A Student Advising System Using Association Rule Mining

Raed Shatnawi, Jordan University of Science and Technology, Jordan
Qutaibah Althebyan, Jordan University of Science and Technology, Jordan
Baraq Ghaleb, Edinburgh Napier University, UK
Mohammed Al-Maolegi, Community College, Yemen

ABSTRACT

Academic advising is a time-consuming activity that takes a considerable effort in guiding students to improve student performance. Traditional advising systems depend greatly on the effort of the advisor to find the best selection of courses to improve student performance in the next semester. There is a need to know the associations and patterns among course registration. Finding associations among courses can guide and direct students in selecting the appropriate courses that leads to performance improvement. In this paper, the authors propose to use association rule mining to help both students and advisors in selecting and prioritizing courses. Association rules find dependences among courses that help students in selecting courses based on their performance in previous courses. The association rule mining is conducted on thousands of student records to find associations between courses that have been registered by students in many previous semesters. The system has successfully generated a list of association rules that guide a particular student to select courses. The system was validated on the registration of 100 students, and the precision and recall showed acceptable prediction of courses.

KEYWORDS

Academic Advising, Association Mining, Smart Systems

1. INTRODUCTION

Building smart software systems are emerging for medium to large scale systems and in particular when large amount of data need processing. Data processing in many systems needs manual processing by supervisors and administrators of the system. For example, large universities need systems to efficiently process the advising of a large number of students. The manual building of students' schedules requires a large effort from advisors. The process of determining the courses, students should take in next semester, should be finished within a constrained time. Such process is inefficient when dealing with large number of students. Moreover, advisors usually use a prescribed and general schedule for the students while not considering individual differences among students. In addition, academic advising has been overlooked even though being central for the learning process and the performance of students (Gutiérrez et al., 2019). Most educational organizations provide simple software systems (web interfaces) to support academic advising.

DOI: 10.4018/IJWLTT.20210501.oa5

Academic advising is required widely in educational systems to provide direction and assistance to students (Iatrellis et al., 2017). In addition, students are usually allowed to take any courses either from the same major or others (Chang et al., 2016). This kind of choice results in difficulties in selecting the appropriate courses and distracts students from realizing their academic goals. Therefore, many research publications have proposed a dashboard of advising tools to provide access to student achievements and registration (Aguilar et al., 2014). A dashboard can show students' activities on an electronic management system and provides comparisons of their grades (Fritz, 2011). However, these systems do not provide help in decision making for both the advisor and the student.

There is a need to have a smart system that can automate the decision making to help students in their registration process. The main focus of this research is to build a smart academic advising system that optimizes the registration process and reduces the efforts of the university administration, advisors and students. The current advising system is either a traditional computer-based or paper-based. In addition, such systems consume a considerable amount of time and effort from the department administration and advisors (Biletskiy et al., 2009). Moreover, the process is inaccurate because the available information to the advisor is not always sufficient. The course-load is a main concern for students and it needs to be considered carefully. Several factors may affect the course-load namely, the nature of the subject, student preferences, and other courses taken by the student in the same term. These factors are not considered in the current advising system, therefore, the need for automated, and smart academic advising system is a must. The proposed system will serve the following objectives:

- Helping students register for classes according to their performances. Discovering course registration patterns can help in selecting courses that are supposed to improve students' performance.
- Drawing associations and dependencies between courses. Finding interdependent courses that were mostly registered together helps the advising process.
- Provide association rules that can help both the advisor and the student in selecting appropriate courses.

To meet these objectives, association rule mining is proposed to find and rank a list of suggested courses that a particular student can consider before registering. Association rules are generated from the history of registration. These rules can direct students to select future courses. The rule X→Y means that a student who has already taken X is most likely to register Y in next semesters. Students usually consult their advisors and preceding students before selecting courses for the next semester. One of the most frequently used data mining techniques in many applications is the association rule mining. It has been used as an underlying technology to improve the decision making process. In this study, the Apriori algorithm is used as a method to build academic advising system.

The rest of the paper is organized as follows. In section 2, the authors discuss related works on academic advising. In section 3, the association rule mining is described. Section 4 discusses the research methodology and the data collection. Section 5 provides a results discussion of the proposed advising system and shows a validation of the association rule mining on 100 students selected from a real data. Section 6 discusses the system limitations. Finally, the authors conclude the work in Section 7.

2. BACKGROUND AND RELATED WORK

An advising system in the university under study provides a web-based interfaces to help both advisors and students to improve the achievements of students. Advisors can access student records including the currently registered courses, and an unofficial transcript of student achievements including grades of courses, semester GPA and cumulative GPA. The system shows different web views (user interfaces) such as currently registered courses, suggested courses by advisor, student card that shows the study plan and highlights the completed courses. The department head distributes students among advisors including the students under probation (CGPA is less than 50). The students under probation cannot register the next semester without the help of advisors. The advisor should fill a proposed schedule for the next semester to help students improving their CGPA. Each advisor should spend a considerable time choosing courses based on his own experience and instinct. This process is lengthy and inaccurate that should be completed at the beginning of each semester. Although the system allows the advisor filling these schedules early on, students were not happy most of the time as they feel more comfortable for other courses. In addition, the advisor has a difficulty in choosing the number of credit hours for the next semester. Students have to communicate many times with advisors to select the appropriate courses. The proposed schedule for students do not take in consideration the associations between courses based on history of student's registration of similar records. The authors propose to update the schedule building by adding the associations rules that shows associations among courses based on students' history. To find associations among course, association rule mining should be use to find the patterns of registration in previous semesters.

