

# Sequential patterns analysis in a student database

R. Campagni, D. Merlini, and R. Sprugnoli

Dipartimento di Sistemi e Informatica  
Viale Morgagni 65, 50134, Firenze, Italia  
`[renza.campagni,donatella.merlini,renzo.sprugnoli]@unifi.it`

**Abstract.** This paper presents a data mining methodology to analyze the careers of students, where a career can be seen as a sequence of exams. The model is built using sequential pattern analysis and uses the algorithm SPAM. We consider an *ideal career* corresponding to a student which has taken each examination just after the end of the corresponding course, without delays. The frequent patterns identified by the sequential pattern analysis are then compared with the career of the ideal student. The most interesting patterns are then used to refine the analysis by using clustering techniques. Finally, we apply this methodology to a real case study and interpret the results.

## 1 Introduction

Rapid advances in data collection and storage technology have enabled organizations to accumulate a vast amount of data, and often tools for traditional data analysis are not sufficiently efficient to produce significant results. Data mining is a data analysis methodology used to identify hidden patterns in a large data set. This technology blends traditional data analysis methods with sophisticated algorithms for processing large volumes of data; it has been successfully used in different areas including the educational environment. Educational data mining is an emerging and interesting research area that produces useful, previously unknown regularities from educational databases for better understanding and improving the educational performance and assessment of the student learning process (see [10] for a detailed description of the state of the art in this context). In fact, universities record data containing information about students which are usually only treated individually and for official purposes. However, if properly analyzed, these data could be used by the academic organizations to understand the behavior of students.

Over the years, several data mining models have been proposed and implemented to analyze the performance of students. For example, in [6,7] a model is proposed which presents the advantages of data mining technology in higher educational systems; the authors give a sort of road map to assist the institutions to identify the ways to improve their processes. In [4], the authors illustrate a classification model to investigate the profile of students which most likely

leave university without ending their career. In particular, they use some classification algorithms implemented in the WEKA system [14]. In [5] a framework is proposed for mining educational data using association rules and, in [8], in order to explore the factors having impact on the success of university students, a system based on the decision tree classification technique is presented. Data mining techniques have also been applied in computer-based and web-based educational systems (see, e.g., [11,9]). The existing literature about the use of data mining in educational systems is mainly concerned with techniques such as clustering, classification and association rules (see, e.g., [5,13,14]).

In this paper, we propose to use the *sequential pattern* mining methodology to study the career of a student. This technique has been introduced in [1] and has become an important method in data mining (see, e.g., [13]). Sequential pattern analysis aims to find relationships between occurrences of sequential events, that is, to find if any specific order of the occurrences exists. In particular, we consider as events the exams taken by a student; the temporal information is the *semester* in which the exam has been taken or the *delay* with which it has been taken. We study an organization of the university which allows students to take an exam in different sessions after the end of the course, as in Italy. Therefore, the career of a student can be seen as a sequence  $i_1 i_2 \dots i_N$  of  $N$  exams, where  $i_k$  precedes  $i_j$  in the sequence if  $i_k$  has been taken at the same time or before  $i_j$ . The temporal information allows us to see the career of a student as a sequence  $\langle e_1 e_2 \dots e_n \rangle$  where each element  $e_j$  is a collection of one or more exams taken in the same semester or having the same delay. By analyzing the sequential patterns, we can explain some behaviors which may seem counterintuitive, e.g., course  $x$  is scheduled before course  $y$  while many students take exam  $y$  before  $x$ . Such information may be helpful for changing course schedules or to find out those courses whose exams are considered as difficult by the students and thus could give insight for reorganizing the curricula. Moreover, sequential patterns can be used to refine the analysis of students by introducing in the database the Boolean information about the most significant patterns, according to the fact that a student verifies a pattern or not. This new information can be explored by using clustering techniques in order to understand if students satisfying the patterns have some common characteristics.

As far as we know, the idea of using sequential pattern mining in the context of education is new. We apply this methodology to the careers of the students attending the Computer Science degree at the University of Florence (Italy), by using the SPAM algorithm proposed in [2], an implementation of which can be found in [12], and by using the *K-means* implementation of WEKA for clustering. The data sets we study are not very large but allow us to illustrate a general methodology which can be applied to any kind of university degree, provided a preprocessing phase is performed on the original data to take into account temporal information.

In the following sections we describe the fundamental concepts and the main steps of our methodology. In Section 2, we introduce the sequential patterns technique and the algorithm SPAM; moreover, we explain how the student data

can be processed to do the pattern analysis and how new information can be inserted and used to perform clustering with the most interesting patterns. In Section 3, we illustrate our methodology on a real case study by analyzing the most interesting sequential patterns found by SPAM and the results of clustering. Finally, in Section 4 we give an interpretation of the results in terms of the student choices and the organization of the laurea degree.

