Student Performance Prediction and Optimal Course Selection: An MDP Approach

Michael Backenköhler(✉) and Verena Wolf

Computer Science Department, Saarland Informatics Campus,
Saarbrücken, Germany
michael.backenkoehler@uni-saarland.de

Abstract. Improving the performance of students is an important challenge for higher education institutions. At most European universities, duration and completion rate of degrees are highly varying and consulting services are offered to increase student achievement. Here, we propose a data analytics approach to determine optimal choices for the courses of the next term. We use machine learning techniques to predict the performance of a student in upcoming courses. These prediction form the transition probabilities of a Markov decision process (MDP) that describes the course of studies of a student. Using this model we plan to explore the effect of different strategies on student performance.

1 Introduction

Understanding and improving learning conditions of students has always been an important objective in educational institutions and different policies and interventions have been developed to improve learning processes and students' performance. Recently, educational data has been systematically analysed and used to predict students' future learning. This includes the design of models that incorporate information such as students' knowledge, talents, motivation, and attitudes. This newly emerging field, called educational data mining (EDM), focuses on the exploration and analysis of data that comes from the educational setting [7]. One goal is the prediction of students' performance and the decrease of high dropout rates at higher education institutions since high dropout rates obviously have social and economic disadvantages. In addition to dropout, also the time-to-degree and study success is an important issue on the European policy agenda and continuously monitored by higher education research institutes. Completion rates at European universities range from 39% to 85% and are highly program dependent, while the average time-to-degree is around 3.5 years for a Bachelor degree [9].

In many study programs, students can choose between different courses, the order in which they take courses, or change the degree program. In the latter case, if the newly chosen program is similar to the previous one, the accomplishments of some earlier courses can be transferred. The goal of this work is to use data of students' performance to analyse the efficiency of choices made by students concerning the courses and the program and to predict their performance

A. Cerone and M. Roveri (Eds.): SEFM 2017 Workshops, LNCS 10729, pp. 40–47, 2018.
https://doi.org/10.1007/978-3-319-74781-1_3

for different organisational strategies. For this, we preprocessed (anonymized) performance data of 3889 computer science students since 2006 which have been enrolled at Saarland University. We design a Markov decision process (MDP) that describes the performance and course selection of a student for every semester until graduation. A state of the MDP corresponds to the transcript of the student in a certain semester and the possible actions correspond to subsets of course that the student may take in the upcoming semester. The MDP will be equipped with costs/rewards that are related to the number of credit points of the course as well as the number of semesters and the expected final graduation grade. We are interested in reaching goal states where the student has successfully graduated. It is possible to consider optimization criteria such as timely graduation or best grades, however, students may only fix bounds for the number of semesters, credit points, and grade and consider other objectives such as maximize knowledge in a certain field,etc. Hence, we aim at a general MDP model that allows an analysis w.r.t. different objectives and constraints. Moreover, our goal is to analyze whether students chose their courses in a sub-optimal way and check whether failure of courses or a disadvantageous order of courses could have been avoided.

A key challenge is the estimation of transition probabilities of the MDP, i.e., the probability to achieve certain grades in a certain set of courses. Grade prediction or the simpler task of estimating the probability of passing a course has been studied earlier using linear regression and matrix factorization models [1,6]. While online learning environments provide richer information for grade prediction, estimations for traditional learning environments rely on similarities between currently offered courses and courses offered in earlier semesters as well as similarities between students.

In this paper, we present preliminary results for the grade prediction based on different machine learning methods. These results were computed on the basis of the data mentioned above. In addition, we discuss our idea of an higher education studies MDP and important challenges that arise during its construction and analysis. Our findings indicate that individual study counseling can be improved by course recommendation that are based on results of the proposed MDP model.

2 Background

We propose to model the higher education studies of a student using an MDP, which necessitates the estimation of model parameters. This and the design of the MDP require careful analysis of real-world data provided by electronic administrative systems of the educational institution. Here, we use data from computer science students at Saarland University, which includes some basic metadata regarding the students, as well as information about course performance. The performance of a student in a certain course is described by an entry that includes the course title, the name of the lecturer, the awarded credit points (CP) in the European credit transfer and accumulation system (ECTS), the number of attempts, and the final grade. Some of the courses are offered regularly such as basic and core lectures. About the student, the year of birth, nationality, gender, and enrolment history is known.

There are many case-specific data-related challenges which make the analysis difficult. For example, the available records do not provide reliable information to which category a certain course belongs. Moreover, grades may be missing since student's have some freedom to choose which grades contribute to their final GPA [8]. Those that do not contribute are removed in the database, which naturally introduces an upward bias in the available data.

