Interacting factors that predict success and failure in a CS1 course

4 authors, including:

Some of the authors of this publication are also working on these related projects:

Time, Eternity and the Afterlife in Antiquity and the early Middle Ages View project

Interacting Factors that Predict Success and Failure in a CS1 Course

Nathan Rountree*, Janet Rountree*, Anthony Robins*, and **Robert Hannah****

Department of Computer Science*
Department of Classics**
University of Otago
Dunedin, New Zealand
{rountree, janet, anthony}@cs.otago.ac.nz
robert.hannah@stonebow.otago.ac.nz

Abstract

The factors that contribute to success and failure in introductory programming courses continue to be a topic of lively debate, with recent conference panels and papers devoted to the subject (e.g. Rountree et al. 2004, Ventura et al., 2004, Gal-Ezer et al., 2003). Most work in this area has concentrated on the ability of single factors (e.g. gender, math background, etc.) to predict success, with the exception of Wilson et al. (2001), which used a general linear model to gauge the effect of combined factors. In Rountree et al. (2002) we presented the results of a survey of our introductory programming class that considered factors (such as student expectations of success, among other things) in isolation. In this paper, we reassess the data from that survey by using a decision tree classifier to identify combinations of factors that interact to predict success or failure more strongly than single, isolated factors.

Keywords: CS1. predictors of success. student background. decision tree, modelling

1. Introduction

An interest in factors which could predict success in a CS1 course was increased in the 1980s due to the rapid growth in popularity of first year programming courses, the varying level of student ability, and the consequent demand placed on faculty resources (Leeper et al. 1982; Barker et al. 1983; Chowdhury et al. 1987). Previously, research interest had largely been focused on occupational aptitude tests—the selection and evaluation of personnel most likely to have a successful and fulfilling career in the new computing industry (Mayer et al. 1968; Cross 1970; Wolfe 1971).

Most studies on predicting achievement in CS1 include math background, gender, programming experience, and previous academic performance as "core" variables for deliberation. Cognitive factors, personality types, and learning styles are also given attention; for example, Piaget's intellectual development levels (Barker et al, 1983; Werth 1986), the Myers-Briggs personality type indicators (Bishop-Clark et al. 1994), and Kolb's learning style inventory (Goold et al. 2000). However, despite the attention given to this topic, a reliable means of predicting the success of students who enter an introductory programming course remains elusive.

There are several factors that make it hard to predict performance, including the sheer number of students who have a wide variety of background skills, differences in levels of motivation, and different expectations of the CS1 course. There is also a relative lack of a CS curriculum at high school level and often a negative student reaction toward the math content of programming courses (Rountree et al. 2004). What makes students succeed (or not) has been of particular interest in large classes with unrestricted entry, as well as programs where previous qualifications are used to determine entry (Gal-Ezer et al. 2003; Boyle 2000).

The most extensive of recent studies predicting success was undertaken by Wilson and Shrock (2001) who developed a model of 12 possible factors including standard variables such as math background and previous programming experience, as well as students' self assessment. Two self assessment factors of particular insight were "comfort level" (questions designed to rate a student's perception of course/programming difficulty and level of anxiety) and "attributions" (questions designed to identify students' belief about their reasons for success or failure—these were ability, ease of task, luck, and effort). Results from this study identified comfort level and math background as having a positive association with success, and student attribution to luck as a negative influence.

The introduction of Java to CS1 has resulted in a change in the content and structure of first year

programming courses. The requirement to teach object-oriented design principles tends to relegate the essential concepts such as selection and iteration into a secondary position in the initial teaching framework. Ventura et al. (2004) noted this trend and investigated whether prior programming experience (a known predictor for success in regard to imperative languages) would remain a strong predictor of success in courses with an "objects first" approach. Findings produced no evidence that prior programming experience correlated with their measures of success. This suggests that, in an "objects first" course (as is the case for CS1 at our university), traditional notions of what is likely to influence success should be reassessed.