Building an advisory system for students is one of the applications that can employ data mining methods. In a student registration system, students should choose various number of courses every time they want to register for a course which makes it hard for a student to choose a specific course. Advisors are often overloaded with too many students and they do not have enough time to help them. Moreover, students might not be satisfied with the preferences provided by advisors. Thus, it is necessary to build a recommendation system to help students in making decisions about what courses should be registered. Although this system is needed and vital to the process of registration, literature reveals that it is difficult to build a global or a standard advising system that works for all universities and institutions because of differences in regulations and rules among different institutions (Feghali et al., 2011; Nambiar & Dutta, 2010; Albalooshi & Shatnawi, 2010).

Many researchers have tried to build advising systems to facilitate the process of registration for students. There are two paradigms for building advising systems: the first paradigm relies solely on the advisor and the second paradigm relies on both student and advisor. The former is not interactive where the student relies on the advisor to suggest him/her courses to register with little or no flexibility (Crookston, 1972). The second paradigm is more interactive and systems allow the student selecting course based on preferences and the advisor provides recommendations based on these preferences (Chando, 1997). This type is more common and usually provides better outcomes compared to the first type.

The authors in (Hsu et al., 2002) developed a Web-based advising system for computer science and computer engineering undergraduate students where the system takes students' preferences as input and generate recommendations based upon these preferences. Students then get recommendations from advisors via a web browser. This facilitates the process of advising and makes the process faster and more reliable. However, this system is not an automated system and needs students to provide their preferences in order to find recommendations.

The student involvement is a shortcoming that limits the strength of an advising system. Siegfried et al. (2003) developed an academic advising system written using an expert system development tool. The tool targets freshman students in selecting appropriate courses to increase the pass rate among freshmen. The system depends on advisors' initial selection of math and composition courses. In another attempt to build a smart advising system, the authors of (Ahmar, 2011) have developed and designed a prototype for a rule-based expert advising system with an object oriented database. The authors built their system on two categories; student preferences and academic rules. The system takes as input all student preferences and then validates these preferences against system rules. Although this system is an advancement towards building a smart system by integrating student preferences with a process of validating these preferences (which refines those preferences), the authors failed to utilize their expert system to build an automated system that automatically generates recommendations based on noticed patterns among previously registered courses taken by previous students. Musso et al. (2013) have proposed to improve student academic performance (GPA) using artificial neural networks and discriminant analysis. The authors found that neural networks can achieve higher accuracy rate than discriminant analysis on independent variables such as orientation, alertness and executive attention. Nagy et al. (2013) proposed classification and clustering techniques to help first-year students following the appropriate track in a department.

Student history was employed by Mostafa et al. (2014) to recommend suitable major. Case-Based Reasoning (CBR) was used to build an advising system that creates recommendations of similar major based on history that were previously stored in a database. The system depends on the availability of similar cases in the knowledge base. The authors also provided forward and backward chaining procedures in order to find the most appropriate advice for a particular student. Asif et al. (2017) have used data mining techniques to identify courses that improve the performance of students both at early stage and at the end of the degree. The authors have also used the courses to find any relationships with student progression. The researchers have focused on identifying indicator courses rather than building a schedule of courses for the next semester. However, these courses were not equally informative for making decisions. Xu et al. (2017) have proposed to predict the future performance of students using information gathered from student history. The authors have aimed to predict the final cumulative GPA of the core courses in a certain subject.

Although the proposed methodologies help in advising students, these methodologies are not smart and adaptive to new data. Previous researches do not predict future semesters based on both the student preferences and performances in current and previous semesters. The proposed methodology on the other hand can be adapted for other universities given they have similar courses in the study plan. The study plans of the computer science and engineering programs have common courses among different universities. Therefore, finding course associations can be used for other similar programs. The proposed methodology provides rules for every course and provide the association among all courses in four years. The students and advisors can use these rules directly without the need to run the classifiers as previous works have proposed.

3. ASSOCIATION RULE MINING

Association rules were introduced in (Agrawal et al., 1993) in order to discover the hidden patterns in transactions of market data, where each transaction consists of a set of items. Association rules mining is applied in many fields such as bioinformatics, network traffic analysis, and cross-marketing. Association rule mining focuses on extracting the frequent patterns from data item sets. Rules are extracted from generated frequent item sets using two measures of interest, support and confidence. There are two well-known algorithms for the extraction of frequent item sets, Apriori (Agrawal & Srikant, 1994) and FP-growth algorithms (Han et al., 2000).

The association rule mining is a well-investigated method for discovering interesting correlations and relations among items in large datasets (Agrawal & Srikant 1994; Agrawal et al. 1993). The association rule mining is defined as: Let $L = \{i_1, i_2 ... i_n\}$ be a set of items, which are the set of courses (i.e., course call numbers). Let D be a set of transactions and each transaction T is a set of items such that $T \subseteq L$. The transaction in the registration system is composed of the registered courses in a particular semester, e.g., Fall or Spring. Each transaction is assigned a unique identifier (TID).