## 2 The methodology

In order to study sequential patterns in a student database we consider the *career* of a student, that is, the way the student implements her or his exams over the degree-learning time: a student can take an exam immediately after a course (the *ideal* choice) or later. In particular the *ideal career* is defined as the sequence of  $N$  exams taken by an ideal student, who takes the exams just after the end of the corresponding course without delay. In this ideal sequence, the order of exams corresponding to courses given in the same semester is the one chosen by most students. By convention, we identify the ideal career by the identity permutation, that is, we identify the first exam in the sequence by code 1, the second by code 2, and so on. The career of a generic student is therefore a permutation of the integers 1 to  $N$  (see [3] for a recent use of the ideal career for analyzing a student database with a clustering technique).

sequence $s$	sequence $t$	Is $t$ a subsequence of $s$ ?
$\langle\{2, 4\} \{3, 5, 6\} \{8\}\rangle$	$\langle\{2\} \{3, 6\} \{8\}\rangle$	Yes
$\langle\{2, 4\} \{3, 5, 6\} \{8\}\rangle$	$\langle\{2\} \{8\}\rangle$	Yes
$\langle\{1, 2\} \{3, 4\}\rangle$	$\langle\{1\} \{2\}\rangle$	No
$\langle\{2, 4\} \{2, 4\} \{2, 5\}\rangle$	$\langle\{2\} \{4\}\rangle$	Yes

Table 1. Sequences and subsequences

### 2.1 Sequential pattern analysis

In this section we present the basic concepts of sequential patterns and the algorithm used for our analysis. A *sequence* is an ordered list of *elements* and can be denoted as

$$s = \langle e_1 e_2 e_3 \dots e_n \rangle,$$

where each element  $e_j$  is a collection of one or more *events*, i.e.,

$$e_j = \{i_1, i_2, \dots, i_{k_j}\}, \quad j = 1, \dots, n.$$

The events of an element correspond to the same temporal information, that is, occur at the same time. The length of a sequence corresponds to the number of elements in the sequence, while a  $k$ -sequence is a sequence that contains  $k$  events.

A sequence  $t$  is a *subsequence* of another sequence  $s$  if each ordered element in  $t$  is a subset of an ordered element in  $s$ . Formally, the sequence  $t = \langle t_1 t_2 \dots t_m \rangle$  is a subsequence of  $s = \langle s_1 s_2 \dots s_n \rangle$  if integers  $1 \leq j_1 \leq j_2 \leq \dots \leq j_m \leq n$  exist such that  $t_1 \subseteq s_{j_1}, t_2 \subseteq s_{j_2}, \dots, t_m \subseteq s_{j_m}$ . If  $t$  is a subsequence of  $s$ , then we say that  $t$  is *contained* in  $s$ . Table 1 gives examples illustrating the idea of subsequences for various sequences. Let  $D$  be a data set containing one or more *data sequences*. The *support* of a sequence  $s$  is the fraction of all data sequences that contain  $s$ . If the support for  $s$  is greater than or equal to a user-specified threshold `minsup`, then  $s$  is a *sequential pattern* or *frequent sequence*. Given a data set of sequences  $D$  and a user-specified minimum support threshold `minsup`, the task of *sequential pattern discovery* is to find all sequences with support  $\geq$  `minsup`.

Table 2 illustrates an example of a table in a student database. The table contains the fields **Student**, **Semester** and **Exams** which represent the student identifier, the temporal information and the list of identifiers for the exams. For example, student with identifier 10 has taken exams with identifier 3, 2 and 7 in the semester 1 and exam 1 in the semester 2. According to Table 2, the identifier

Student	Semester	Exams
10	1	3,2,7
10	2	1
10	3	4,5
⋮	⋮	⋮
20	1	3
20	2	7,2
20	4	4,1
⋮	⋮	⋮
30	2	4,7
30	3	1
⋮	⋮	⋮
40	2	2,1
40	3	3,5
⋮	⋮	⋮

**Table 2.** Database of sequences

of an exam can be seen as an event and a list of exams can be seen as an element. The sequence  $\langle \{3, 2, 7\} \{1\} \{4, 5\} \rangle$  is associated to student 10 and represents a portion of his (her) career. This sequence has length 3 and corresponds to a 6-sequence. Moreover, let  $D$  be given by

$$s_{10} = \langle \{3, 2, 7\} \{1\} \{4, 5\} \rangle$$

$$s_{20} = \langle \{3\} \{7, 2\} \{4, 1\} \rangle$$

$$s_{30} = \langle \{4, 7\} \{1\} \rangle$$

$$s_{40} = \langle \{2, 1\} \{3, 5\} \rangle$$

that is, the sequences of careers for students 10, 20, 30 and 40. The new sequence  $s = \langle \{1\} \{5\} \rangle$  has support 0.5 in  $D$  since  $s$  is a subsequence for  $s_{10}$  and  $s_{40}$ .