2. Background

In Rountree et al. (2002), we investigated the backgrounds and expectations of two years worth of CS1 students. After collecting results from final exams, we tested for association between how the students answered and whether they passed, failed, or achieved a B-grade or better in the course. The survey asked for information on:

- status: gender, age, enrolment status (part or full-time), year of study at university, intended major, how keen they were to take CS1;
- background: what recent mathematics courses they had taken, whether they felt their strongest background was in humanities, science, or commerce subjects, whether they knew any programming language(s) already;
- expectations: how difficult they anticipated the course would be, what they expected of the workload, what grade they expected to achieve, whether they intended to enrol in second-year computer science courses.

Data consisted of answers to a multiple-choice questionnaire, with 472 students responding out of a total of 748. Of the students who completed the course, 66% passed and 39% got a B-grade or better (70% or higher on their final mark).

The data were subjected to contingency table analysis using Chi-square tests with the following results proving the strongest:

- Students who chose to answer the survey were somewhat more likely to pass the course (73% rather than 66%) and a little more likely to get a B or better (45% rather than 39%).
- Students who reported that they were "extremely" keen to do the course had an 85% pass rate.
- Students who reported that they were intending to get an A-grade had a 90% pass rate.

We concluded that the *desire* of students to pick up the skills of programming was the strongest influence on passing the course.

Having identified the most influential factors in isolation, we began to wonder if responses to our survey might be combined to identify students with an unusually high probability of passing or failing the course.

Generalised linear modelling and logistic regression techniques proved ineffective at generating any such combination. However, decision tree analysis identified clusters of students that had a surprisingly high incidence of failure.

Decision trees work by partitioning the data to reduce the diversity of the classes of interest. In this case, each student was labelled "pass" or "fail" and a decision tree induction algorithm used to find which answer could be used to split the data so that one partition contained as many "fails" as possible, and the other as many "passes". This process was then applied recursively to the two partitions until groups could be split no further due to containing only one label.

The resulting model is usually accepted as being "over-fitted", i.e. it is not expected to generalise well. The remedy is to "prune" the tree to the point where it represents a good trade-off between the complexity of the model and its predictive power. The result was a small set of features for students in a given year of study that, taken together, indicated a far higher tendency to fail CS1.

3. Decision tree analysis

To determine if there was any combination of answers that would suggest that a student was more likely to fail COMP103, or more likely to do particularly well, we ran a decision tree inducer over the first year of the survey data, first with pass/fail and then with over 75+/under 75 as class labels. (Although 70% was used as the threshold for "success" in our previous study, this was changed to 75 due to the observation that our students' final marks tend to cluster into three natural groupings, with boundaries at 50% and 75%.) The classification software was the "rpart" library, based on CART (Breiman et. al., 1984), in the R statistical package. Paths from the root of the tree to "fail" leaves can be seen as leading to regions in the possible space of answers that contain an unusually high density of failing students.

```
 1) root 195 68 pass (0.3487179 0.6512821)
 2) grade=B,C,unsure 139 60 pass (0.4316547 0.5683453)
 4) year=second,third 41 16 fail (0.6097561 0.3902439)
 6) background=humanities 7  0 fail (1.0000000 0.0000000) *
 7) background=commerce,science 34 16 fail (0.5294118 0.4705882)
10) keenness=not,very 25  9 fail (0.6400000 0.3600000) *
11) keenness=extremely,neutral 9  2 pass (0.2222222 0.7777778) *
 5) year=first,fourth_plus 98 35 pass (0.3571429 0.6428571)
 8) major=csinf,info,neither 57 25 pass (0.4385965 0.5614035)
12) background=commerce,humanities 30 14 fail (0.5333333 0.4666667)
14) major=csinf,neither 12  3 fail (0.7500000 0.2500000) *
15) major=info 18  7 pass (0.3888889 0.6111111) *
13) background=science 27  9 pass (0.3333333 0.6666667) *
 9) major=cosc 41 10 pass (0.2439024 0.7560976) *
 3) grade=A 56  8 pass (0.1428571 0.8571429) *
```

Figure 1: A decision-tree showing clusters of high fail-rate.