A transaction T contains X, a set of some items in L, if $X \subseteq T$. An association rule is an implication of the form $X \rightarrow Y$, where $X \subseteq L$, $Y \subseteq L$, and $X \subseteq Y = \phi$. A rule such as $X \rightarrow Y$ is assessed using two measures, support and confidence. The rule $X \rightarrow Y$ has a support S in the transaction set D if a particular percentage of the transactions in D contain $X \cup Y$. The rule $X \rightarrow Y$ holds in the transaction set D with confidence C if a percentage of the transactions in D that contain X also contain Y. The rules that have a support and confidence greater than the user-specified minimum support and minimum confidence are called interesting rules (Fjällström, 2016).

Frequent itemsets are the ones that have a support greater than the minimum support. A candidate itemset is the one that is expected to be frequent. The support for a rule is calculated as illustrated in equation 1.

$$\text{Sup}\ (\text{X}\rightarrow\text{Y}) = \sigma(X\cup Y)/|\text{D}| \tag{1}$$

σ(X∪Y) is the number of transactions that contain X and Y itemsets, |D| is the total number of all transactions. A rule is strong if the minimum support is high. Large values of minimum support produce small number of rules.

The confidence of X→Y is the probability occurrence of item X and Y together over the occurrences of items in X, which calculated by equation 2.

$$\text{Conf}\ (\text{X}\rightarrow\text{Y}) = Sup\ (\text{X}\rightarrow\text{Y})/Sup(X) \tag{2}$$

The rules generation process has two steps: (i) finding all frequent itemsets that have support above a predefined minimum support value; (ii) generating the association rules by using the frequent itemsets. Many algorithms have been proposed to generate association rules. In this study, the authors use the Apriori algorithm in order to generate the association rules. To compute the support of the itemsets, the Apriori algorithm uses the breadth first search technique. In this research, the authors propose to use an efficient Apriori association mining algorithm. To generate the interesting rules, the algorithm makes multiple passes over the database (Liu et al., 1998). It uses the frequent itemsets in previous pass to generate candidate itemsets for the next pass. At pass k, it *g*enerates the candidate k-item*s*ets from frequent k-1-it*e*msets and then scans the database to find the support of each candidate k-item*s*et. Itemsets with support above the minimum support constitute the frequent k-item*s*ets while other itemsets are discarded. The process continues until no more candidate itemsets could be generated. Generating candidate itemsets in the Apriori algorithm consists of two steps; join step and prune step. In join step, the algorithm joins different k-1-it*e*msets to generate k-item*s*et if they share the first k-2 it*e*ms. In prune step, the candidate k-item*s*ets that are generated in the join step and have infrequent k-1-su*b*set of k-1-it*e*msets, are removed. The prune step exploits the Apriori property that guarantees that all non-empty subsets of frequent itemset must also be frequent.

4. RESEARCH METHODOLOGY

In this section, the authors discuss how the Smart Academic Advisory System (SAAS) is implemented using association rule mining. Association rule mining is considered as one of the most important data mining techniques. Its ultimate goal is to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transactional databases or other data repositories. The research methodology includes collecting real data for m history of the registration system. The data are kept in a structured format and stored at the datacenter of the university. The data are then transformed to transactions that are suitable for mining. The itemsets of each transaction are prepared as a list of courses passed in a particular semester. The data are then feed to the system and association rules are generated.

4.1 Data Collection

The authors have selected the computer information technology faculty at Jordan University of Science and Technology as a subject of the system. The data are stored in a central database that can be accessed directly only by authorized personal in the data center. The system uses a real data from the registration pool. Four-year programs were selected for this study. The registration of students (total 4 years) were exported from the registration database. The university data center has provided the registration of students in an Excel sheet format. The dataset consists of 1530 student records of many semesters. The number of registered courses by these students in the examined period was 39510 before preprocessing. After filtering out the courses that were not pass such as failed, exempted or transferred, the number of registered courses was 27776. Data were collected for the first and second semesters of each student. The student cards of each semester include the student's ID, student's name, semester, course call number, course title, section number, mark (from 35 to 100) and grade (pass, fail, others).

4.2 Data Preprocessing

Student data is subjected to the knowledge mining processes to generate knowledge from it. The mining process include data cleaning, data integration, data selection, data transformation, In Data Cleaning the irrelevant data are removed from the collected data which includes transferred semesters and courses. In data Integration step, the regeneration of two major departments are included in the study. The sources are combined into one dataset. In data selection, irrelevant instances are removed from the data sets such as student failures. The selected data is transformed to the appropriate form for the mining procedure.

Before using the Apriori algorithm to discover patterns among courses, the registration data should be transformed into a format that is more readable by association rule mining. For each student in in each major (target student major), the system will create one transaction for each semester that the student had been registered in. The transaction consists of the courses taken by the specified student given that the course score is equal to or greater than 50. For example, suppose that two students with IDs 1 and 2 are the set of students who are in the same department of the target student and they take the courses as depicted in Table 1. These data should be transformed into transactions as depicted in Table 2. A transaction is composed of an item set that includes the courses registered together in one semester and the student has passed all these courses.