Finding sequential patterns in large databases is an important data mining problem. There are several algorithms implementing techniques for finding frequent sequences based on the *Apriori* principle [13]. An efficient algorithm integrating a variety of algorithmic contributions into a practical tool is **SPAM** (Sequential PAttern Mining), described in [2]. The algorithm is based on the construction of a lexicographic tree of sequences which is traversed in a depth-first manner and in which the root is conventionally labeled with 0. An additional feature of **SPAM** is its property of online outputting sequential patterns of different length (compare this to a breadth-first search strategy that first outputs all patterns of length one, then all patterns of length two, and so on). The implementation of **SPAM** we consider uses a vertical bitmap data layout allowing for simple and efficient counting.

Student	Semester	Exam
10	1	3
10	1	2
10	1	7
10	2	1
⋮	⋮	⋮
20	1	3
20	2	7
20	2	2
⋮	⋮	⋮

**Table 3.** Example of data input for **SPAM**

Table 4 shows the output obtained from the data of Table 2, in the format accepted by **SPAM** and illustrated in Table 3, by using **SPAM** with a support equal to 0.5. Each line of the output file is a frequent sequence and can be interpreted as follows. The last number is the frequency of the sequence; the data in Table 4 correspond to 4 students and we choose `minsup` equal to 0.5, so we obtained all the sequences that have frequency  $\geq 2$ . The number  $-1$  indicates a change of semester and the symbol  $-$  indicates the end of the sequence. For example, the first line of the output indicates that 4 students have taken exam 1; the eighth line indicates that 2 students have taken exams 2 and 7 in the same semester and then have taken exam 1 in a later semester.

The example illustrated in Tables 3 and 4 refers to the use of sequential pattern analysis where the temporal information corresponds to the *semester* in which the student takes exams. Another temporal information which can be

**Frequent sequences**

1 - 4  
1 -1 5 - 2  
2 - 3  
2 -1 1 - 2  
2 -1 4 - 2  
2 -1 5 - 2  
2 7 - 2  
2 7 -1 1 - 2  
2 7 -1 4 - 2  
3 - 3  
3 -1 1 - 2  
3 -1 4 - 2  
4 - 3  
5 - 2  
7 - 3  
7 -1 1 - 3  
7 -1 4 - 2

**Table 4.** Output of SPAM on data of Table 3

used is the *delay* with which a student takes exams. This delay is expressed in semesters and is the difference between the semester in which the student takes an exam and the semester in which the course has been given by the teacher. The frequent patterns obtained by using one or the other temporal information have a different meaning. With the semester, we obtain patterns which emphasize the order used by students to take their exams; with the delay, we obtain patterns grouping the exams according to the delay with which they are taken by students. These patterns can be used to understand which exams are more difficult to be taken by the students or, on the contrary, are taken without great difficulties. Considerations of this type can suggest to modify the schedule of the laurea degree or confirm that the laurea degree is well structured. We observe explicitly, that the use of one or the other temporal information changes the representation as a sequence of the ideal career introduced at the beginning of Section 2. More precisely, if  $N$  is the number of exams,  $S$  the number of semesters in which the laurea degree is organized and  $p_j$  the number of exams corresponding to the semester  $j$ , then the ideal career corresponds to the sequence

$$s_I^{[s]} = \langle e_1 \dots e_S \rangle, \quad e_j = \left\{ \sum_{k=1}^{j-1} p_k + 1, \dots, \sum_{k=1}^{j-1} p_k + p_j \right\},$$

with  $\sum_{j=1}^S p_j = N$ , where the temporal information is the semester. If we consider the delay as temporal information then the ideal career corresponds to the sequence

$$s_I^{[d]} = \langle \{1, \dots, N\} \rangle,$$

since the ideal student takes all exams with delay zero.

In Section 3, we examine a real student database by using both approaches.

## 2.2 Clustering on sequential patterns

Among the sequential patterns found in the previous section, we have to discover the most meaningful ones, that is, those having a higher support or showing a particular behavior. An obvious criterion is to consider the patterns of greater length or involving the maximum number of exams, and between them, to select those more regular, for example corresponding to subsequences of the ideal career; alternatively we can consider the patterns showing unexpected irregularities. Since we chose to denote the ideal career as the identity permutation, it is simple to discover patterns which are ordered subsequences of the ideal career. For example, according to the previous section, if  $N = 12$ ,  $S = 4$ ,  $p_1 = p_3 = 3$ ,  $p_2 = 4$ ,  $p_4 = 2$  and

$$s_I^{[s]} = \langle \{1, 2, 3\}\{4, 5, 6, 7\}\{8, 9, 10\}\{11, 12\} \rangle$$

then the sequence  $\langle \{3\}\{5, 7\} \rangle$  is obviously a subsequence of  $s_I$  while  $\langle \{8\}\{5\}\{7\} \rangle$  is not. The pattern selection step can be simplified by processing the result of SPAM to obtain a more readable format in which the frequent sequences are sorted according to the number of exams and to the length. However, this analysis cannot be completely automated and requires a deep knowledge of the context under examination. In particular, it is mandatory to know the organization of the laurea degree and the schedule of courses.