3 MDP for Higher Education Studies

In this section we define the Markov decision process (MDP) that models the progression of a student through the program of studies. A state of the MDP is given by the student's birth date, nationality, and transcript $T \in \mathcal{T}$ after t semesters, where t is a non-negative integer that denotes the semester relative to the student beginning the program and $\mathcal{T}$ is the set of all possible transcripts. The transcript consists of tuples (C, G), where C describes a course and $G \in \{1, 1.3, 1.7, 2, 2.3, 2.7, 3, 3.3, 3.7, 4, 5\}$ the assigned grade, where 1 is best and 5 is worst. For courses, we record the title, lecturer, amount of awarded credit points, and the category c according to the program regulations.

The available actions of the student are given by all possible subsets of the set of courses offered in the upcoming semester, constrained by the program regulations. The grades of a student, i.e. the outcome probabilities, are predicted using the methods described in Sect. 4. At the core of the model there is the reward function, that specifies how *valuable* a certain outcome is to the student. We consider two primary goals which are universal for a great majority of students:

1. Achieve a good overall grade.
2. Obtain a degree in a timely manner.

The first goal can be dealt with by a defining function $g : \mathcal{T} \to [1, 5]$ of an arbitrary (non-empty) transcript to the current overall grade according to the program's regulations. This function can now be used to compute the improvement (positive or negative) the outcome of some action imposes on the current overall grade. The second goal can be handled by consideration of the amount a certain course contributes to obtaining a degree. A program usually prescribes a certain amount of credit points that have to be achieved in certain categories along with some mandatory courses. All necessary credit points amount to a fixed number of credit points. The progress towards reaching that number gives a measure of the second goal. Let $p : \mathcal{T} \to [0, 1]$ be a function that computes the ratio of completed to missing requirements for some degree. With these two measures at our disposal we can define the reward as some combination, e.g. a linear function, of both criteria.

Following the above intuition we can now define an MDP that describes higher education studies of a student over time (HES-MDP).

Definition 1. HES-MDP. *Let $\mathcal{T}_0$ be an initial non-empty transcript of the student under consideration. Then the* Higher Education Studies MDP (HES-MDP) *is the tuple $(\mathcal{T}, \{\mathcal{A}_t | t > 0\}, P, R_c)$ where each component is defined as follows.*

- *The* state space $\mathcal{T}$ *is the set of all transcripts of grades $T \in \mathcal{T}$.*
- *The set of* actions, $\mathcal{A}_t$, *that are possible at time t consists of all subsets of available courses that the student can choose in semester t.*
- *The* transition probability function $P : \mathcal{T} \times \mathcal{A}_{t+1} \times \mathcal{T} \rightarrow [0, 1]$ *computes for a given transcript in semester t the probability of each possible transcript after semester t if the student chooses a certain set of courses in semester t. Thus, if T_t is the transcript in semester t then, for a set A of courses, $P(T_t, A, T_{t+1})$ is the probability to achieve transcript T_{t+1} after semester t.*
- *For a fixed scaling factor $c > 0$, the* (immediate) reward function $R_c : \mathcal{T} \times \mathcal{T} \rightarrow \mathbb{R}$ *computes the reward gained during a state transition from T_t to T_{t+1} as the linear combination $R_c(T_t, T_{t+1}) = c \cdot (g(T_t) - g(T_{t+1})) + p(T_{t+1}) - p(T_t)$.*

The above MDP model can be simplified by disregarding the specific grade of a student and only recording whether the student passed the course or not. Accordingly, the grade difference term is dropped from the reward function.

Since we are considering the progression of a student through a program, not only the course performance is important. Additionally, a student may drop out of the program at any given time. There are two possible drop-out scenarios: (i) The student is forced out of the program due to poor performance. (ii) The student chooses to quit. In case of scenario (i) there are simply no applicable actions left. While (i) can be computed based on the transcript, (ii) is an event that, in principle, can occur for any student. Here, it could be sensible to train a separate model, predicting a dropout probability, which is then used to condition the grade prediction. Then the set of possible actions of a state is enriched by a transition to an absorbing state corresponding to student dropout with the probability taken from the dropout model.

3.1 Simulation

We plan to investigate the MDP by means of a standard Monte-Carlo tree search (MCTS) algorithm. In each semester a student has, in principle, the opportunity to select any subset of offered lectures. It is therefore sensible to exclude choices which are "unreasonable", which primarily bounds the size of considered subsets of courses.

Furthermore it is not known a-priori for many lectures whether they will be offered in some future semester. For example, advanced lectures or seminars may only be offered only once or irregularly. Therefore we can only assume to know which lectures are offered in the next semester. In other future semesters courses can be treated as dummies of the corresponding category. Consequently, outcome probabilities are computed by aggregation of all courses of the same category. The simulation could be done by standard algorithms such as MCTS with UCB action selection [3].

