Analysis of Student Result Data Using Association Rule Mining

Nilesh Mahajan
Professor, Institute of Management and Entrepreneurship Development, Bharati Vidyapeeth Deemed University, Pune, Maharashtra, India

&

Jyoti Namdeo
Research Scholar, Institute of Management and Entrepreneurship Development, Bharati Vidyapeeth Deemed University, Pune, Maharashtra, India

Abstract:
Student performance in university courses is of great concern to the higher education managements where several factors may affect the performance. Data mining may give an institution the information necessary to take action before a student drops out. In the present paper we applied predictive apriori algorithm for finding the hidden association between different subjects. For a student to successfully pass the MCA course, a sound knowledge of programming is must. The associated model thus generated will help that student in the future who are getting similar grades in the elementary algorithm, procedural oriented language and object oriented programming. In the curriculum design two associated subjects should be placed in adjacent semesters so that the expertise gained in previous subject can be used in the associated subject coming next in the semester.

Keywords: Educational data mining, discretization, association rule mining,

Educational Data Mining
Gartner Inc.’s defined data mining as: “…the process of discovering meaningful new correlations, patterns, and trends by sifting through large amounts of data stored in repositories, and by using pattern recognition technologies, as well as statistical and mathematical techniques.” (Gartner Group)

Educational data mining (also referred to as “EDM”) is defined as the area of scientific inquiry centered around the development of methods for making discoveries within the unique kinds of data that come from educational settings, and using those methods to better understand students and the settings which they learn in. (Baker, 2008)

Literature review
Student performance in university courses is of great concern to the higher education managements where several factors may affect the performance. Data mining may give an institution the information necessary to take action before a student drops out. The performance of a student in a course is assessed by a variety of assessment techniques i.e. assignments, projects, laboratory work, semester end examinations etc. They used apriori algorithm in association rule mining and their analysis revealed that more students got excellent grades in supervised assessment but failed to attain similar level of performance in the unsupervised assessments (Anwar and Naseer, 2011).

The paper analyzed the potential use of one of the data mining technique called association rule mining in enhancing the quality of student’s performances and to predict the performance of the students as poor, good and excellent (Bambrah et.al. 2014).

The paper observed that the students who have scored badly in their Graduation have done relatively well in their Post Graduation in the subjects which are common in both Graduate and Post Graduate courses. They used of one of the data mining technique called association rule mining in enhancing the quality of students’ performances at Post Graduation level. The mined association rules reveal various factors like student’s interest, curriculum design; teaching and assessment methodologies that can
affect students who have failed to attain a satisfactory level of performance in the Post Graduation level (Varun and Anupama, 2012).

Suchita and Rajeswari concluded that the student’s performance level can be improved in university result by identifying students who perform poorly in unit Test, Attendance, Assignment and graduation and giving them additional guidance to improve the university result by using the apriori algorithm in association rule mining.

The author concluded that CHAID prediction model was useful to analyze the interrelation between variables that are used to predict the outcome on the performance at higher secondary school education. The features like medium of instruction, marks obtained in secondary education, location of school, living area and type of secondary education were the strongest indicators for the student performance in higher secondary education (Ramaswami and Bhaskaran, 2010)

**Data collection and preprocessing**

We opted for the professional course of Masters in Computer Applications as my field is of the same background. Being a research scholar in computer applications, data mining on MCA students was a natural choice. Hence it will be easy to understand the data coming from result of MCA. For the present study, the secondary data was obtained from the examination section of a University, identity of which was not disclosed to maintain the confidentiality.

Masters in Computer Applications (MCA) is a professional program of three years having six semesters. Each semester consists of 7 subjects and thus the total number of subjects including compulsory and elective was about 40.

These 40 subjects are called as the conditional subjects or the conditional attributes. The final result of the student is declared after the completion of the sixth semester. This final result is called as decision attribute. The main emphasis of present work was to study the effect of different subjects in the result of the students. Therefore in the collection of data we recorded the marks of different subjects. We had not collected the result of those students who got failed in one or more subject.

