Using a Pre-Assessment Exam to Construct an Effective Concept-Based Genetic Program for Predicting Course Success

Gary D. Boetticher
UHCL
2700 Bay Area Blvd.
Houston, Texas 77058
1 281 283 3805
boetticher@cl.uh.edu

Wei Ding
UHCL
2700 Bay Area Blvd.
Houston, Texas 77058
1 281 283 3871
ding@cl.uh.edu

Charles Moen
UHCL
2700 Bay Area Blvd.
Houston, Texas 77058
1 281 283 3848
moench@cl.uh.edu

Kwok-Bun Yue
UHCL
2700 Bay Area Blvd.
Houston, Texas 77058
1 281 283 3864
yue@cl.uh.edu

ABSTRACT

There is a limit on the amount of time a faculty member may devote to each student. As a consequence, a faculty member must quickly determine which student needs more attention than others throughout a semester. One of the most demanding courses in the CS curriculum is a data structures course. This course has a tendency for high drop rates at our university. A pre-assessment exam is developed for the data structures class in order to provide feedback to both faculty and students. This exam helps students determine how well prepared they are for the course. In order to determine a student's chance of success in this course, a Genetic Program-based experiment is constructed based upon the pre-assessment exam. The result is a model that produces an average accuracy of 79 percent.

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education**]: *Computer science education*, *Information systems education, Self-assessment*.

General Terms

Management, Measurement, Performance, Experimentation

Keywords

Pre-assessment Exam, Concept-based, Machine Learner, Genetic Program, Academic Success Prediction, Course Prediction, Classroom Management, Data Structures

1. INTRODUCTION

Anyone who teaches can appreciate the challenge of assessing students early in a semester in order to identify those students who may need additional attention. Poorly qualified students may not be competent enough to know that they are incompetent, thus overestimating their preparedness for the course. The key question is, ***How to effectively identify high-risk students early in a semester?***

This paper addresses this question by using a twofold process. Students entering the data structures course are required to take a pre-assessment examination where questions are classified into concepts according to the ACM curriculum guidelines on Computer Science programs [2]. The results culled from this exam serve as the basis for constructing a Genetic Program (GP) model for predicting students' grades, with emphasis on the pass/failure results.

An efficient and accurate grade predictor provides many benefits:

- Students can better plan for the targeted course. Weaker students typically overestimate their expected grades and underestimate the necessary effort. This is in agreement with the general study on the premise that incompetent persons may not be competent enough to evaluate their own competency [8]. A feedback in the beginning of the course thus assists these students in managing their study time more efficiently and realistically.
- Instructors can quickly realize what concept deficiencies exist in a class and adjust lectures accordingly.
- Administrators obtain timely feedback on important issues, such as student retention and resource management.

There are many existing experiments on academic performance predictions. Many of them utilize readily available information as predictive factors, such as GPA and SAT. Alternatively, this approach uses an automated Web-based pre-assessment examination, which is taken by the students at the start of the semester. Using a concept-based pre-assessment examination provides additional benefits:

- The examination focuses on problem-solving and critical thinking skills. The questions, which are language independent, are built on essential programming facts and skills conforming to ACM Curriculum Guidelines on Programming Fundamentals [2].
- Feedback to instructors and students can be more targeted. For example, weak responses on given concepts can prompt the instructors to devote more time for reviewing the concepts in the class. Similarly, instead of receiving just

SIGCSE'05, February 23–27, 2005, St. Louis, Missouri, USA.

bleak predictions, lagging students can improve their study habits by committing extra time to their academic deficiencies.

- The examination can easily be integrated to support concept-based outcome assessments, which are preferred by many accreditation organizations.
- The examination can easily be modified to serve as a placement examination, in order to take advantage of its predictive nature [10].

This approach uses Genetic Programs (GPs) to predict course outcome. Traditional approaches often need large sample data sets, which may be difficult to attain over one semester. An important advantage of GPs is that they work well with small data sets and require minimal assumptions [3]. The average enrollment in the data structures class over the last 10 semesters is 91.5 students per semester. Thus, the GP approach fits well in this application domain.

This research uses the results from the pre-assessment exam to determine whether a model can be constructed to predict the probability of success. If this is possible, then the approach can be universally applied to other courses at other universities.