4.3 Association Rule Mining Generation

In this work, associative classification is constructed in two stages as suggested in (Sun et al., 2009). The students' registration data are transformed into a transactional dataset by generating a transaction for each student that includes the student ID and a list of registered courses in a particular semester. For example, if a student is registered in four semesters then four transactions are generated. These transactions are then fed to the Apriori algorithm to generate the association rules. First, the system identifies all the associations that have a significant frequency of occurrences according to minimum confidence and support. Second, classification rules are generated to build a classifier. Rules generation is conducted as shown in the pseudo-code in Figure 1.

Many rules are expected and therefore there should be filtering and ordering by importance. Rules may not be appropriate for each student, for example, students have not registered the left hand side of a rule. Such rules are not reported for the student under supervision. Some rules may include already registered courses on the right hand side, then the rule is trimmed and the rest of courses are left in the right hand side of the rule. After applying these filters, the rest of the rules are then sorted by the confidence level for the student currently under supervision. The system extracts the right-hand side of all the rules, i.e., course ID. These course are reported to the advisor as an ordered list of suggested courses for the student to register in next semester.

To validate the methodology of the work, 100 student's records are selected from the registration records. Each student has registered two semesters at least. The advisory system is used to predict the registration schedule for each student in a semester and the results are compared with the actual registration. The results of each student are evaluated by selecting courses using the system and comparing the results with the actual registration. The results are then evaluated using the precision and recall measures. To find the precision and recall for each student, first the system should provide

Table 1. Students registration

Student ID	The semester	Course Name	Grade
1	First-Semester-2010-2011	A	50
1	First-Semester-2010-2011	B	78
1	First-Semester-2010-2011	C	40
1	Second-Semester-2010-2011	D	67
1	Second-Semester-2010-2011	E	30
1	Second-Semester-2010-2011	F	50
1	Summer-Semester-2010-2011	G	40
1	Summer-Semester-2010-2011	H	90
2	First-Semester-2010-2011	A	90
2	First-Semester-2010-2011	C	78
2	First-Semester-2010-2011	D	56
2	First-Semester-2010-2011	F	84
2	Second-Semester-2010-2011	G	76
2	Second-Semester-2010-2011	H	54
2	Second-Semester-2010-2011	B	94
2	Summer-Semester-2010-2011	L	87
2	Summer-Semester-2010-2011	Z	67

Table 2. Student transactions

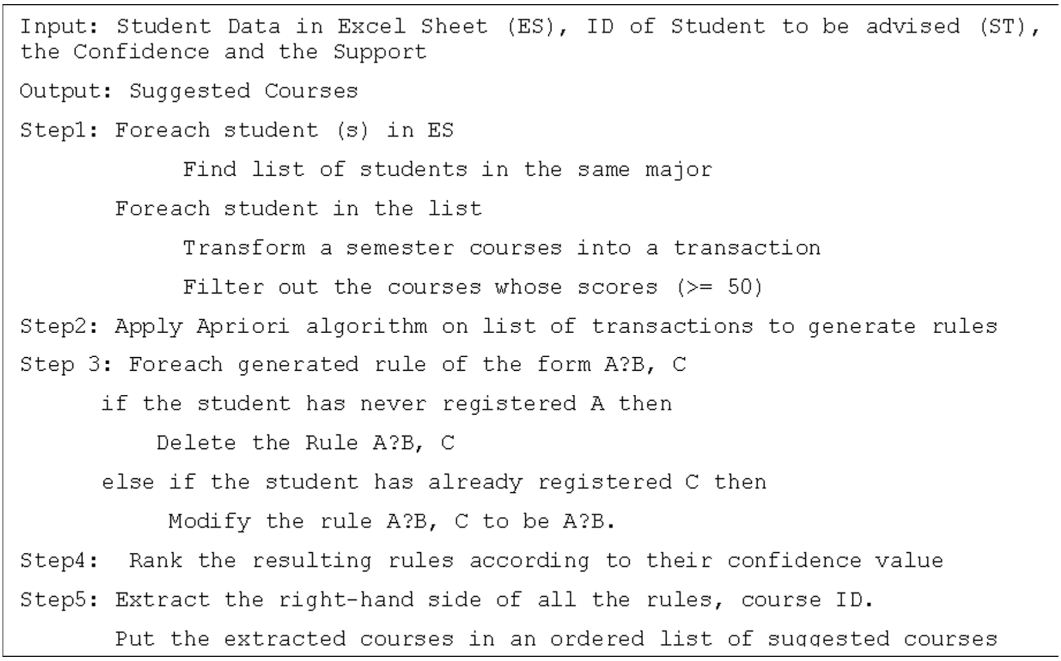
Transaction ID	Transactions
1	A,B
2	D,F
3	H
4	A,C,D,F
5	G,H,B
6	L,Z

the number of predicted (suggested) courses and the number of actually registered courses. The precision and recall are calculated as follows for each student.

