. Knowl. Discov. **21**(3), 371–397 (2010)
25. Gopalakrishnan, A.: A Multifaceted Data Mining Approach to Analyzing College Students' Persistence and Graduation. San Francisco State University, San Francisco (2016)
26. Angeline, D.M.D.: Association rule generation for student performance analysis using Apriori algorithm. SIJ Trans. Comput. Sci. Eng. Appl. **1**(1), 12–16 (2013)
27. Borkar, S., Rajeswari, K.: Predicting students academic performance using education data mining. Int. J. Comput. Sci. Mob. Comput. **2**(7), 273–279 (2013)
28. Ahmed, S., Paul, R., Hoque, A.S.M.L.: Knowledge discovery from academic data using Association Rule Mining. In: 2014 17th International Conference on Computer and Information Technology (ICCIT), pp. 314–319 (2014)
29. Mashat, A.F., Fouad, M.M., Yu, P.S., Gharib, T.F.: Discovery of association rules from university admission system data. Int. J. Mod. Educ. Comput. Sci. **5**(4), 1–7 (2013)
30. Abdullah, Z., Herawan, T., Deris, M.M.: Discovering interesting association rules from student admission dataset. Lect. Notes Electr. Eng. **285**, 135–142 (2014)
31. Damaševičius, R.: Analysis of academic results for informatics course improvement using association rule mining. In: Information Systems Development: Towards a Service Provision Society, pp. 357–363 (2009)
32. Upendran, D., Chatterjee, S., Sindhumol, S., Bijlani, K.: Application of predictive analytics in intelligent course recommendation. Procedia Comput. Sci. **93**, 917–923 (2016)
33. Buldu, A., Üçgün, K.: Data mining application on students' data. Procedia. Soc. Behav. Sci. **2**(2), 5251–5259 (2010)