Initial results of the GP experiment for predicting a student's chances of success in the data structures class are encouraging. Four sub-experiments produce an average accuracy of 79 percent in predicting student success.

2. RELATED RESEARCH

Academic performance prediction has captured the interest of educators and researchers for a long time and there are a wide variety of results.

Many models predict broad academic performances, such as GPA, degree completion and earned credit hours [4]. Alternatively, other researchers, including this paper, focus on more specific courses or group of courses [9, 11, and 15].

Various factors have been used to build these predictive models. Important categories of predictive factors include:

- General background of the students, such as race, age and gender [4, 15, and 17]
- General academic background, such as SAT, ACT, GPA, etc. [1, 4, 9, and 17]
- Technical academic background, such as Mathematics background, grades in Computer Science courses, etc. [17]
- Behavioral factors: such as personality, attitude, etc. [9, 15, and 17]
- Internal examination results [10]

This paper differs from other authors in the use of a pre-assessment examination at the beginning of a semester for building a predictive model. This is in contrast to Rosbottom's research where a formative assessment is performed close to the final examination [13], whose primary goal is to help students to prepare for their final examination.

Finally, different educators/researchers use different techniques for constructing the predictors. The majority of the experiments use traditional statistical methods of varying degrees of sophistication [1, 9, 11, and 15]. However, there are also works that are based on other techniques, such as artificial neural network [4].

Compared to others, this paper is distinct in the following aspects:

- A pre-assessment examination is used.
- A GP is used as the basis for building a predictor.
- Examination questions are classified by concepts. Instead of using only the overall examination score, every concept score is used as a factor for constructing the predictive model.
- The predictor model is validated by using a four-fold stratified cross-validation.

To the authors' best knowledge, no other researchers have incorporated a similar approach.

3. APPROACH

3.1 The Pre-assessment Exam

The undergraduate Computer Science (CS) program at the University of Houston-Clear Lake (UHCL) is an ABET-accredited program. It is unique in that it does not offer lower level undergraduate courses. As a consequence, undergraduate students transfer into the program after completing their first two years of study at another academic institution. There is no guarantee of the quality of education at their previous institutions. One of the first classes taken by an undergraduate is a data structures course. This is one of the most critical courses within the computing programs. The data structures course serves as a prerequisite for six other undergraduate CS courses. Doing well in this class is absolutely essential to the successful completion of the undergraduate computing programs. Historically, the drop rate for this course ranged between 20 to 40 percent. Similar problems are shared by other academic institutions. For example, Glasgow University experienced a 42.2% failure rate in their introductory computer science course [12].

To address the high drop rate, a pre-assessment exam has been created for the data structures course. The pre-assessment exam is an hour-long quiz that consists of twenty-nine technical questions and three demographic questions in multiple-choice format.

The technical questions test the student's programming knowledge by asking about programming topics that would have been addressed in the first CS course. The questions were designed by ten faculty members, all of whom had taught the data structures course. The questions were designed to be language-independent, and they uncover variables and assignments, mathematical expressions, conditional control structures, iterative control structures, functions and recursion, parameter passing, arrays, records, and syntax and semantics. All these topics coincide with the core fundamental programming constructs from the Programming Fundamentals area of the ACM's Computing Curricula guidelines for Computer Science [2].

The pre-assessment exam was implemented as an online quiz in WebCT [14], a learning management system that facilitates distance learning. Instructors can use it for developing and teaching online courses, and it supports posting lecture notes, creating discussion boards, and giving Web-based quizzes to assess the students' progress. An important feature of WebCT is its capability to grade and track all quizzes automatically. This feature dramatically reduces the amount of time an instructor needs to commit to monitoring and maintaining a quiz.

The pre-assessment exam was administered to all three sections of data structures during the spring 2004 semester, and each of these three sections had a different instructor. While it is acknowledged that the differences between the three instructors may have affected the modeling process, no obvious effect was observed.

Students could access the pre-assessment exam immediately after the first class of the semester. A slight grade incentive, independent of a student's score, was offered to motivate students to complete the exam.

3.2 Genetic Programs

A Genetic Program (GP) learner is chosen for conducting the experiment for the following reasons:

- It is able to produce a human-readable solution in the form of a polynomial equation.
- It requires minimum human intervention with very little previous domain knowledge.
- Relevant attributes receive greater emphasis.
- GPs tend to scale well with problem size.
- GP modeling does not introduce human bias (quantity/quality of rules) in solution formation.
- GPs allow better abstract representation [7].
- GPs can learn on small data samples.