The database may contain several kinds of information about students. For example, for each student we can have the grade obtained at the high school level, the type of high school, the year of enrollment at the university, the mark of final examination and the length of the studies.

The next step consists in inserting in the database the information relative to the sequential patterns previously identified; in practice, every student verifying the pattern  $P$  has assigned value 1, 0 otherwise. The aim is to understand if students satisfying the patterns have some common characteristics. In Table 5 we illustrate a sample scenario corresponding to the patterns  $\langle \{1\}\{5\} \rangle$ ,  $\langle \{7\}\{5\} \rangle$ ,  $\langle \{7\}\{1\} \rangle$  and to Table 2. Once the database has been updated in this way, we can perform a clustering analysis to find out if there is correlation between student attributes and the sequential patterns. In Section 3, we apply this methodology to a real case study.

<b>Student</b>	$\langle \{1\}\{5\} \rangle$	$\langle \{7\}\{5\} \rangle$	$\langle \{7\}\{1\} \rangle$
10	1	1	1
20	0	0	1
30	0	0	1
40	1	0	0

**Table 5.** Some patterns for Table 2.

### 3 The case study

Our data set originates from a transactional database containing data of students in Computer Science at the University of Florence; in particular we started our analysis working on two data sets, the personal but anonymous data of students and their exams. The database contains data of graduated students which began their career during the years 2001-2007. For each student, the first data set contains, among other fields, the identifier of the student, **Student**, the year of enrollment, **Enrollment**, the date, **Date** and the mark, **FinalVote**, of final examination. In the second data set we have, for each exam, the identifier of the student, **Student**, the identifier of the exam, **Exam**, the date, **Date**, and the corresponding mark, **Vote**. For each student in the data set there are many items, one for each examination taken by the student. In Table 6 and Table 7 we show some data that are concerned with our analysis.

<b>Student</b>	<b>Enrollment</b>	<b>Date</b>	<b>FinalVote</b>
75	2001	2005 - 07 - 08	106
148	2002	2008 - 04 - 29	104
156	2002	2007 - 04 - 27	106
314	2003	2006 - 09 - 06	110 cum laude
⋮	⋮	⋮	⋮

**Table 6.** Data for students

<b>Student</b>	<b>Exam</b>	<b>Date</b>	<b>Vote</b>
75	18	2002-01-14	27
75	21	2002-02-20	28
75	15	2002-09-04	26
75	19	2004-01-29	25
⋮	⋮	⋮	⋮
148	2	2003-01-31	22
148	21	2003-02-25	25
148	1	2003-06-10	24
148	17	2004-06-04	27
⋮	⋮	⋮	⋮

**Table 7.** Data for exams

The academic degree in Computer Science at the University of Florence is structured in three years (Laurea triennale). A higher degree (Laurea Magistrale) exists but it was not considered in our analysis. Every academic year is organized in two semesters; there are several courses in each semester and at the end



of a semester students can take their examinations. Exams can be taken in different sessions during the same year and students can try to pass their exams in any of these sessions, after the end of the course. In particular, by searching on different sources, such as logs of old web-sites or old documents, we have been able to integrate original data with important information, that is, the semester in which the course has been given by a teacher. In fact, the original data set did not contain this information, which is fundamental for our purposes. Then, after some computations, for every exam of each student, we obtained the semester in which the exam was taken, a very important datum for frequent pattern analysis. Many other cleaning and error removing operations have been done on our original data, in particular we corrected dates or general errors about registration of exams. After the preprocessing step, our data set has been organized in a single file. In Table 8 we can see the most important information resulting from this preprocessing step. For each exam we have inserted three different types of temporal information:

1. **Semester1** is the semester in which the course has been given by the teacher and can assume values 1 through 6, relative to the year and the semester;
2. **Semester2** is the semester in which the student has taken the exam;
3. **Delay** is the difference between the previous two fields.

<b>Student</b>	<b>Exam</b>	<b>Date</b>	<b>Semester1</b>	<b>Semester2</b>	<b>Delay</b>
75	18	2002-01-14	1	1	0
75	21	2002-02-20	1	1	0
75	15	2002-09-04	2	2	0
75	19	2004-01-29	3	5	2
⋮	⋮	⋮	⋮	⋮	⋮
148	2	2003-01-31	1	1	0
148	21	2003-02-25	1	1	0
148	1	2003-06-10	2	2	0
148	17	2004-06-04	2	4	2
⋮	⋮	⋮	⋮	⋮	⋮

**Table 8.** Data for careers

Moreover, for each student, we added the attribute **Years** corresponding to the length of studies. From Table 8 we can easily obtain the file to use as input for the algorithm *SPAM*, by selecting, for each student, the fields **Student**, **Exam** and **Semester2** (or **Delay**). The file obtained in this way is just in the format accepted by *SPAM* and illustrated in Table 3.

