Data Mining for a student database

R. Campagni, D. Merlini, R. Sprugnoli

Dipartimento di Sistemi e Informatica
Università di Firenze, Italia

ICTCS 2012, September 19-21 2012 - Villa Toeplitz, Varese (Italy)
Summary

1. Introduction

2. The perspective of the student
   - The case study

3. The perspective of each course

4. Conclusion and future analysis
EDM is an emerging research area that produces useful, previously unknown issues from educational database. It allows us to understand and improve the performance and assessment of the student learning process.


We consider the perspective of the student, who evaluates how difficult and important an exam is, in order to decide when to take it.

We consider the perspective of the single course, by analyzing the distribution of students with respect to the delay with which they take an examination.

The aim is to understand how the order of the exams affects the performance of the students in terms of *graduation time* and *final vote* and to know the behaviour of the distributions of delays.
We refer to an organization which allows students to give an exam in different sessions after the end of the course, as in Italy.

An academic year is divided into two semesters, during which the courses are taken according to the established curriculum.

Some constrains between exams can be fixed in order to force students to take some exams in a specific order.

Many students end up graduating with a significant delay.
The student database

The database contains information about the career of \( N \) students, characterized by \( n \) exams:

1. general information such as the sex, the place of birth, the grade obtained at the high school level, the year of enrollment at the university, the date and the grade of final examination;
2. information about each exam: the identifier of the exam, the date and the grade.
A student can take an exam in the same semester of the course or later, with a delay of one or more semesters.

- By introducing in the database the *semester* we can define an *ideal path* to be compared with the path of a generic student.
- The *ideal path* corresponds to a student, the *ideal student*, which has taken every examination just after the end of the corresponding course, without delay.
We define the *ideal path* as the sequence of exams \( \mathcal{I} = (e_1, e_2, \cdots, e_n) \), corresponding to the ideal student. The sequence for a generic student \( k \) is:

\[
S_k = (e_{\pi_k(1)}, e_{\pi_k(2)}, \cdots, e_{\pi_k(n)}),
\]

where \( e_{\pi_k(i)}, i = 1, \cdots, n \), is the identifier of the exam taken at time \( i \).

We can assume that \( e_i = i, i = 1, \cdots, n \); the *ideal path* becomes \( \mathcal{I} = (1, 2, \cdots, n) \) and \( S_k \) can be seen as a permutation of the integers 1 through \( n \).
The graphical representation of student careers

The career \((2, 1, 3, 5, 4)\) and the corresponding \textbf{ideal path}. 

---

R. Campagni, D. Merlini, R. Sprugnoli

Data Mining for a student database
An alternative representation of student careers

Since in the same semester there are many courses, the *ideal path is not unique*.

- We chose to sort courses relative to the same semester according to the preference of students.
- A different solution consists in giving the same identifier to courses in the same semester; for example, \((1, 1, 2, 2, 2, 3, 3)\) would represent a sequence of 7 exams, two in the first and third semester and three in the second.
- In the case study we will discuss this possibility.
A distance between paths and the inversion table

- We compare a path \( S_k \) with \( I \) by using the *Bubblesort distance*.

- The Bubblesort distance can be computed easily: it corresponds exactly to the number \( \sigma(\pi) \) of inversions in the permutation \( \pi = (\pi_1, \pi_2, \cdots, \pi_n) \) of the integers 1 through \( n \). An *inversion* is a pair \( i < j \) with \( \pi_i > \pi_j \).

- If \( q_j \) is the number of \( i < j \) with \( \pi_i > \pi_j \) then \( q = (q_1, q_2, \cdots, q_n) \) is called the *inversion table* of \( \pi \) and \( \sigma(\pi) = \sum_{j=1}^{n} q_j \).

<table>
<thead>
<tr>
<th>index</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \pi )</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>( q )</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
For each student $k$ we compute $\sigma(S_k)$, $k = 1, \ldots, N$, and we add this information in the database.

For each student the database contains:

- the graduation time, $\text{Time}$;
- the final grade, $\text{Vote}$;
- the Bubblesort distance and other personal information.

The idea is to understand if there exists a relation between the Bubblesort distance and the success of students.
We observe explicitly that students who have taken the exams in the same order can have different final grade and graduation time.

- We apply the K-means algorithm to student data and verify how it splits the students into $K$ groups.
- If we obtain clusters characterized by similar Bubblesort distance and well separated, we can have:
  1. the students having small distance achieve good performance $\Rightarrow$ the academic degree is well structured;
  2. there exist many good students with large distances $\Rightarrow$ the organization should probably be modified.

R. Campagni, D. Merlini, R. Sprugnoli

Data Mining for a student database
The aim is to classify students as talented or not and find the attributes which most influence their careers.

- We can use the data mining techniques based on decision trees for further analysis.
- We need to add to the database a new field class, which labels the students in different ways, according to the ranges of values of a particular attribute.
- This new attribute can be used to classify students, for example by using the C4.5 algorithm.
We can classify with respect to different class-attributes:

1. **Bubblesort_class** to divide students into $K$ different ways, according to the ranges of values of Bubblesort distance in the $K$ clusters previously found;

2. **Time_class** by which we can predict whether a student has a long (short) career;

3. **Vote_class** to know if a student will have a high (low) final grade.
The student database

- Students in Computer Science at the University of Florence beginning their career during the years 2001-2003 and graduated up to now.
- No constrains between exams were fixed, so students could take their exams almost in any order.
- $N = 100$ careers of students characterized by a sequence of $n = 25$ exams, for a total of 2500 exams.
We performed several tests by using the $K$-means implementation of the open source system WEKA by using the first ideal path.