Figure 1 shows an example tree induced on the group of students from the year 2000. The values at each node indicate the number of examples at that node, the number

of examples misclassified by that node, and the predominant class label. The numbers in parentheses give the proportions of each class at that node. An asterisk indicates a leaf node. For example, node 4 contains those students who answered that they were expecting a B, C, or were unsure AND were in their second or third year of study. There were 41 of them, and they predominantly failed, but 16 of them passed. The proportions of fail/pass are 0.61 and 0.39.

Rules to recognise danger zones may be derived from a depth-first traversal of the tree and simplified by removing clauses that are subsumed by later clauses. The rules were then checked against the data from year 2001 and those that did not appear to hold across both years were discarded. This process was repeated with the data from 2001 and checked against the year 2000. The four rules that could be derived on one year but held across both were:

RULE1: not looking for an A AND in second or third year AND background not science:
Year 2000: 21 students, 15 of whom failed. Year 2001: 34 students, 15 of whom failed.

RULE2: background is humanities AND not looking for either an A or a B:
Year 2000: 9 students, 6 of whom failed. Year 2001: 21 students, 12 of whom failed.

RULE3: in third year AND no prior mathematics:
Year 2000: 4 students, 3 of whom failed. Year 2001: 9 students, 5 of whom failed.

RULE4: background is not science AND not in first year AND under 25 years old AND no prior mathematics:
Year 2000: 7 students, 4 of whom failed. Year 2001: 13 students, 10 of whom failed.

Taken together, the four clusters consist of 77 students, of whom 41 failed the paper (53% ± 10.9% instead of the expected 27% of those who answered the survey). We tend to think of the students who respond this way as occupying a "danger-zone" in the overall landscape of survey answers. Of the 8 students who are identified by either 3 or 4 of the rules, 7 failed, including one student who failed the paper in both years.

All four rules highlight the situation of those who pick up CS1 later in their degree programme, although Rule 4 suggests that "mature" students are on safer ground. All four rules also pick up on the way backgrounds in math or science interact with what the students are aiming for and/or what year of study they are in. All four rules also seem more likely to include students who are taking CS1 as a "filler" topic, but do not necessarily intend to develop real skill as a programmer.

In attempting to predict strong success (i.e. a B+ or better, over 75%) in CS1, an identical process identified the following two rules:

RULE1: looking for an A AND not expecting the course to be harder than any other:

Year 2000: 46 students, 6 of whom failed, 13 of whom passed, and 26 of whom got over 75%
Year 2001: 46 students, 2 of whom failed, 10 of whom passed, and 34 of whom got over 75%

RULE2: looking for an A AND age is 16 to 18
Year 2000: 22 students, 0 of whom failed, 11 of whom passed, and 11 of whom got over 75%
Year 2001: 23 students, 0 of whom failed, 3 of whom passed, and 20 of whom got over 75%

These two clusters contain quite a lot of overlap, consisting of 100 students in total, 67 of whom got over 75% (67% ± 9.2% instead of the expected 38% of those who answered the survey).

At this point, it is no surprise to see that "looking for an A" appears in both "success" rules. The first rule seems to suggest that success is associated with personal confidence combined with high expectations. The second rule can be interpreted the same way, in that it captures the youngest possible age-group at the university: those who are confident enough to claim an A grade seem most likely to get one.

4. Conclusion

When the subjects under investigation are people, factors interact: students' expectation of workload differs from first year to fourth year, as do the meaning and depth of their mathematical background. However, studies of students' backgrounds as predictors of success in CS1 have tended to ignore this interaction; it is, after all, extremely difficult to organise large groups of students into the factorial design necessary to capture interactions statistically. Decision trees can capture interacting factors in a natural and effective manner, by recursively partitioning the data based on whichever feature will reduce the diversity of class labels. Because decision tree induction is by nature a "greedy" algorithm, it cannot be guaranteed that the combinations of features are the best possible; nevertheless, the usual result is a set of factor/threshold pairs that identify regions of particularly high density of the class of interest. In our results, we observe regions of students where the failure rate is nearly twice as high as the rest of students who answered the survey.