During the years 2001-2007 the organization of the academic degree suffered several changes and provided different curricula. For this reason, it is not possible to analyze together all the exams of all the students in the database. So we decided to analyze the following two groups of students characterized by the same subset of exams:

- 1) a subset of the data containing 950 records corresponding to the 25 exams taken by 38 students of the same curriculum during the years 2001-2004;
- 2) a subset of the data containing 2256 records corresponding to the 16 exams taken by all the 141 students.

These data sets are not very large but allow us to illustrate the methodology on a real case study.

Frequent sequences
$\langle \{4\} \{8\} \{16, 17\} \{12\} \rangle 0.5$
$\langle \{2\} \{8\} \{16, 17\} \{25\} \rangle 0.5$
$\langle \{1\} \{10\} \{16, 17\} \{12\} \rangle 0.7$
$\langle \{2\} \{8\} \{3\} \{7\} \{12\} \rangle 0.5$
⋮
$\langle \{2\} \{8\} \{21\} \{12\} \rangle 0.5$
$\langle \{2\} \{8\} \{19\} \{12\} \rangle 0.6$
$\langle \{2\} \{8\} \{17\} \{25\} \rangle 0.6$
$\langle \{1\} \{13\} \{19\} \{18\} \rangle 0.6$
$\langle \{4\} \{8\} \{17\} \{12\} \rangle 0.5$
⋮
$\langle \{8\} \{23\} \rangle 1$
$\langle \{13\} \{23\} \rangle 1$
$\langle \{1\} \{23\} \rangle 1$
⋮
$\langle \{3\} \rangle 1$
$\langle \{2\} \rangle 1$
$\langle \{1\} \rangle 1$

**Table 9.** Some frequent sequences for the data set of students 1) :  $\text{minsup} = 0.5$

### 3.1 Postprocessing and analysis of the results

We ran SPAM on our data sets of students, by using several values of support. In Table 9 we can see a small subset of the 1271 patterns produced by SPAM on the students data set 1) by using the semester as temporal information and  $\text{minsup}$  equal to 0.5. We processed this result to obtain a more readable format in which the frequent sequences are sorted according to the number of events; the maximum value of events in a pattern was 5 and, in particular, we obtained twelve 5-sequences of length 4, one 5-sequence of length 5, almost two hundred 4-sequences and many other sequences involving a smaller number of exams. In this case the ideal career corresponds to the identity permutation of length 25 and the career of a generic student is a permutation of the integers 1 to 25. In particular, when the temporal information is the semester, we have

$$s_I^{[s]} = \langle \{1, 2, 3\} \{4, 5, 6, 7\} \{8, 9, 10, 11, 12\} \rangle$$

{13, 14, 15, 16, 17, 18}{19, 20, 21, 22, 23}{24, 25}.

By comparing this sequence with Table 9 we can find some interesting results. For example, we can observe that 70% of students take exams according to the pattern in the third line. This pattern corresponds to the ideal career except for exam with code 12 which should be taken before the pair 16, 17. Exam code 12 corresponds to *Numerical Calculus*, while the pair 16, 17 to *Operating systems* and the corresponding *Laboratory*. A more detailed analysis on the data base shows also that the average rating for exam 12 is lower than the corresponding rates for 16, 17 and that the average delay of exam 12 is much larger than the delay of the pair 16, 17. Therefore we can conclude that exam 12 is considered very difficult by students.

A similar analysis has been done for other frequent sequences. For example, line 6 in Table 9 show that the exams 19, *Software Engineering*, and 12, *Probability and Statistics*, are made in reverse order than expected from the ideal career by a large part of the students. Again, the average grade of exam 12 is smaller than that of exam 19.