GPs solve problems by evolving solutions. The GP algorithm consists of the following steps [5, and 6]:

```
1.  Initialize the Population
2.  While a Desired Fitness not reached
    3.  Select Parents        ┐
    4.  Perform Crossover     │  For entire
    5.  Perform Mutation      │  Population
    6.  Evaluate Fitness      ┘
7.  End While
```

Collectively, a group of *chromosomes* (polynomial equations) make up a *population*. All the data is plugged into a chromosome. The calculated results are compared with actual values in order to determine the chromosome's fitness value. In *crossover*, chromosomes are paired up and a subtree equation is chosen and swapped. In *mutation*, a random node is chosen from the new equation tree and that value is slightly modified. See Whitely [16] for further information regarding GPs.

An in-house, vanilla-based GP is used for the experimentation process. This program is available at the following link: http://nas.cl.uh.edu/boetticher/GDB_GP.ZIP. Figure 1 shows a sample screenshot with the GP from one of the sub-experiments.

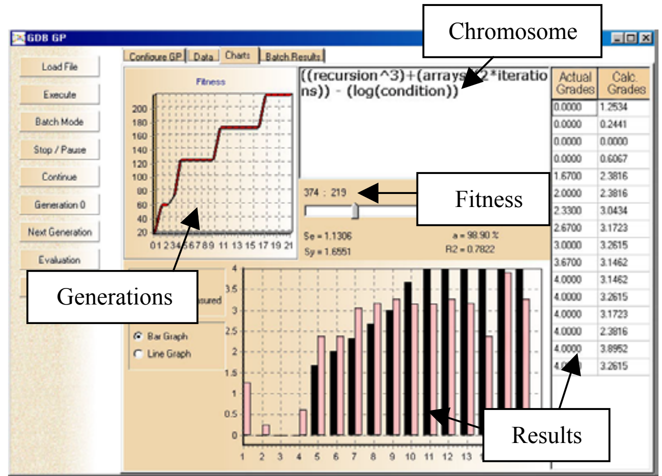

Figure 1: Sample Genetic Program Screenshot

4. EXPERIMENTS

4.1 Experiment Overview

The goal of this experiment is to determine whether a GP can successfully predict whether a student will succeed in a data structures class using the results from a pre-assessment exam.

Sixty-three out of seventy-four data structures students completed the pre-assessment exam. The average score and standard deviation are *21.63* and *4.85* respectively. An *A* is defined as *4.0*, an *A-* is *3.67*, a *B+* is *3.33*, etc. A *WX* grade is assigned a numeric value of *0*. For this course, *Success* is defined is defined as a score of 1.67, which equates to a *C-*, or higher.

A four-fold validation is performed by dividing the initial data set into independent groups. Four separate sub-experiments are performed when each independent group is rotated into the test set. This insures experimental integrity and validates the results.

4.2 Concept-Based GP Experiment

A raw dataset consists of 30 columns. The first 29 columns contain binary values where a "1" means the student answered a question correctly, and a "0" means a student missed the question. The last column is the course grade, which ranged from 0 to 4 as explained earlier.

One difficulty in building models on the raw dataset is that the binary values for the independent variables make it difficult to perform subtle discriminations between questions. Thus, the data was processed by characterizing each of the 29 questions into one of nine concepts which best describes the nature of the question. Table 1 shows the distribution of each question to a corresponding concept.

Mapping questions into concepts offers several advantages. It allows knowledge to be assessed at a higher abstract level, which may be applied to various courses in multiple disciplines. The data now assumes a wider range of values, which makes it easier to differentiate between students.

Table 1. Questions Distributed by Concept

Concept	No. of Questions
Variables and assignment	2
Mathematical expressions	2
Conditional control structures	5
Iterative control structures	5
Functions and recursion	1
Parameter passing	6
Arrays	5
Records	1
Syntax and semantics	2

After identifying a question to concept mapping, the results for each concept are averaged. For example, if 5 questions map to one concept and a student had 4 out of these 5 questions correct, then the student would receive a "0.8" for that concept. Table 2 shows the data layout for this experiment.