4 Performance Prediction

We now discuss the estimation of MDP transition probabilities, which amounts to the prediction of grades. The prediction of grade values can be interpreted as either a *classification* problem, or a *regression* problem in which the grade domain is extended to the real line. Here we consider the latter approach.

4.1 Prediction of Course Performance

We obtained the most reliable predictions when using aggregated performance features instead of individual performances. As features for a fixed pair of student and course, we use the current

- percentage of achieved CP w.r.t. the attempted CPs
- grade point average (GPA) weighted by credit points
- average achieved CPs per semester
- GPA weighted by the correlation of the course with other courses
- semesters of the student in the program
- age of the student
- number of attempts for the course
- overall number of CPs
- deviation of attempted CPs in a semester from the average achieved CPs

If there is no data for an aggregate value such as a mean the overall mean is imputed. As the response variable we used the grade value as given by the course data. A fundamental challenge in the prediction of grades is, that the failing grade 5 does not capture how far the student was from not failing.

All of these features are normalized to mean 0 and standard deviation 1. To perform grade predictions, we trained models separately for individual (recurring) courses. The techniques considered here are

- a simple linear regression (LR)
- Regularized linear regression (LASSO)
- Random Forests (RF)
- Gaussian Processes (GPR)
- Nearest Neighbors regression (NNR)

Except for the first two, each of the methods returns not only an estimated grade but also estimated variances, which can be used to fit a discretised normal distribution over possible grades. For LR and LASSO, variances can be obtained via bootstrapping. When simulating the MDP, we run the grade prediction in an on-the-fly fashion for the current state. For details on the above methods we refer the interested reader to the common literature [2,4].

Despite factors that take into account the semester workload of the student, the fitted models do often not realistically predict the performance, when a large set of courses is chosen. In general, the performance may be worse due to the high workload. Information on attempted versus completed courses is in general sparse since students can revoke their registration late in the semester. Thus, the data does not include lectures that have been dropped earlier in the semester. An approach to tackle this, is to estimate the expected amount of credit points per semester for a student. However, it is possible to learn models to estimate the amount of credit points a student is going to complete in a given model. For example, Fig. 1 (left) shows that students with higher GPA clearly achieve on average more credit points per semester.

For now, we make the simplifying assumption that, as long as the sum of the CPs of the chosen courses (attempted CPs) remains in a certain range, obtained grades are independent. Hence, assume that the student chooses m courses in the current semester. Then, we estimate the probability of obtaining a corresponding vector of m grades based on a multivariate normal distribution, whose vector of means contains the estimated grades (cf. Fig. 1 (right)) using one of the methods mentioned above and whose covariance matrix is a diagonal matrix with the corresponding variances. Methods like GPR directly yield such distributions while other methods such as linear regression may require bootstrapping. The grades drawn from the normal distribution have then to be discretised in order to keep the state space discrete. For example, a threshold has to be fixed which corresponds to a student failing a lecture (cf. Fig. 1 (right)). Depending on the computational effort one can also scale the "resolution" to larger ranges to reduce the state space's size.

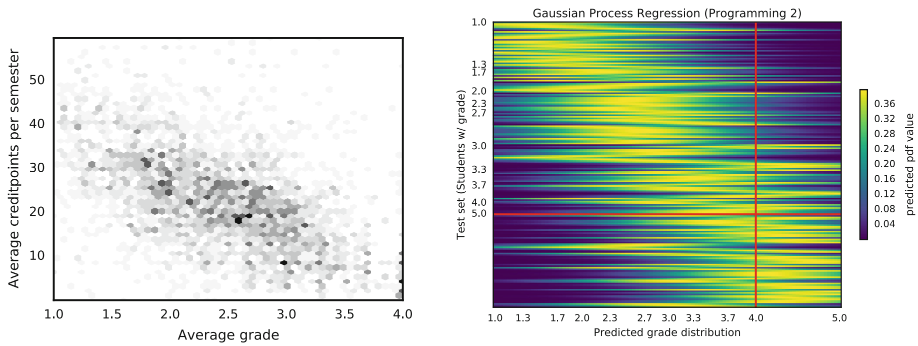

Fig. 1. (left) The average number of CPs per semester versus the overall average grade. Outliers above an average of 60CPs are excluded for clarity. The Pearson correlation coefficient is approx. -0.5217. (right) The predicted probability densities resulting from GPR for the lecture "Programming 2" in the summer semester 2013.

5 Evaluation

A prototype implementation of the feature extraction, grade prediction, and simulation was implemented in Python using standard libraries. Notably, the implementations of the machine learning methods as provided by Scikit-learn [5] were used.

5.1 Evaluating Course Performance

For a fixed semester t and a fixed course, we split the data into test and training data, i.e., all previous data ($t < t_k$) constitutes the training data, while the data of the threshold semester ($t = t_k$) gives the test data. We compute hyper-parameters of the model using a three-fold cross validation on the training-data. Then the regressor is refitted on all the training data and grade estimation on the test data is performed. This is performed for all semesters in which the course was offered (except for the first two times to avoid a cold start).