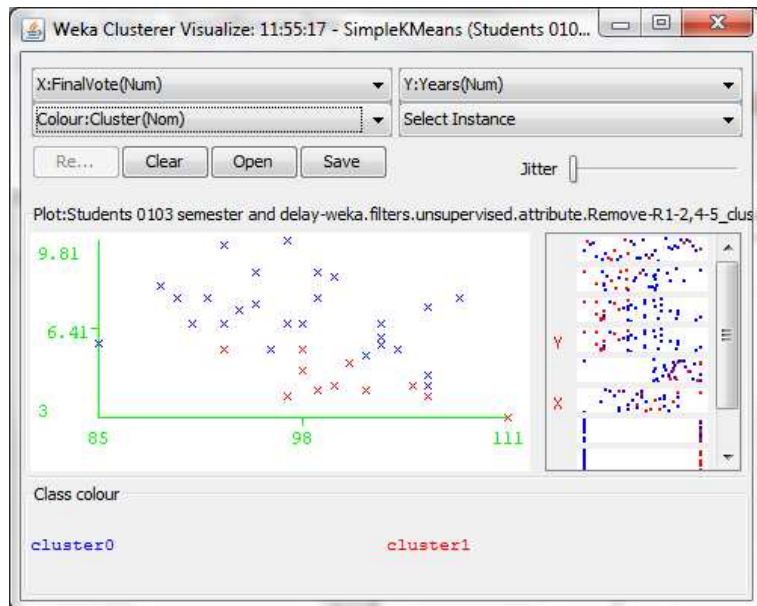
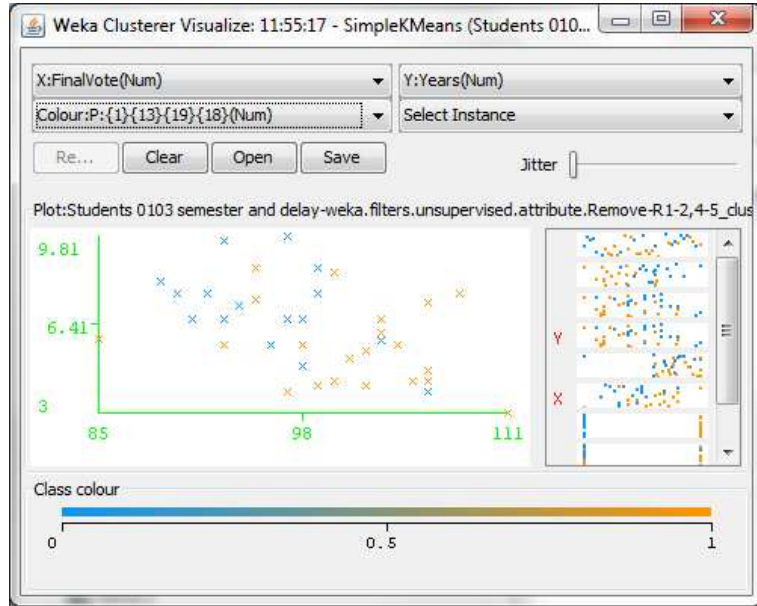


Fig. 1. Students of data set 1) with respect to FinalVote and Years.

By running SPAM with minsup equal to 0.5 we also obtained frequent patterns with minsup equal to 1; some of them are obvious, for example each exam was taken by 100% of the students, while others are non trivial, as for example  $\langle\{8\}\{23\}\rangle$ ,  $\langle\{13\}\{23\}\rangle$  and  $\langle\{1\}\{23\}\rangle$ . These patterns show that all students



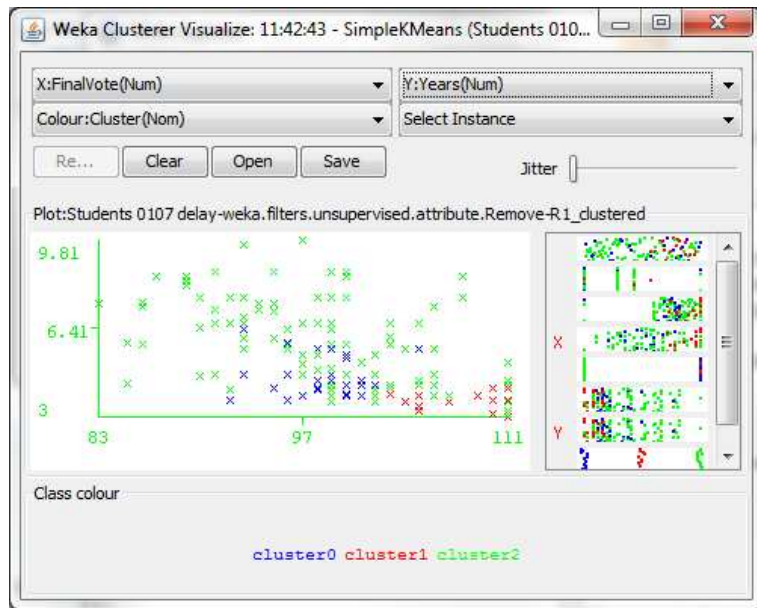
**Fig. 2.** Students of data set 1) with respect to **FinalVote** and **Years**: pattern  $P_1$  in evidence.

gave exam 23, *Theoretical Computer Science*, after exams 8, 13 and 1, corresponding to *Databases and Information Systems*, *Languages and Compilers* and *Programming*. At a first sight, these patterns might seem not interesting, because the exam 23 corresponds to a course of semester 5 (first semester of the third year), while exams 8, 13 and 1 correspond to courses given in previous semesters (3, 4 and 1, respectively). However, SPAM gives this kind of result only for exam 23, while there are other exams corresponding to courses given in semesters 5 and 6, for example exams 19, *Software Engineering*, and exam 25, *Data structures for databases*, which are not so frequently taken after other particular exams. Moreover, we explicitly observe that in the years under consideration, prerequisites between exams were not fixed. Again, the analysis highlights a difficulty of students to take the exam 23.