Precision = (predicted and registered) / registered
Recall = (predicted and registered) / predicted

The predicted courses are the courses that results from the system. The predicted courses alone are not enough to measure the performance of the model. The precision of the model is calculated

Figure 1.



```
Input: Student Data in Excel Sheet (ES), ID of Student to be advised (ST),
the Confidence and the Support
Output: Suggested Courses
Step1: Foreach student (s) in ES
            Find list of students in the same major
       Foreach student in the list
            Transform a semester courses into a transaction
            Filter out the courses whose scores (>= 50)
Step2: Apply Apriori algorithm on list of transactions to generate rules
Step 3: Foreach generated rule of the form A?B, C
      if the student has never registered A then
          Delete the Rule A?B, C
      else if the student has already registered C then
           Modify the rule A?B, C to be A?B.
Step4:  Rank the resulting rules according to their confidence value
Step5: Extract the right-hand side of all the rules, course ID.
       Put the extracted courses in an ordered list of suggested courses
```

through finding the ratio of the predicted courses that were actually registered. A value of precision that is less than one means that there are some predicted courses out of the actually registered courses. Recall on the other hand measures the ratio of the predicted and registered courses out of the total predicted courses. A recall value that is less than one means that there are many predicted courses that are not registered by the student. Therefore, both the precision and recall are reported in this study.

5. RESULTS ANALYSIS

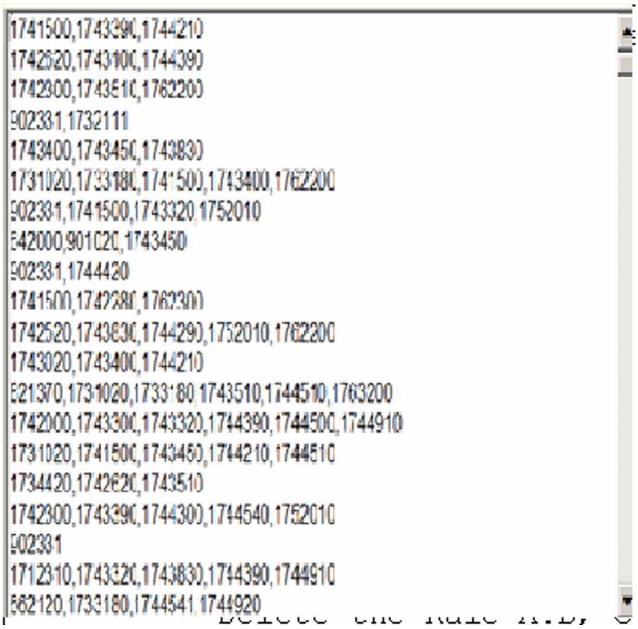
This section includes two parts: first, running the mining process to build the advisory system; second, running an evaluation of the system on 100 students to validate the use of the system in practice.

5.1 Building the Advisory System

After creating the transactional database (transactions), the data become ready for mining using the Apriori algorithm and the system can generate the rules that the target user can use to get recommendations about the courses to register. The number of the rules that should be generated depends on the support and the confidence as provided by the supervisors or the target student. Figure 2 shows a snapshot of the generated transactions after running the system on real data while Figure 3 depicts a snapshot of the generated rules after applying the Apriori algorithm on the transactions. Some of the generated rules in Figure 3 are not valid. These rules have a course in the right-hand side that has not been taken by the target user or the rules that have a course in the left-hand side that has been taken already by the target user. Thus, these rules should be filtered out. For example, look at the following rule:

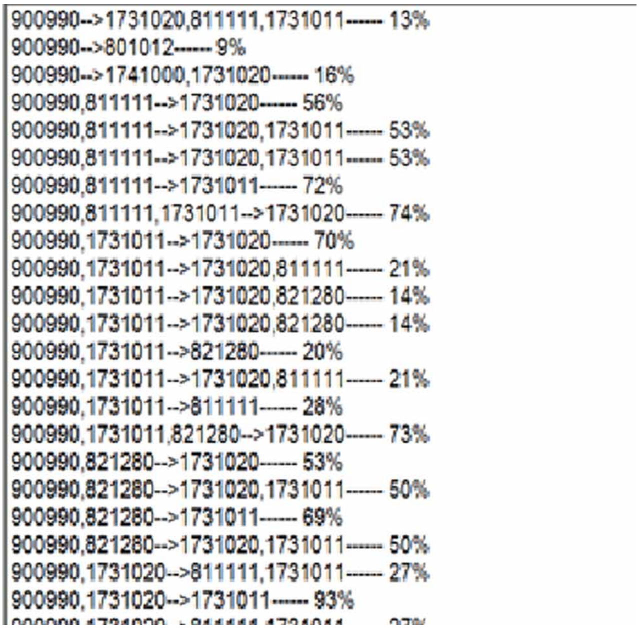
9900990, 1731020→1731011

Figure 2.



```
1741500,1743390,1744210
1742620,1743100,1744390
1742300,1743510,1762200
902331,1732111
1743400,1743450,1743830
1731020,1733180,1741500,1743400,1762200
902331,1741500,1743320,1752010
542000,901020,1743450
902331,1744420
1741500,1742280,1762300
1742520,1743830,1744290,1752010,1762200
1743020,1743400,1744210
521370,1731020,1733180,1743510,1744510,1763200
1742000,1743300,1743320,1744390,1744500,1744910
1731020,1741500,1743450,1744210,1744510
1734420,1742620,1743510
1742300,1743390,1744300,1744540,1752010
902331
1712310,1743320,1743830,1744390,1744910
562120,1733180,1744541,1744920
```

