

Exploring and Following Students' Strategies When Completing Their Weekly Tasks

Jessica McBroom, Bryn Jeffries, Irena Koprinska and Kalina Yacef

School of Information Technologies, University of Sydney, Sydney, NSW 2006, Australia

jmc6755@uni.sydney.edu.au, {bryn.jeffries irena.koprinska kalina.yacef}@sydney.edu.au

ABSTRACT

In this paper, we explore methods of analysing data obtained from an autograding system involving weekly tasks and a finite set of possible strategies for completing these tasks. We present an approach to handling partially missing information and also investigate the usefulness of a sliding window rule mining technique in following changes in student strategy over time.

Keywords

Mining student behaviour and strategies, autograding system

1. INTRODUCTION

Teaching activities are often not offered in a linear way: it is sometimes useful to provide students with several choices of task, or to provide a gradual approach to learning by allowing a choice of tasks of varying difficulty. Maximum points could be achieved through implementing all the hard tasks, but students unsure of their ability might choose to take a more gradual approach, starting with the easy task and working up. We wish to understand how students manage their learning when presented with such choices by analysing the order in which students attempt such tasks. We investigate the following research questions: What strategies do students take in attempting the different tasks each week? Are there differences between the strategies of the regular and advanced students? In this work we report on several techniques applied to the data collected through an autograding system in a university database course. Our main contribution is in showing how to represent and mine data from student attempts of tasks with different levels of difficulty.

2. DATA

The data comes from weekly programming tasks in a third-year database course with students in a regular stream [2] ($n=92$), and an advanced stream [3] ($n=20$). Part of the assessment, for 10% of the final grade, was a set of weekly programming tasks for which students were required to implement various algorithms in Java and submit these implementations using the PASTA online submission platform [1]. Tasks included skeleton code and unit tests, and students were encouraged to write and test their implementations locally before submitting. Once submitted to PASTA, the unit tests were applied again, and students received automated feedback of the outcomes of these tests. Students then had the option of submitting a revised attempt, or trying another of the three tasks, until the submission deadline had been reached.

Each week there was a choice of three tasks with different levels of difficulty - easy, medium and hard. More marks were allocated for the more difficult tasks: 4 points for hard, 3 for medium, and 2 for easy tasks. Partial implementation of any task received 1 point. The data extracted from PASTA consisted of the marks for every student's attempt on each task.

3. STRATEGIES

There are 16 possible strategies that can be taken by a student for each weekly set of tasks: the 15 possible permutations of Easy (E), Medium (M) and Hard (H) tasks attempted, and no attempt at any task (None). Figure 1 shows the relative frequency of the different strategies taken by all students each week, and in total across all weeks. We labelled each strategy according to the order in which the tasks were completed. So, for instance, in the strategy EH a student completes that week's Easy task first, followed by the Hard task. Note though that this information is imperfect: students were only awarded marks for the most difficult task completed and had access to the unit tests at home, so may have completed multiple tasks while only submitting the most difficult of these. In addition, due to dependencies in tasks in some weeks, certain completion orders were forced. For example, in some weeks the medium task extended the easy task, so students were required to complete easy before medium. However, the most common strategies according to our data are None (30%), E (31%), EM (11%), EMH (15%), EH (4%), H (6%). The remaining 4% is a mixture of the other combinations with support less than 1%, including some where easier questions attempted later: we saw at least one instance of EHM, ME, MH, HE, HM and HME. Two strategies, MHE and HEM, were not observed at all.

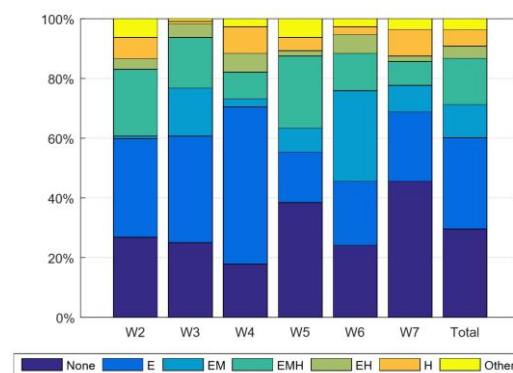


Figure 1. Relative frequency of strategies used by students in each week, and in total across all weeks

4. CLUSTERING

Since students had been allowed to test their code at home, we did not have access to perfect information about the order in which they completed the tasks. We therefore clustered students based

only on the highest difficulty task completed each week, ranking difficulties from 1 (easy) to 3 (hard). E.g., $\langle 3, 2, 3, \dots \rangle$ would represent a student who completed the hard task in Week 2, the medium task in Week 3 and the hard task in Week 4. Using this representation we applied the k -means algorithm with $k=5$ (determined empirically). Cluster centroids are shown in Figure 2.