We also run SPAM on the students data set 1) by using the delay as temporal information and `minsup` equal to 0.3, thus obtaining 3853 patterns with maximum value of events equal to 6. In this case, we have  $s_I^{[d]} = \langle \{1, \dots, 25\} \rangle$  and we found the interesting 5-sequence  $\langle \{8, 13, 14, 15, 24\} \rangle$  of length 1 verified by 30% of students. Exam 14 corresponds to *Computer Networks*, exam 15 to *Laboratory of Information Systems* and, finally, exam 24 to *IT Work Organization*. We explicitly observe that all the exams in the pattern are Computer Science exams. An analysis of the database shows that the students verifying the pattern

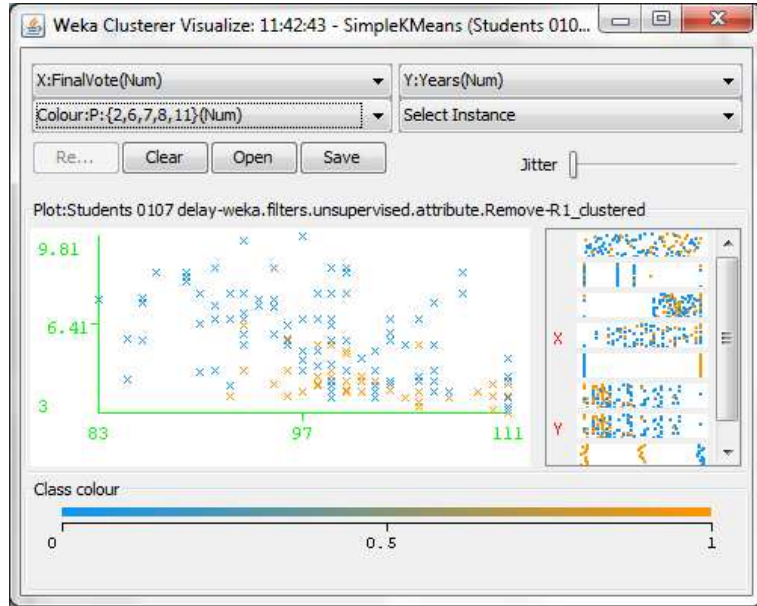
take the corresponding exams with delay equal to zero. We can conclude that students give without difficulties the exams in the pattern.

As illustrated in Section 2.2, we updated the database with the Boolean information about the frequent patterns which appeared to be most interesting and performed many tests by using the K-means implementation of WEKA (see, e.g., [14]). Among the various patterns, for data set 1), we considered the pattern  $P_1 = \langle \{1\} \{13\} \{19\} \{18\} \rangle$  with support 60% (see Table 9) and the pattern  $P_2 = \langle \{8, 13, 14, 15, 24\} \rangle$ , examined above. The cluster analysis on the attributes  $P_1$  and  $P_2$ , together with the final vote and graduation time, with  $K = 2$ , shows that students who satisfy both patterns achieve better results than those who do not satisfy them. Figure 1 illustrates the two distinct clusters: the red one corresponds to the group of students who graduated relatively quickly and with high grades; the other cluster corresponds to students who obtained worse results. Figure 2 highlights students who verify pattern  $P_1$ . A possible interpretation of this result is that pattern  $P_1$  involves an almost ordered subsequence of the ideal career, with only an inversion between 18 and 19, and  $P_2$  identify students without delay in the corresponding exams.



**Fig. 3.** Students of data set 2) with respect to FinalVote and Years.

For what concerns the larger data set 2), having  $s_7^{[d]} = \langle \{1, \dots, 16\} \rangle$ , the sequential pattern and clustering analysis gave similar results. In particular, we point out the pattern  $P_3 = \langle \{2, 6, 7, 8, 11\} \rangle$  verified by 45 students and obtained



**Fig. 4.** Students of data set 2) with respect to **FinalVote** and **Years**: pattern  $P_3$  in evidence.

by using the delay as temporal information. In this case, by using `minsup` equal to 0.3, SPAM produced 708 patterns with maximum value of events equal to 6. The cluster analysis on the attribute  $P_3$ , together with the final vote and graduation time, with  $K = 3$ , shows that students who satisfy the pattern (red and blue in Figure 3) achieve good results. Figure 4 highlights students verifying pattern  $P_3$ .