Figure 3.



```
900990-->1731020,811111,1731011------ 13%
900990-->801012------ 9%
900990-->1741000,1731020------ 16%
900990,811111-->1731020------ 56%
900990,811111-->1731020,1731011------ 53%
900990,811111-->1731020,1731011------ 53%
900990,811111-->1731011------ 72%
900990,811111,1731011-->1731020------ 74%
900990,1731011-->1731020------ 70%
900990,1731011-->1731020,811111------ 21%
900990,1731011-->1731020,821280------ 14%
900990,1731011-->1731020,821280------ 14%
900990,1731011-->821280------ 20%
900990,1731011-->1731020,811111------ 21%
900990,1731011-->811111------ 28%
900990,1731011,821280-->1731020------ 73%
900990,821280-->1731020------ 53%
900990,821280-->1731020,1731011------ 50%
900990,821280-->1731011------ 69%
900990,821280-->1731020,1731011------ 50%
900990,1731020-->811111,1731011------ 27%
900990,1731020-->1731011------ 93%
```

The course (9900990) has not been taken by the target student. The condition of this rule is incorrect and consequently the rule is incorrect and it should be removed from the set of the generated rules. Other types of the rules can be filtered out of the form:

1732111→1743450, 9900990

The course (1743450) was already taken by the target student. Thus, this rule should only be modified in order to be correct by removing the course (1743450) from the left-hand side of the rule. The modified rule will look as: 1732111→ 9900990. A snapshot of the generated rules after filtering are presented in Figure 4. The results in Figure 4 show the associations between courses and can help both the advisor and the student in drawing conclusions about related courses.

Finally, the system is used to extract a list of suggested courses. These courses are sorted according to the confidence of the rules. For example, Figure 5 depicts a snapshot of the final suggested courses sorted by their confidence. In Figure 5, the system recommends the student to select the courses that have the highest confidences. This list gives priorities to courses and helps the student in selecting a short list of courses out of many courses that are offered during the registration period. For example, a student can select the first five courses in the list or can select course based on the confidence level.

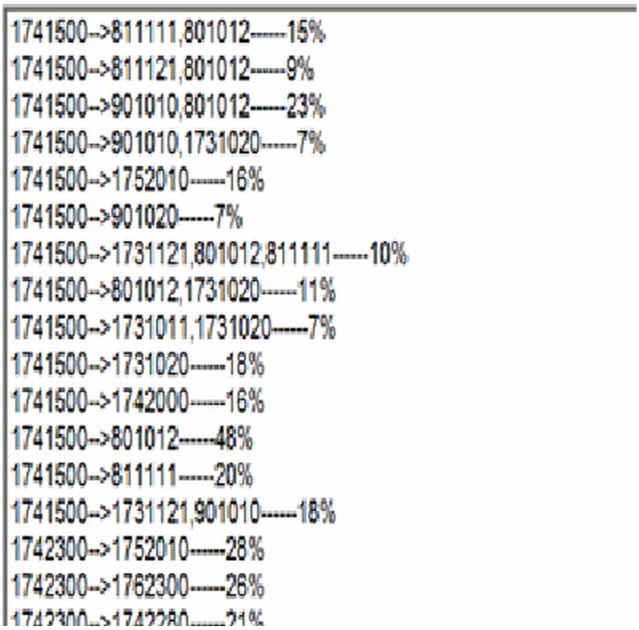
5.2 System Results Validation

For 100 students, the boxplot of precision and recall measures are shown in Figure 6. On average the precision values are larger than the recall. The median of both is more than 0.50, which is considered high and acceptable for an advisory system. The advisory system can be used for most students with high precision and recall. Students have selected courses that also predicted by the advisory system. Although the system has suggested courses that were not selected by students, these courses are from the study plan and does not have detrimental effects on students' performance. The selected courses are believed to improve students' performance measured as CGPA.

6. LIMITATIONS OF THE ADVISORY SYSTEM

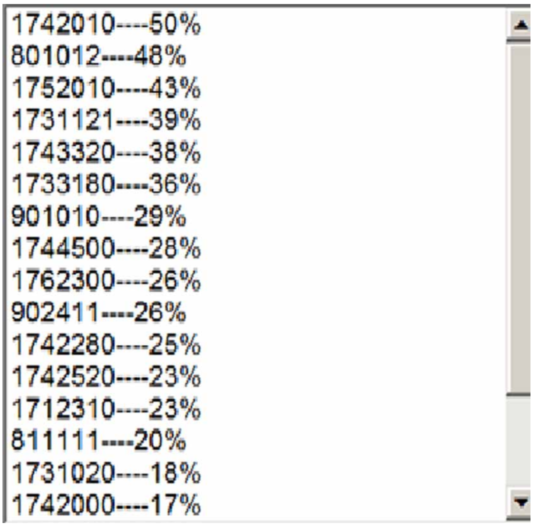
The advisory system has used the data of students in one faculty, the computer and information technology faculty. The faculty has three separate programs with different study plans. As part

Figure 4.



