Data mining for 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

1 Introduction

Educational data mining is an emerging research area that produces useful, previously unknown issues from educational database for better understanding and improving the performance and assessment of the student learning process (see [2] and the included references, for a detailed description of the state of the art in this context). This paper presents some data mining models to analyze the *careers* of University students and extends the research illustrated in [1], introducing a new approach to this research area. The career of a student can be analyzed from various points of view, among which the following two are particularly important: i) the perspective of the student, who evaluates how difficult and important an exam is, in order to decide to take it immediately at the end of the course, or delay it as much as possible; this aspect is studied in Section 2 with cluster and classification algorithms by introducing a notion of distance between careers; and ii) the perspective of each course, by analyzing the distribution of students with respect to the delay with which they take an examination, to discover common characteristics between two or more courses; this is done in Section 3 in terms of Poisson distributions.

2 The perspective of the student

The methodology we propose is based on a database containing information about students and their exams in a University organization. In particular, for each student, the database contains 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 besides information about each exam, that is, the identifier of the exam, the date and the grade. We refer to an organization of the university which allows students to take an exam in different sessions after the end of the course, as in Italy. Some constrains between exams can be fixed in order to force students to take some exams in a specific order, however, usually students have many degrees of freedom to choose their own order of exams. An important information which is a basilar aspect of our methodology is the *semester*; an academic year is divided into two semesters, during which the courses are taken according to the established curriculum. A student can take an exam in the same semester of the course, that is just after the end of the course, or later, with a delay of one or more semesters. This information allows us to define an *ideal path* to be compared with the path of a generic student. More precisely, we consider a database containing the data of N students, each student characterized by a sequence of n exams identifiers and a particular path $\mathcal{I} = (e_1, e_2, \cdots, e_n)$, the *ideal path*[1], corresponding to the ideal student who has taken every examination just after the end of the corresponding course, without delay. Without loss of generality, we can assume that $e_i = i$, $i = 1, \cdots, n$, that is, $\mathcal{I} = (1, 2, \cdots, n)$. The path of a generic student k with $k = 1, \cdots, N$, can be seen as a sequence $\mathcal{S}_k = (e_{\pi_k(1)}, e_{\pi_k(2)}, \cdots, e_{\pi_k(n)})$ of n exams, where $e_{\pi_k(i)}$, $i = 1, \cdots, n$, is the identifier of the exam taken by the student k at time i and π_k indicates the corresponding permutation of $1, \cdots, n$. Therefore, $\mathcal{S}_k$ can be seen as a permutation of the integers 1 through n. The idea is to understand how the order of the exams affects the final result of students. To this purpose, we compare a path $\mathcal{S}_k$ with $\mathcal{I}$ by using the *Bubblesort distance*, which is defined as the number of exchanges performed by the Bubblesort algorithm to sort an array containing the numbers from 1 to n. The number of exchanges can be computed easily since it corresponds exactly to the number of inversions in the permutation. Given a 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 π. We use the notation $\sigma(\pi)$ to denote the number of inversions in the permutation, that is, the sum of the entries in the inversion table: $\sigma(\pi) = \sum_{j=1}^{n} q_j$. For example, the permutation $\pi = (5, 2, 3, 1, 4)$ corresponds to $q = (0, 1, 1, 3, 1)$ and $\sigma(\pi) = 6$.

The path $\mathcal{S}_k$ of a generic student k can be compared with the ideal path $\mathcal{I}$ by computing $\sigma(\mathcal{S}_k)$, $k = 1, \cdots, N$. After this preprocessing phase, we can assume that for each student our database contains at least the following information: the graduation time, `Time`, the final grade, `Vote`, and the `Bubblesort` distance, together with other personal information. We can proceed by applying a cluster algorithm, for example `K-means`[2] (see e.g, [3]). If the cluster algorithm splits the students into K well defined groups characterized by similar Bubblesort distance, we can infer important conclusions about students and the laurea degree. We observe explicitly that students who have taken the exams in the same order, that is, students with the same path, can have different final grade and graduation time. The idea is to understand if there exists a relation between the Bubblesort distance and the success of students. If the students having small distance achieve good performance, then we may conclude that the academic degree is well structured but if there exist many good students with large distances, then the organization should probably be modified. We can extend our analysis

[1] Since in the same semester there are many courses, the ideal path is not unique. In this paper we 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.