Table 2. Data Layout for Concept-Based GP Experiment

	Concept_1	..	Concept_N	Course Grade
Student_N				

The 4-fold stratified cross-validation partitions the data into independent groups of 16, 16, 16, and 15 samples respectively. All 4 sub-experiments use a GP configuration of 1000 chromosomes, 100 generations, and a maximum equation length of 9999 characters.

Figure 2 shows the results from the experiments. For each experiment, the black vertical bar represents the actual grade a student received in the course. The actual grades are sorted from lowest to highest along the *x*-axis. Next to each black bar is a gray (red) bar which shows the predicted grade. The black horizontal line in each graph represents the threshold for success in the data structures class.

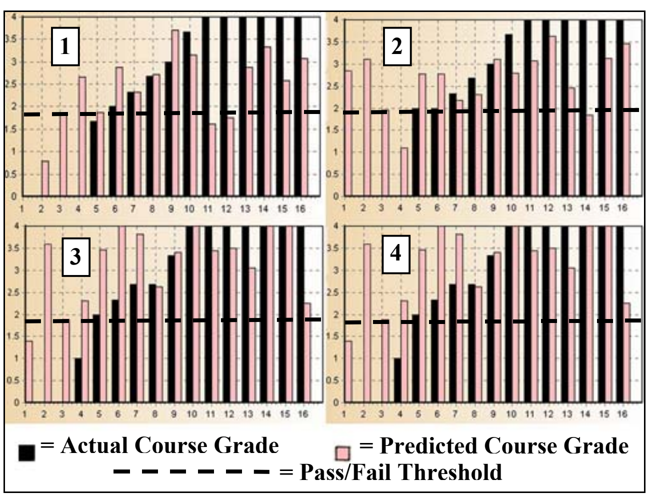

Figure 2: Results from the Concept-Based Experiment

Table 3 shows the results in terms of how well the GP did in predicting whether a student would pass the data structures class. These results show modeling consistency in all experiments.

Table 3. Results from Each Sub-Experiment

	Predicted to Pass		Predicted to Repeat	
Sub. Exp. #	Actually Passed	Had to Repeat	Had to Repeat	Actually Passed
1.	11	2	2	1
2.	12	3	1	0
3.	12	3	1	0
4.	10	3	1	1
Total	**45**	**11**	**5**	**2**

As seen in Table 3, the GP could predict which students would pass the course with 80.3 percent (45/(45+11)) accuracy. This same model could predict which students would need to repeat the course with a 71.4 percent (5/(5+2)) accuracy. Overall, this model is correct for 79 percent (50/63) of the cases.

5. GP VERSUS STATISTICAL MODELS

Various exponential regression models and second order non-linear models are produced using DataFit 8.0. These statistical models use the exact same data configuration (4-fold stratified partition) as the GP. Thus, each model produces 4 sets of results.

The GP and exponential regression models produce results within 1 percent of each other. On average, the GP appears to be superior to the second order non-linear model by 5 percent. However, a t-test assessment between the GP and each of the statistical models does not reveal any superiority of any of the approaches. This is probably due to the limited sample size of 4 experiments per model.

6. DISCUSSION

Considering this experiment was conducted for only one semester and that there are many other contributing factors which influence a student's chances for success, the results are interpreted to be very good. The pre-assessment exam tries to correlate pre-requisite knowledge with the probability that a student will successfully acquire data structures for the remainder of the semester. A major reason for the good results is that the participating faculty brought extensive academic domain knowledge and related teaching experience into the process in terms of what preliminary knowledge is needed to succeed. The exam offers both breadth (nine concept areas) and depth (multiple questions in most areas).

Developing a tool to predict a student's specific grade with 100 percent accuracy is unrealistic and unnecessary. Instead, a major goal of this research is to identify high-risk students. Even when high-risk students are notified, it is their responsibility to determine their course of action. This may require enrolling in one or more foundation courses, improving study habits, or considering another major.

A second major goal seeks to identify concept deficiencies. An automated process aggregates the results by concept then ranks these results by score. This enables an instructor to focus on areas of greatest academic need.

This approach is characterized as bottom-up. It assesses success by course, rather than by program. As a consequence, this approach may be adapted to other courses in other disciplines at other universities.

7. CONCLUSIONS

A process for developing and analyzing a pre-assessment exam is described. Conducting a GP-based experiment predicts success and failure with 80.3 and 71 percent accuracy respectively. These results were comparable and slightly superior to various statistical models. This exam allows students to decide what is necessary to succeed in the course. It also shows an instructor what concept deficiencies exist in a class.