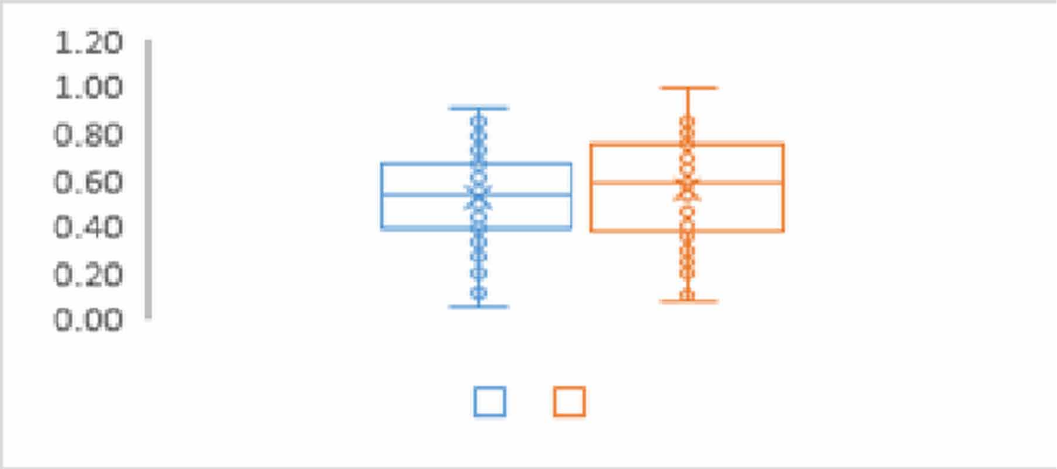
```
1741500-->811111,801012------15%
1741500-->811121,801012------9%
1741500-->901010,801012------23%
1741500-->901010,1731020------7%
1741500-->1752010------16%
1741500-->901020------7%
1741500-->1731121,801012,811111------10%
1741500-->801012,1731020------11%
1741500-->1731011,1731020------7%
1741500-->1731020------18%
1741500-->1742000------16%
1741500-->801012------48%
1741500-->811111------20%
1741500-->1731121,901010------18%
1742300-->1752010------28%
1742300-->1762300------26%
1742300-->1742280------21%
```

Figure 5.



```
1742010----50%
801012----48%
1752010----43%
1731121----39%
1743320----38%
1733180----36%
901010----29%
1744500----28%
1762300----26%
902411----26%
1742280----25%
1742520----23%
1712310----23%
811111----20%
1731020----18%
1742000----17%
```

Figure 6.



of the study plan, there are many courses managed by the university or the faculty and that are common for students in the three departments. The study has considered data for four years only for each student. The time required for students' graduation is four years in the three programs. However, the data was large enough to conclude the results. The validation of the model was performed on advising 100 students only, which is considered statistically enough for conclusions. All student records are coming from one university, which makes the generalization of the model difficult to other universities. In addition, the change of study plans requires rerunning the advisory system to generate the rules for the new study plans. The course content was not considered because it is fixed at least for four years.

7. CONCLUSION

In this research, the authors have proposed a new smart advising system to help undergraduate students during the registration period. Using a data mining technique, a smart academic advising system is implemented to help undergraduate students to register in the next semester. The system uses association rule mining to study the interdependences among courses. The association rule mining provides rules that help both students and advisors selecting the courses that were taken together by previous students and the performance of students were acceptable. The system uses a real data from the registration pool of Jordan University of Science and Technology. Although the advising process is not automated, the proposed system can reduce the amount of efforts that the advisors should spend advising a large number of students. In addition, students can select courses based on association rules among courses which is similar to what they are used to do in real life.

Course content is fixed by the department for each course. Including the course content to each transaction shows repetitiveness and will not help the association mining. However, course contents will be added to the mining process in future works. Other machine learning techniques can be employed to build advising systems.

7.1 Availability of Data and Material

The data sets under study belongs to Jordan University of Science and Technology. The data was exported to an excel sheet for the appropriate use by the authors.

7.2 Funding

This project is supported in part by an IBM corporation award and Jordan University of Science and Technology (Grant# 271/2011).

REFERENCES

Feghali, T., Zbib, I., & Hallal, S. (2011). A web-based decision support tool for academic advising. *Journal of Educational Technology & Society*, *14*(1), 82–94.

Nambiar, A. N., & Dutta, A. K. (2010). Expert system for student advising using JESS. *International Conference on Educational and Information Technology (ICEIT)*. doi:10.1109/ICEIT.2010.5607701

Albalooshi, F., & Shatnawi, S. (2010). HE-Advisor: A multidisciplinary web-based higher education advisory system. *Global Journal of Computer Science & Technology*, *10*(7), 37–49.

Crookston, B. (1972). A developmental view of academic advising as teaching. *Journal of College Student Personnel*, *13*(1), 12–17.

Chando, C. M. (1997). *Predicting advising style preference from student characteristics* (Doctoral dissertation). University of Memphis.