[2] We wish to point out that the Bubblesort distance is an attribute inserted in our database and that we use `K-means` with the Euclidean distance.

through the technique based on decision trees. To this purpose, we need to add to the database a new attribute `Bubblesort_class` which labels the students into K different ways, according to the ranges of values of Bubblesort distance in the K clusters previously found. This new attribute can be used to classify students, for example by using the `C4.5` algorithm (see e.g, [3]). The aim is to classify students as talented or not and find the attributes which most influence their career. We can also try to classify with respect to other attributes: for example, we can predict whether a student has a long (short) career or obtains a high (low) final grade by introducing a `Time_class` or a `Vote_class` attribute in the database. The greater are the database and the information in it, the more accurate will result the model based on this technique.

The database we analyze contains data of students in Computer Science at the University of Florence beginning their career during the years 2001-2003 and graduated up to now. This academic degree is structured in three years, each divided in two semesters. In the years under consideration, no constrains between exams were fixed, so students could take their exams almost in any order. In particular, we analyzed the careers of $N = 100$ students characterized by a sequence of $n = 25$ exams. We computed the ideal path through an important pre-processing phase, which allowed us to identify the semester in which courses were originally hold. Then, for each student we computed the `Bubblesort` distance and added this value to the database. To understand how the order of the exams affects the career of the students, we have performed several tests by using the `K-means` implementation of `WEKA` (see, e.g., [4]). We obtained significant result with $K = 2$ by selecting as clustering attributes `Time`, `Vote` and `Bubblesort` distance. In fact, with these parameters we can see that students are well divided into two groups: students who graduated relatively quickly and with high grades and students who obtained worse results. Luckily, we observed that students in the first group are characterized by *small* values of `Bubblesort` while students in the second group have *larger* values. This result confirms that the more students follow the order taken by the ideal path, the more they obtain good performance in terms of graduation time and final grade. For what concerns classification, 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* (≤ 100) and *large* (> 100) values of Bubblesort distance confirming the result of clustering and, moreover, highlights that the results obtained at the high school influence the performance of students.

3 The perspective of the course: delayed exams

Usually, "good" students try to pass early every exam, but "not so good" students prefer to postpone most exams, especially if they are considered too difficult or too technical. We are interested in studying the delay distribution of every exam in the hypothesis that it is a good parameter for classifying students and/or courses. In general, delays conform to some Poisson distribution, with average (and variance) λ and probability mass function $P_\lambda(k) = e^{-\lambda} \cdot \lambda^k/k!$ for

$k \geq 0$. The Poisson distribution is discrete and, in our case, k represents the delay of the exam from the end of the course, measured in full years. So, 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. Finally, the distribution is unimodal and attains its maximum value at $k \approx \lambda$. If we look at the actual distributions of students with respect to the delay with which they took their examinations, we observe that most of them are bimodal, with a sharp peak at $k = 0$ and a second and smoother peak at $k = 2$ or $k = 3$. The obvious interpretation is that there are two different distributions, the first one relative to "good" students and the second relative to "not so good" students, who delay their exams of about two years. The two distributions are superimposed and generate the two peaks. In other words, by examining the distributions for each exam, we can infer that students are divided into two classes: students who tend to take an exam as soon as a course is terminated, and students who delay difficult exams to the end of their career. In order to analyze this behavior in a more formal way, we need to find the two Poisson distributions. We consider n courses $c_1, c_2, \cdots, c_n$ taken by N students and a database containing, for each course c_i, 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 relative to course c_i. We then use the following algorithm to determine the average values λ_g and λ_{ng} characterizing the two Poisson distributions and the corresponding numbers $N(\lambda_g)$ and $N(\lambda_{ng})$ of students. We can make the hypothesis that the λ_g-distribution decreases very fast so that it reduces to $k = 0, 1$ as meaningful values. Our first step consists in separating the first two values from the rest and try to approximate the λ_{ng}-distribution. 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 λ_g-distribution. Again, we proceed until a fixed point is found. The algorithm stops here returning, for each course, the two desired approximations.

We applied the algorithm to $n = 15$ courses taken by $N = 152$ students in Computer Science at the University of Florence. The analysis confirmed that for each course c_i we have $D_{c_i}(k) \sim P_{\lambda_{g_i}}(k) \cdot N(\lambda_{g_i}) + P_{\lambda_{ng_i}}(k) \cdot N(\lambda_{ng_i})$, with a good approximation. In particular, we found that Computer Science exams are characterized by $N(\lambda_g)/N \sim 70\%$. Instead, Mathematics exams are delayed and often appear as the last exams taken before the final examination.

References

1. R. Campagni, D. Merlini, and R. Sprugnoli. Analyzing paths in a student database. In *The 5th International Conference on Educational Data Mining*, 208–209, 2012.
2. 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.
3. P. N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining.* Addison-Wesley, 2006.
4. I. H. Witten, E. Frank, and M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques, Third Edition.* Morgan Kaufmann, 2011.