8. FUTURE DIRECTIONS

This whole process is an evolving process. Natural improvements to this research include data, modeling, test, and pedagogical improvements.

- **Data improvements.** Collecting more data samples, especially failure rates (grades of *D+* or lower) would help the GP learn better. It might also be interesting to include more attributes such as SAT score, or High School GPA.
- **Modeling improvements.** Different machine learners could be utilized. Results could be compared/contrasted with the GP results.
- **Test improvements.** A larger test pool may be added which randomly selects questions (by concept). This would allow a student to take the exam multiple times to gauge their progress.
- **Pedagogical improvements.** Adding a corresponding set of tutorials would allow a student to not only identify deficiencies, but to acquire the knowledge or skills that they might be missing.

9. ACKNOWLEDGEMENTS

This work is supported in part by the UHCL Alumni Association Program Endowment, July 2003.

We would also like to thank Dr. Bettayeb, Dr. Davari, Dr. Murphy, Dr. Perez-Davila, Dr. Perkins-Hall, and Dr. Shiau, for their comments and ideas throughout this whole process.

10. REFERENCES

[1] Butcher, D.F., and W. A. Muth, Predicting performance in an introductory computer science course, *Communications of the ACM,* Volume 28, Issue 3, 1985, Pp. 263-268.

[2] CC2001 Joint Task Force. (2001, Dec.). Computing Curricula 2001: Computer Science. ACM/IEEE-CS. [Online].

[3] Chidambaran, N.K., “Genetic Programming with Monte Carlo Simulation for Option Pricing,” *Proceedings of the 2003 Winter Simulation Conference,* December 2003.

[4] Cripps, Al, “Using artificial neural nets to predict academic performance,” *Proceedings of the 1996 ACM Symposium on Applied Computing*, February 1996, Pp. 33-37.

[5] Goldberg, David, *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, USA, 1987.

[6] Grefenstette, John J. *Incorporating Problem Specific Knowledge into Genetic Algorithms.* In L. Davis, editor, *Genetic Algorithms and Simulated Annealing,* Chapter 4, Morgan Kaufmann Publishers, Inc., 1987, Pp. 42-60.

[7] Koza, John. (1992) *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge, Massachusetts.

[8] Kruger J. and David Dunning, Unskilled and Unaware of It: How Difficulties in Recognizing One's Own Incompetence Lead to Inflated Self-Assessments, *Journal of Personality and Social Psychology*, Dec. 1999 Vol. 77, No. 6, Pp. 1121-1134. (http://www.apa.org/journals/psp/psp7761121.html).

[9] Lending, D. and S. E. Kruck, “IT Education: What predicts student performance in the first college-level IS course?: is it different for men and women?” *Proceedings of the 2002 ACM SIGCPR conference on Computer personnel research*, Kristiansand, Norway, May 2002, Pp. 100-102.

[10] Miller, N.E. and Donna S. Reess, A placement examination for computer science II, *ACM SIGCSE Bulletin*, Volume 33 Issue 2, June 2001.

[11] Newsted, P.R., “Grade and ability predictions in an introductory programming course,” *ACM SIGCSE Bulletin*, Volume 7 Issue 2, June 1975.

[12] Roddan, M., The determinants of student failure and attrition in first year computer science.

[13] Rosbottom, John, “Computer managed, open question, open book assessment,” *ACM Proceedings of the 2nd conference on Integrating technology into computer science education*, 1997, Pp. 100-102.

[14] WebCT Campus Edition. [Computer software]. Lynnfield, MA: WebCT, Inc.

[15] Werth, L.H., “Predicting student performance in a beginning computer science class,” *ACM SIGCSE Bulletin, Proceedings of the seventeenth SIGCSE technical symposium on Computer science education*, Volume 18 Issue 1, Feb. 1986.

[16] Whitley, Darrel, “A Genetic Algorithm Tutorial,” *Technical Report CS-93-103*, Department of Computer Science, Colorado State University, November 10, 1993.

[17] Wilson, B.C. and Sharon Shrock, “Contributing to success in an introductory computer science course: a study of twelve factors,” *ACM SIGCSE Bulletin, Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education*, Volume 33 Issue 1, February 2001.