- We obtained significant results with $K = 2$ by selecting as clustering attributes Time, Vote and Bubblesort distance.
- Students are well divided into two groups:
  - students who graduated relatively quickly and with high grades;
  - students who obtained worse results.
- Some relevant results have been also obtained with $K = 3$.
- We obtained analogous results by performing tests using the alternative ideal path in which courses of the same semester have the same identifiers.
Introduction
The perspective of the student
The perspective of each course
Conclusion and future analysis

The case study

Weka Clusterer Visualize: 14:27:49 - SimpleKM...

X: Bubblesort (Num)  Y: Time (Num)
Colour: Cluster (Nom) Select Instance

Plot: Student careers-weka.filters.unsupervised.attribute.Remove-R2,12_clustered

Class colour

cluster0 cluster1
Introduction

The perspective of the student

The perspective of each course

Conclusion and future analysis

The case study

R. Campagni, D. Merlini, R. Sprugnoli

Data Mining for a student database
We applied the C4.5 implementation of WEKA with different choices of attributes and class.

The most interesting tree we obtained classifies students with respect to small ($\leq 100$) and large ($> 100$) values of Bubblesort distance.

The results of clustering are confirmed; moreover, we found that the grades obtained at the high school influence the performance of students.
The perspective of the student

The perspective of each course

Conclusion and future analysis

The case study

R. Campagni, D. Merlini, R. Sprugnoli

Data Mining for a student database
We know that “good” students pass early each exam, but “not so good” students postpone most exams.

- We study the delay distribution of each exam in the hypothesis that it is a good parameter for classifying courses.
- In order to study the delays of exams, we consider the Poisson distribution with average $\lambda$ and probability mass function $P_\lambda(k) = e^{-\lambda} \cdot \lambda^k / k!$, where $k \geq 0$ is the delay, in years.
The Poisson distribution is *unimodal* and attains its maximum value at $k \approx \lambda$.

If $N$ is the number of students, $P_{\lambda}(0) \cdot N$ is the number of those who passed the exam within the first year; $P_{\lambda}(1) \cdot N$ are the students who passed during the second year, and so on.

Most of our distributions are *bimodal*, with a sharp peak at $k = 0$ and a second and smoother peak at $k = 2$ or $k = 3$. 
Introduction
The perspective of the student
The perspective of each course
Conclusion and future analysis

R. Campagni, D. Merlini, R. Sprugnoli
Data Mining for a student database
We can infer that there are two different distributions, one for *good* students and another for *not so good* students.

The two distributions are superimposed and generate the two peaks. By examining the distributions for each exam, we can expect that students are divided into two classes:

1. students who tend to take an exam as soon as a course is terminated;
2. students who delay difficult exams to the end of their career.
The method -1

We consider $n$ courses $c_1, c_2, \cdots, c_n$ taken by $N$ students and a database containing the number of students $D_{c_i}(k)$ which take the exam with delay $k$, for $k = 0, \cdots, d_i$, where $d_i$ is the maximum delay.

- Our algorithm determines the average values $\lambda_g$ and $\lambda_{ng}$, which characterize the two Poisson distributions, and the corresponding numbers $N(\lambda_g)$ and $N(\lambda_{ng})$ of students.

- We can make the hypothesis that the $\lambda_g$-distribution decreases very fast so that it reduces to $k = 0, 1$ as meaningful values.
The method -2

- Our first step consists in separating the first two values of the exam distribution from the rest.
- We approximate the $\lambda_{ng}$-distribution determining the relative average value; we iterate this approximation process until a fixed point is obtained.
- This process can modify the values for $k = 0$ and $k = 1$, so that we have to use these new values to approximate the $\lambda_{g}$-distribution. We proceed until a fixed point is found.
- The algorithm stops here returning, for each course, the approximation of the two distributions.
We applied the algorithm to $n = 15$ courses taken by $N = 152$ students in Computer Science at the University of Florence.

- For each course $c_i$ we obtained
  $$D_{c_i}(k) \sim P_{\lambda_{g_i}}(k) \cdot N(\lambda_{g_i}) + P_{\lambda_{ng_i}}(k) \cdot N(\lambda_{ng_i}).$$
- Computer Science exams are characterized by
  $$N(\lambda_g)/N \sim 70\%.$$  
- Mathematics exams are delayed and often appear as the last exams taken before the final examination.
Our methodology analyzes a student database to understand how the order of the exams affects the performance of the students and if this order has relations with student attributes.

This analysis can help us to understand if a laurea degree is well structured or if we have to modify and improve it, by introducing some constraints between the exams.

By considering that students take their exams with delays, we showed that the corresponding distributions can be approximate by two Poisson distributions.

Our analysis can be tested on different and larger databases, after an important preprocessing phase.

Thanks for your attention!