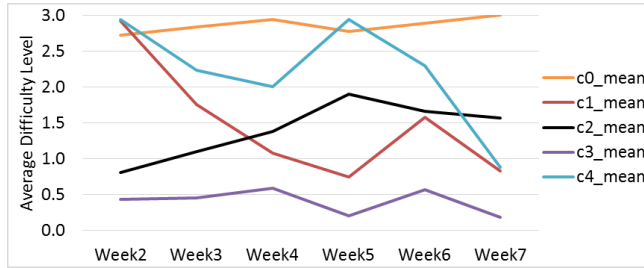


Figure 2. Average highest difficulty of tasks completed by students in each cluster, each week

We complemented the cluster analysis with the information on completion order which involved student submissions with and without potentially missing information. For example, if a student's strategy was EMH, then they definitely completed all three tasks. However, if a student's strategy was H, then they may have only completed the hard task, or they may have completed all three at home and only submitted the hard task. Since strategies with missing information were less frequent, we took the mode weekly strategy for each cluster as shown in 1, which allowed us to still compare student strategies despite the missing information.

Table 1. Mode weekly strategy per cluster. Last column shows proportion of regular and advanced (in parentheses) students.

Cluster	W2	W3	W4	W5	W6	W7	%(adv)
0	EMH	EMH	EMH	EMH	EMH	EMH	9(50)
1	EMH	EM	E	None	EM	None	13(0)
2	E	E	E	E	E	None	18(20)
3	None	None	E	None	None	None	45(15)
4	EMH	EM	E	EMH	EM	E	15(15)

We note that in some weeks there may have been dependencies between tasks that are ignored in this analysis. This limitation notwithstanding, we can broadly summarise behaviour in each cluster. Cluster 0 students complete the hardest task every week, by starting from the easy task and gradually progressing to the hardest task (EMH strategy). Cluster 1 students start well in Week 1 but then gradual drop in the difficulty of the completed tasks towards Week 7. Cluster 2 students start poorly but improve gradually, completing mainly easy tasks. Cluster 3 students consistently make very few submissions, and only of the lowest difficulty. Cluster 4 students generally perform well, often working through tasks of increasing difficulty but not always completing the medium or hard tasks. We speculate that Cluster 3 students may be investing little effort due to the relatively low weighting of the weekly tasks, while Cluster 4 students may have run out of time or found the later tasks too difficult to complete.

5. SLIDING WINDOW RULE MINING

To find trends in changes of strategy we looked for association rules $X \rightarrow Y$ in which X occurred before Y in time, since only these rules are likely to be of use. We further restricted our analysis to periods of three week. We extracted length-3 itemsets

by using a sliding 3-week window over each student's strategy vector. Hence a student's 6-week behaviour vector $\langle 2EMH\ 3EM\ 4E\ 5EMH\ 6EM\ 7E \rangle$ would generate 4 item sets $\langle 1EMH\ 2EM\ 3E \rangle$, $\langle 1EM\ 2E\ 3EMH \rangle$, $\langle 1E\ 2EMH\ 3EM \rangle$, $\langle 1EMH\ 2EM\ 3E \rangle$. This process is similar to rule mining in time-series subsequences [1], but here we encode the time into each item to allow us to use traditional association rule techniques.

Table 2. Highest-confidence rules found using length-3 sliding window rule mining technique

Rule	Support	Confidence	Lift
1None,2None \rightarrow 3None	14%	85%	2.70
1EMH,2EMH \rightarrow 3EMH	5%	62%	4.63
1EMH,2EM \rightarrow 3E	3%	57%	2.00
1None,2E \rightarrow 3E	3%	45%	1.58
1None,2E \rightarrow 3None	3%	45%	1.43

From these item sets ($n = 448$) we searched for rules $1a,2b \rightarrow 3c$ where a , b and c were the strategies used in consecutive weeks. The 5 highest confidence rules are shown in Table 2. The first rule shows that the likelihood of not attempting a task was very high if the student had not submitted two previous tasks. The second two rules suggest a student is likely to work through all three tasks progressively if they did so in the previous two tasks. Most other rules indicate that many students' strategies were on the borderline between completing the task only or none at all. Our technique was limited by task dependencies; we believe its effectiveness could be improved if applied to data without these deficiencies.

6. CONCLUSION

We have demonstrated how clustering can be applied to data from tasks in which students have choices between several activities, with a particular focus on handling missing information. We have also demonstrated how rule mining can elucidate trends in behaviour over a window of time, though the application of this technique was limited by missing information. These techniques were both limited by variability in dependencies in the different tasks, but still demonstrate how useful knowledge can be extracted from such data.

7. ACKNOWLEDGMENTS

This work was funded by the Human-Centred Technology Cluster of the University of Sydney.

8. REFERENCES

- [1] Shokoohi-Yekta, M., Chen, Y., Campana, B., Hu, B., Zakaria, J., Keogh, E., 2015, Discovery of Meaningful Rules in Time Series. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 1085-1094
- [2] INFO3404: Database Systems 2 (2015 - Semester 2). https://cusp.sydney.edu.au/students/view-unit-page/uos_id/125755/vid/309912. Accessed: 2016-05-20.
- [3] INFO3504: Database Systems 2 (Adv) (2015 - Semester 2). https://cusp.sydney.edu.au/students/view-unit-page/uos_id/125790/vid/309914. Accessed: 2016-05-20.
- [4] Radu, A. and Stretton, J. PASTA, School of Information Technologies, University of Sydney, <http://www.it.usyd.edu.au/~bjef8061/pasta/>. Accessed: 2016-05-20