Hsu, S., Marques, O., Ilyas, M., & Ding, X. (2002). Web-Based Undergraduate Academic Advising System. *International Conference on Engineering Education*.

Ahmar, M. (2011). A Prototype Student Advising Expert System Supported with an Object-Oriented Database. *International Journal of Advanced Computer Science and Applications*.

Witten, I., Frank, E., & Hall, M. (2011). *Data Mining: Practical machine learning tools and techniques* (3rd ed.). Academic Press.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. *20th International Conference on Very Large Data Bases*, 478-499.

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining associations between sets of items in large databases. *Proc. of the ACM-SIGMOD 1993 Int'l Conference on Management of Data*.

Liu, B., Hsu, W., & Ma, Y. (1998). Integrating classification and association rule mining. *Fourth International Conference on Knowledge Discovery and Data Mining*, 80-86.

Sun, Y., Wong, A., & Kamel, M. (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, *23*(4), 687–719. doi:10.1142/S0218001409007326

Mostafa, L., Oately, G., Khalifa, N., & Rabie, W. (2014). A case based reasoning system for academic advising in egyptian educational institutions. *2nd International Conference on Research in Science, Engineering and Technology (ICRSET'2014)*.

Gutiérrez, F., Seipp, K., Ochoa, X., Chiluiza, K., De Laet, T., & Verbert, K. (2019). LADA: A learning analytics dashboard for academic advising. *Computers in Human Behavior*.

Fritz, J. (2011). Classroom walls that talk: Using online course activity data of successful students to raise self-awareness of underperforming peers. *The Internet and Higher Education*, *14*(2), 89–97. doi:10.1016/j.iheduc.2010.07.007

Aguilar, S., Lonn, S., & Teasley, S. D. (2014). Perceptions and use of an early warning system during a higher education transition program. In *Proceedings of the Fourth International Conference on Learning Analytics and Knowledge - LAK '14*. ACM Press. doi:10.1145/2567574.2567625

Biletskiy, Y., Brown, J. A., & Ranganathan, G. (2009). Information extraction from syllabi for academic e-Advising. *Expert Systems with Applications*, *36*(3), 4508–4516. doi:10.1016/j.eswa.2008.05.011

Asif, R., Merceron, A., Ali, S. A., & Haider, N. G. (2017). Analyzing undergraduate students' performance using educational data mining. *Comput. Edu.*, *113*, 177–194. doi:10.1016/j.compedu.2017.05.007

Xu, J., Moon, K. H., & van der Schaar, M. (2017). A machine learning approach for tracking and predicting student performance in degree programs. *IEEE Journal of Selected Topics in Signal Processing*, *11*(5), 742–753. doi:10.1109/JSTSP.2017.2692560

Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. *Proceedings of the 2000 ACM SIGMOD international conference on Management of data (SIGMOD '00)*, 1-12.

Fjällström, P. (2016). *A way to compare measures in association rule mining*. Available from: http://urn.kb.se/resolve?urn=urn:nbn:se:umu:diva-124903

Siegfried, R. M., Wittenstein, A. M., & Sharma, T. (2003). An automated advising system for course selection and scheduling. *Journal of Computing Sciences in Colleges*, *18*(3), 17–25.

Musso, M. F., Kyndt, E., Cascallar, E. C., & Dochy, F. (2013). Predicting general academic performance and identifying the differential contribution of participating variables using artificial neural networks. *Frontline Learning Research*, *1*(1), 42–71. doi:10.14786/flr.v1i1.13

Nagy, H. M., Aly, W. M., & Hegazy, O. F. (2013). An educational data mining system for advising higher education students. World Academy of Science. *Engineering and Technology International Journal of Information Engineering*, *7*(10), 175–179.

Iatrellis, O., Kameas, A., & Fitsilis, P. (2017). Academic advising systems: A systematic literature review of empirical evidence. *Education in Science*, *7*(4), 90. doi:10.3390/educsci7040090

Chang, P. C., Lin, C. H., & Chen, M. H. (2016). A hybrid course recommendation system by integrating collaborative filtering and artificial immune systems. *Algorithms*, *9*(3), 47. doi:10.3390/a9030047

Raed Shatnawi received the MSc degree in software engineering in 2004 and the PhD degree in computer science from the University of Alabama in Huntsville. He is currently an associate professor in the Department of Software Engineering at Jordan University of Science and Technology. He has published many papers in high ranked journals and conferences (IEEE, ScienceDirect and Interscience). He has reviewed papers for the Journal of Systems and Software, Empirical Software Engineering, and Information Sciences, and many international conferences. His main interests are in software metrics, software refactoring, software maintenance, and open-source systems development.

Qutaibah Althebyan has got Ph.D. form University of Arkansas. Dr. Althebyan has published many research papers in the fields of software engineering, cloud computing and security.

Baraq Ghaleb is currently a lecturer at Edinburgh Napier University where he is completing PhD. Ghaleb completed hos M.Sc. in computer science from Joran University of science and technology.

Mohammed Al-Maolegi is currently a lecturer in Community College, Amran, Yemen. Al-Maolegi has completed is MSc from Jordan University of Science and Technology.