We took the marks of those subjects which are considered important in respect to the professional course of computer applications. As we were interested to find the effect of these subjects on the final result we gave emphasis only on the theory exam marks leaving aside the internal exam marks hence had not incorporated the internal marks of the subjects.

Preliminary data of 53 students was taken from the batch of 2007. Final marks to students were given out of 100. The 100 marks were divided into ratio of 80:20 in which, 80 marks were allotted for theory paper whereas 20 marks were kept for internal assessment.

Initially more than 20 attributes have been collected. Out of these, 12 conditional attributes and one class attribute have been chosen because professional teaching has to be centered on these main subjects and have to be taught theoretically. The chosen 12 subjects along with their codes are given in the following Table 1.

**Table 1: Description of Subjects with Code**

<table>
<thead>
<tr>
<th>Attribute Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Elementary Algorithm</td>
</tr>
<tr>
<td>103</td>
<td>Procedure Oriented Programming</td>
</tr>
<tr>
<td>201</td>
<td>Data Structure</td>
</tr>
<tr>
<td>202</td>
<td>Operating System</td>
</tr>
<tr>
<td>203</td>
<td>Data Base Management System</td>
</tr>
<tr>
<td>301</td>
<td>Software Engineering</td>
</tr>
<tr>
<td>302</td>
<td>Computer Networking</td>
</tr>
<tr>
<td>303</td>
<td>Object Oriented Programming</td>
</tr>
<tr>
<td>401</td>
<td>UML</td>
</tr>
<tr>
<td>402</td>
<td>Unix</td>
</tr>
<tr>
<td>501</td>
<td>Software Project Management</td>
</tr>
<tr>
<td>502</td>
<td>Artificial Intelligence</td>
</tr>
</tbody>
</table>
We had taken 12 subjects which were considered important for the understanding of computers and programming language. In other words we can say that these 12 subjects were the backbone of the professional course of computer applications. Hence the final result of the student was recalculated on the basis of the subjects chosen which consists of theory marks only. Since the theory marks were obtained from 80 marks in their university exams, so we had converted the 80 marks into the percentage marks. Thus consecutively we converted each mark into percentage. By taking the total of such obtained marks we got the final total of all 12 subjects taken. We again converted the final total into the percentage.

After getting the percentage each subject percentage as well as the final percent is transformed into the grades. The following table Table 2 shows how the grading was done on the marks.

<table>
<thead>
<tr>
<th>MARKS</th>
<th>GRADE</th>
</tr>
</thead>
<tbody>
<tr>
<td>[75-100]</td>
<td>O</td>
</tr>
<tr>
<td>[70-74.9]</td>
<td>A+</td>
</tr>
<tr>
<td>[65-69.9]</td>
<td>A</td>
</tr>
<tr>
<td>[60-64.9]</td>
<td>B+</td>
</tr>
<tr>
<td>[55-59.9]</td>
<td>B</td>
</tr>
<tr>
<td>[50-54.9]</td>
<td>C+</td>
</tr>
<tr>
<td>[45-49.9]</td>
<td>C</td>
</tr>
<tr>
<td>[40-44.9]</td>
<td>D</td>
</tr>
<tr>
<td>[00-39.9]</td>
<td>F</td>
</tr>
</tbody>
</table>

The result of the student depends on the conditional attributes. The final result of student is declared by evaluating the average marks scored at the completion of the program. These marks contain only the total of theoretical marks out of 80. Table 3 illustrates the Discretization of decision attribute (final results).