## 4 Conclusions

In this paper we investigated issues for mining sequential patterns in a database of university students. As far as we know, this is the first attempt to use the sequential pattern technique in the context of education. In particular, we proposed a methodology that is based on the concept of the *ideal career*, defined as the sequence of the exams followed by an ideal student who takes them without delay, and consists into the following main steps: i) the generation of the frequent patterns by using the SPAM algorithm, ii) the selection of the most interesting patterns and, finally, iii) the cluster analysis based on the results of the previous step. For what concerns the first step, we considered two possible types of temporal information: the *semester* in which students take exams and the *delay* with which exams are taken. A critical step concerns the selection of patterns and in Section 2.2 we give some hints to find the more interesting

ones. During this phase it is necessary a deep knowledge of the context under examination. As suggested by one of the referees, frequent pattern analysis produces a lot of useless patterns and therefore it is not easy to select the patterns which explain some interesting student behaviors. Instead of mining the complete set of frequent patterns, an alternative solution could be mining only *frequent closed sequential patterns* by using the **CloSpan** algorithm introduced in [15]. For example, according to Table 9, the SPAM algorithm produced the frequent patterns  $\langle\{4\}\{8\}\{16,17\}\{12\}\rangle$  and  $\langle\{4\}\{8\}\{17\}\{12\}\rangle$ , with the same support 0.5. It is clear that the second pattern does not give any additional information to our analysis. This pattern would not be generated by using the closed sequential pattern technique. Therefore, the **CloSpan** algorithm could simplify our methodology and we intend to proceed in this direction for future research. The frequent patterns found at this step allow us to understand which exams, or typology of exams, are considered difficult by students. For example, in a Computer Science degree, we expect that Mathematics exams tend to be considered more difficult than Computer Science exams. Concerning the third step, in order to find if there is a relation between the curriculum of students and the patterns, we decided to use them as binary attributes of our database and tried to cluster students by using the **K-means** algorithm. We can wonder whether information such as the sex, the place of birth, the grade obtained at the high school level and the year of enrollment at the university have a correlation with the frequent pattern analysis.

For example, in the case study examined in this paper, the patterns  $P_1$  and  $P_2$  (corresponding to the semester and to the delay as temporal information) divide students into *good* and *not so good*, according to the final grade and the length of study. Pattern  $P_1$  is a subsequence of the ideal career while  $P_2$  corresponds to a 5-sequence of computer science exams taken without delay. In other words, good students have taken most exams according to the order planned by the laurea degree, and this can mean that it was well structured. Moreover, they have taken several Computer Science exams without delay, which not happened for Mathematics exams.

We wish to point out that the results presented in this paper are intended to illustrate a methodology which can be applied to databases of any dimension containing various curriculum data of students and corresponding to different laurea degree programs.

## Acknowledgments

We wish to thank the referees for their useful comments and suggestions and Dino Pedreschi for the interesting discussions about the methodology presented in this paper.

## References

1. R. Agraval and R. Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, pages 3–14, 1995.

2. J. Ayres, J. Gehrke, T. Yiu, and J. Flannick. Sequential pattern mining using a bitmap representation. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 429–435, 2002.
3. R. Campagni, D. Merlini, and R. Sprugnoli. Analyzing paths in a student database. In *The 5th International Conference on Educational Data Mining*, pages 208–209, 2012.
4. K. Daimi and R. Miller. Analyzing student retention with data mining. In *Proceedings of the 2009 International Conference on Data Mining*, pages 55–60, 2009.
5. R. Damaševičius. Analysis of academic results for informatics course improvement using association rule mining. In *Information Systems Development*, pages 357–363. Springer, 2010.
6. N. Delavari, M. R. A. Shirazi, and M. R. Beikzadeh. A new model for using data mining technology in higher educational systems. In *Proceedings of the Fifth International Conference on Information Technology Based Higher Education and Training*, 2004.
7. N. Delavari, P. A. Somnuk, and M. R. Beikzadeh. Data mining application in higher learning institutions. *Informatics in Education*, 7(1):31–54, 2008.
8. H. Guruler, A. Istanbulu, and M. Karahasan. A new student performance analysing system using knowledge discovery in higher educational databases. *Computers & Education*, 5(1):247–254, 2010.
9. C. Romero, J. R. Romero, J. M. Luna, and S. Ventura. Mining rare association rules from e-learning data. In *The 3rd International Conference on Educational Data Mining*, pages 171–180, 2010.
10. C. Romero and S. Ventura. Educational Data Mining: A Review of the State of the Art. *IEEE Transactions on systems, man and cybernetics*, 40(6):601–618, 2010.
11. C. Romero, S. Ventura, and E. García. Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, 51(1):368–384, 2008.
12. SPAM. <http://himalaya-tools.sourceforge.net/Spam/>.
13. P. N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2006.
14. I. H. Witten, E. Frank, and M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Third Edition, Morgan Kaufmann, 2011.
15. X. Yan, J. Han, and R. Afshar. Clospan: Mining closed sequential patterns in large datasets. In *The Third SIAM International Conference on Data Mining*, pages 166–177, 2003.