The decision tree developed from our CS1 data indicated that the "failure" regions are characterised by interactions between academic background, mathematical background, age, year of study, and—most importantly—expectation of a grade other than "A". These regions are most likely to capture students who are transferring from other programmes, or who are taking CS1 as a "filler" to complete their programme. We suggest that course advisors spend some time explaining to these students the level of commitment required of our students and how necessary it is to be aiming for mastery of the skills involved rather than merely aiming to pass. Advisors should also note that confidence alone is not enough: those

people most likely to succeed combine confidence with high expectations of themselves. Alas, there seems to be no substitute for a combination of confidence, high expectation, and youth!

While stronger methods of prediction are of interest for purposes of recruitment, course advising, and restricted entry, it is important to remember that the specificity of such methods is still quite weak; we are recognising "danger zones" rather than "students who will almost certainly fail". However, this may be particularly useful in those situations where departments are under pressure to reduce attrition rates and increase pass rates without diluting the course content. If it is possible to identify students who are "at risk" early in the course, then it may be possible to target specific support to assist them toward success in CS1.

Acknowledgments

References

Barker, R., and Unger, E., (1983), A Predictor for Success in an Introductory Programming Class based upon Abstract Reasoning Development, *ACM SIGCSE Bulletin*, 15(1), 154-158.

Bishop-Clark, C., and Wheeler, D., (1994), The Myers-Briggs Personality Type and its Relationship to Computer Programming, *Journal of Research on Computing in Education*, 26(3), 358-370.

Boyle, R., Carter, J., and Clark, M., (2002), What Makes Them Succeed? Entry, Progression and Graduation in Computer Science, *Journal of Further and Higher Education*, 26(1).

Breiman, L., Friedman, J.,, Olshen, R., and Stone, C., (1984), *Classification and Regression Trees*, Wadsworth International Group.

Chowdhury, A., Van Nelson, C., Fuelling, C., and McCormick, R., (1987), Predicting Success of a Beginning Computer Course Using Logistic Regression, in *Proceedings of the 5th Annual Conference on Computer Science*, 449.

Cross, E., (1970), The Behavioural Styles of Programmers, in *Proceedings of the 85th Annual SIGCPR Conference*, 69-91.

Gal-Ezer, J., Vliner, T., and Zur, E., (2003), Characteristics of Students who Failed (or Succeeded) the Introductory CS Course, work in progress in *Proceedings of the 33rd ASEE/IEEE Frontiers in Education Conference*.

Goold, A., and Rimmer, R., (2000), Factors Affecting Performance in First-year Computing, *ACM SIGCSE Bulletin*, 32(2), 39-43.

Leeper, R., and Silver, J., (1982), Predicting Success in a First Programming Course, *ACM SIGCSE Bulletin*, 14(1), 147-50.

Mayer, D., and Stalnaker, A., (1968), Selection and Evaluation of Computer Personnel – the Research History of SIG/CPR, in *Proceedings of the 23rd ACM National Conference*, 657-670.

Rountree, N., Rountree, J., and Robins, A., (2002), Predictors of Success and Failure in a CS1 Course, *ACM SIGCSE Bulletin*, 34(4), 121-124.

Rountree, N., Vilner, T., Wilson., B., and Boyle, R., (2004), Predictors for Success in Studying CS, panel session in *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education,* 145-146.

Ventura, P., and Ramamurthy, B., (2004), Wanted: CS1 Students. No Experience Required, in *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education*, 240-244.

Werth, L., (1986), Predicting Student Performance in a Beginning Computer Science Class, in *Proceedings of the 17th SIGCSE Technical Symposium on Computer Science Education*, 138-143.

Wilson, B., and Shrock, S., (2001), Contributing to Success in an Introductory Computer Science Course: A Study of Twelve Factors, in *Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education*, 184-188.

Wolfe, J., (1971), Perspectives on Testing for Programming Aptitude, in *Proceedings of the 26th ACM National Conference*, 268-277.