We evaluate the predictor quality by a metric that normalizes the mean squared error (MSE) of the predictor with the MSE of the mean predictor. Consider a test set of N students with transcripts T_i and grade G_i in a fixed course ($i \in \{1, 2 \ldots, N\}$). Let $R(T_i)$ be the estimated grade in model R. Then, we define

$$E_R(\{T_1, \ldots, T_N\}) = 1 - \frac{\sum_i (G_i - R(T_i))^2}{\sum_i (G_i - \bar{G}^{\text{train}})^2}$$

where $\bar{G}^{\text{train}}$ is the average grade over the training data.

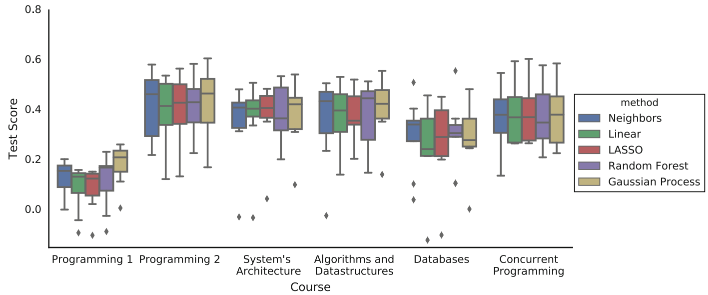

Fig. 2. The prediction error over in different semesters for a selection of recurring lectures.

In Fig. 2 we present the estimation error for some recurring lectures. We can observe that very simple models such as a linear regressor, yield predictions of similar accuracy as more involved models. The weaker results for the lecture "Programming 1" (left) are due to the fact, that for most students no information beyond their metadata is available. Here we can also observe that GPR outperforms other methods such as simple linear models. Disregarding "Programming 1" the average scores over basic lectures is highest using GPR (0.4106) while it is slightly lower using other techniques: The average values are 0.3716 (LR), 0.3947 (LASSO), 0.3950 (NNR), and 0.4033 (RF).

6 Conclusion

In this work, we sketched a Markov decision process to describe the choices of a student during her higher education studies. We identified challenges and strategies to perform simulations of that MDP based on data from earlier semesters. To this end we used machine learning methods to predict in each semester the grades the student is likely to obtain based on her previous performance. This prediction can be used to determine the transition probabilities of the MDP and simulate the progress and performance of the student until graduation.

We found that for regularly offered large courses, accurate predictions are possible, while in uncommon cases (e.g. predicting a grade for an advanced maths course if basic maths courses have not been taken before) obviously an accurate prediction is hard. As future work, we plan to improve the estimation of transition probabilities by modeling and predicting dropout (probabilities), and by investigating further the dependency between workload and obtained grades. We will analyse the MDP in detail and compare action choices which are optimal w.r.t. certain criteria against the choice that students made. Finally, we plan to provide a tool that suggests for each student and semester a set of courses optimal w.r.t. adjustable criteria.

Acknowledgments. We thank Jens Dittrich, Endre Palatinus, and Thilo Krüger for pre-processing the data and interesting discussions.

References

1. Elbadrawy, A., Polyzou, A., Ren, Z., Sweeney, M., Karypis, G., Rangwala, H.: Predicting student performance using personalized analytics. Computer **49**(4), 61–69 (2016)
2. Friedman, J., Hastie, T., Tibshirani, R.: The Elements of Statistical Learning, vol. 1. Springer series in statistics Springer, Berlin (2001). https://doi.org/10.1007/978-0-387-84858-7
3. Kocsis, L., Szepesvári, C.: Bandit based Monte-Carlo planning. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212, pp. 282–293. Springer, Heidelberg (2006). https://doi.org/10.1007/11871842_29
4. Murphy, K.P.: Machine Learning: A Probabilistic Perspective. MIT press, Cambridge (2012)
5. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
6. Polyzou, A., Karypis, G.: Grade prediction with models specific to students and courses. Int. J. Data Sci. Anal. **2**(3–4), 159–171 (2016)
7. Romero, C., Ventura, S.: Educational data mining: a review of the state of the art. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **40**(6), 601–618 (2010)
8. Bachelor Program Regulations, Saarland University (2015). http://www.ps-mint.uni-saarland.de/fileadmin/user_upload/_imported/fileadmin/Benutzerdaten/Downloads/Formulare/Informatik/StO_PO/StO_BA_Informatik_2015.pdf. Accessed June 2017
9. Vossensteyn, H., Kottmann, A., Jongbloed, B., Kaiser, F., Cremonini, L., Stensaker, B., Hovdhaugen, E., Wollscheid, S.: Dropout and completion in higher education in Europe: main report (2015)