<table>
<thead>
<tr>
<th>MARKS</th>
<th>GRADES</th>
<th>EXPLAINATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>[70-100]</td>
<td>O</td>
<td>D, Distinction</td>
</tr>
<tr>
<td>[60-69.99]</td>
<td>A+</td>
<td>F, First class</td>
</tr>
<tr>
<td>[55-59.99]</td>
<td>B+</td>
<td>HS, Higher Second class</td>
</tr>
<tr>
<td>[50-54.99]</td>
<td>B</td>
<td>S, Second class</td>
</tr>
<tr>
<td>[40-49.99]</td>
<td>C</td>
<td>P, Pass class</td>
</tr>
<tr>
<td>[00-39.99]</td>
<td>F</td>
<td>F, Fail class</td>
</tr>
</tbody>
</table>

**Methodology**

The preprocessed data set was given to the WEKA, an open source data mining tool. The data was converted into the csv format for the compatibility of WEKA tool. The Predictive Apriori algorithm given by Tobias Scheffer(2004) is a part of WEKA tool. We applied the same algorithm for finding the hidden association between different attributes.

In comparison to the Apriori algorithm (Agrawal and Shrikant, 1996), the Predictive Apriori algorithm returns the best n rules which maximize the expected predictive accuracy and the user only has to specify how many rules user wants to be presented. This is a more natural parameter than minsup and minconf, required by the Apriori algorithm.
Table 1: Algorithm PredictiveApriori (Tobias Scheffer, 2004): discovery of $n$ most predictive association rules.

1. **Input**: $n$ (desired number of association rules), database with items $a_1 \ldots a_k$.
2. **Let** $\tau = 1$.
3. **For** $i = 1 \ldots k$ **Do**: Draw a number of association rules $[x \Rightarrow y]$ with $i$ items at random. Measure their confidence (provided $s(x) > 0$). Let $\pi(c)$ be the distribution of confidences.
4. **For** all $c$, let $\pi(c) = \frac{\Sigma_{i=1}^{k} \pi_i(c)}{\Sigma_{i=1}^{k} (\pi_i(c))(2^i-1)}$.
5. **Let** $X_0 = \{\emptyset\}$; **Let** $X_i = \{a_1 \ldots a_k\}$ be all item sets with one single element.
6. **For** $i = 1 \ldots k$ **While** ($i = 1$ or $X_{i-1} \neq \emptyset$).
   (a) **If** $i > 1$ **Then** determine the set of candidate item sets of length $l$ as
   $X_i = \{x \cup x' \mid x, x' \in X_{i-1}, |x \cup x'| = i\}$. Generation of $X_i$ can be optimized by considering only item sets $x$ and $x' \in X_{i-1}$ that differ only in the element with highest item index. Eliminate double occurrences of item sets in $X_i$.
   (b) Run a database pass and determine the support of the generated item sets. Eliminate item sets with support less than $\tau$ from $X_i$.
   (c) **For** all $x \in X_i$, **Call Rule Gen(x)**.
   (d) **If** best has been changed, **Then** increase $\tau$ to be the smallest number such that $E(c[1, \tau] > E(c(best[n])) | c(best[n]), s(best[n]))$ (refer to Equation 6. If $\tau >$ database size, **Then** Exit.
   (e) **If** $\tau$ has been increased in the last step, **Then** eliminate all item sets from $X_i$ which have support below $\tau$.
7. **Output** $best[1] \ldots best[n]$, the list of the $n$ best association rules.

**Algorithm Rule Gen(x)** (find the best rules with body $x$ efficiently)

10. **Let** $y$ be the smallest number such that $E(c|y/s(x), s(x)) > E(c(best[n])| \hat{c}(best[n]), s(best[n]))$.
11. **For** $j = 1 \ldots |x|$ (number of items not in $x$)
   (a) **If** $j = 1$ **Then** let $Y_1 = \{a_1 \ldots a_k\} \setminus x$.
   (b) **Else** let $Y_j = \{y \cup y' \mid y, y' \in Y_{j-1}, |y \cup y'| = j\}$ analogous to the generation of candidates in step 6a.
   (c) **For** all $y \in Y_j$ **Do**
   i. Measure the support $s(x \cup y)$. **If** $s(x \cup y) \leq y$, **Then** eliminate $y$ from $Y_j$ and **Continue** the for loop with the next $y$.
   ii. Calculate predictive accuracy $E(c ([x \Rightarrow y]) | s(x \cup y) / s(x), s(x))$ according to Equation 6.
   iii. **If** the predictive accuracy is among the $n$ best found so far (recorded in best), **Then** update best, remove rules in best that are subsumed by other, at least equally accurate rules (utilize Theorem 1 and test for $x \subseteq x' \quad y \quad y'$), and **Increase** $y$ to be the smallest number such that
\[ E(c \mid \gamma \mid s(x), s(x)) \geq E(c \mid \text{best}[n]) \mid \hat{c} \mid \text{best}[n]), s \mid \text{best}[n]) \].

12. **If** any subsumed rule has been erased in 11(c) iii, **Then** recur from step 10. The discretized data was supplied to this algorithm. Some interesting rules are shown below.

**Result**

When all 12 theoretical subjects were taken and predictive apriori algorithm was applied to result data it gave the following interesting association rules.

1. UGP101=C 10 ==> UGP303=D 10 acc:(0.936)
   i.e. Elementary algorithm = C ==> OOP = D

3. UGP202=F 9 ==> UGP402=F 9 acc:(0.92888)
   i.e. Operating system = F ==> Unix = F

4. UGP101=F UGP103=F 7 ==> UGP303=F UGP401=F 7 acc:(0.90952)
   i.e. Elementary algorithm = F, procedure oriented programming = F ==> OOP = F, UML = F

5. UGP101=F UGP303=F 7 ==> UGP401=F 7 acc:(0.90952)
   i.e. Elementary algorithm = F, OOP = F ==> UML = F

6. UGP103=B+ 4 ==> UGP303=C+ UGP401=C 4 acc:(0.85457)
   i.e. procedure oriented programming = B+ ==> OOP = C+, UML = C

7. UGP303=O 4 ==> UGP401=O 4 acc:(0.85457)
   i.e. OOP = O ==> UML = O

12. UGP301=F 10 ==> UGP401=F 9 acc:(0.8359)
   i.e. Software engineering = F ==> UML = F

18. UGP201=F UGP203=D 3 ==> UGP501=F 3 acc:(0.82095)

**Analysis**

Rule 3 – the above generated rules are giving warning for those students who are weak in operating system. Such students who got failed in operating system also have probability of getting failed in UNIX.

Rule 4 - If the student is not performing well in elementary algorithm than students result is consistently poor in procedural algorithm, object oriented programming and UML. This implies that in the beginning of the program student should be given very deep understanding of the elementary algorithm along with its practical application. Teachers should give more emphasis on understanding of languages so that an associated subject with language also automatically improves. Rule 1 and rule 5 is also included in Rule 4.

Rule 7 – 201 i.e. data structure is related to 502 i.e artificial intelligence. Student having good understanding of data structure can have sound understanding of AI also.

Rule 18 - data structure and data base management are related to software project management. This is confirmed by the rule 18.

**Conclusions**

For a student to successfully pass the MCA course, a sound knowledge of programming is must. The associated model thus generated will help that student in the future who are getting similar grades in the elementary algorithm, procedural oriented language and object oriented programming. The system will warn such students at a very early stage and corrective precautionary measures can be taken. Teachers can also pay special attention from very early. So our suggestion is that in the curriculum design two associated subjects should be placed in adjacent semesters. In fact algorithm, languages forms basis in the information technology, hence fundamentals must be made clear.

**Future work**

The association rules developed in this study will be helpful to predict the extent of learning by students in various courses and thereby their performance can be analyzed using result database. This work can be further analyzed using different data mining technique.
References:


Varun Kumar and Anupama Chadha, Mining Association Rules in Student’s Assessment Data, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 3, September 2012 ISSN (Online): 1694-0814


