Evaluating Potential Student Progress Indicators for Use in Fostering a Growth Mindset

1st Author

1st author's affiliation  
1st line of address  
2nd line of address  
Telephone number, incl. country code

1st author's E-mail address

2nd Author

2nd author's affiliation  
1st line of address  
2nd line of address  
Telephone number, incl. country code

2nd E-mail

3rd Author

3rd author's affiliation  
1st line of address  
2nd line of address  
Telephone number, incl. country code

3rd E-mail

**ABSTRACT**

For students who are learning to write computer programs, simply measuring whether a program works correctly may not be the ideal way of providing feedback or assessing progress. To keep students engaged and reinforce proper practices, it is helpful to give students feedback along the way as they develop their solutions. However, the correctness of the product as it is being developed does not necessarily give an indication of how much effort the student has put into creating the program, or whether the student is following productive practices as they work.

This paper describes a preliminary study that evaluates a number of potential indicators of student progress that could be applied on partial results as a student develops a solution, rather than waiting for the final product to be completed. The aim is to determine which indicators capture useful information about a student's level of effort or adherence to advisable practices, for the purposes of providing formative feedback as the student works on a solution. These indicators are aimed at reinforcing a "growth mindset"--that is, reinforcing the idea that programming ability improves with practice and effort, rather than being set at a fixed, predetermined level that cannot be changed.

**Keywords**

growth mindset, fixed mindset, indicator measure, automated grading, feedback

# INTRODUCTION

In an era of technology-driven world, computer programming is essential and an integral part in building software. Right from the time when students opt for computer science, they are taught computer programming. Learning to write computer programs is not any impossible task. Applying the right logic to a given problem with the help of utilizing resources (syntax and semantics) the programming language offers, is what it all takes for a student to write a good computer program.

Whenever a student submits a programming assignment, giving feedback to that particular student, regarding the solution, helps student improve on his programming ability. There are generally two kinds of mindsets that a student can possess - fixed mindset and growth mindset. A student with a fixed mindset differs from a student with growth mindset. In any case, students with growth mindset perform better, albeit any task given. When students have growth mindset, they understand that intelligence can be developed. Students focus on improvement instead of worrying about how smart they are. The following picture illustrates the qualities that a student with growth mindset can possess.

# Growth_Mindset_Poster.bmp

Figure : Growth Mindset

# The above figure describes how intelligence can be developed when a student is adhered to a growth mindset.

# Often, students with a fixed mindset tend to be negative in their thinking when they do not achieve what they want to. For example, if a student with a fixed mindset fails in anything, he may think of giving up. But a student with a growth mindset, would learn from failures and improve on his skills to do better.

Whenever we give a feedback for a student regarding his/her programming assignment, it should improve and motivate the student's thinking/approach towards the assignment. Simply giving a feedback about the correctness of the solution would not help the student much. Providing a detailed analysis for a particular programming assignment, would enhance the thinking abilities of the student when they go for the next programming assignment. Overall, making the students realize their potential and intelligence that they can develop helps students make good progress when they deal with different levels of programming tasks.

To enhance such student potentials and to provide detailed analysis, this paper describes a preliminary study that evaluates a number of potential indicators of student progress.

Section 2 describes more in detail about the progress indicators - the different types of indicators and what exactly do these indicators mean. Section 3 talks about the dataset used for computing these indicators. Also the implementation of these indicators is discussed. We then move to Section 4 where we talk about the results obtained from the implementation and the observations we gather. Finally, Section 5 and Section 6 gives a brief description of the conclusion and the future work where this work can be extended to.

# PROGRESS INDICATORS

We define a set of progress indicators to be used for student progress. We define these list of indicators with respect to the detailed analysis to be given to the student regarding the assignment. Every indicator has its own significance. Some students might "hit" particular set of indicators. It does vary from student to student. These indicators have been proposed as mechanisms for evaluating student effort, persistence and proper practices.

There are a set of ten indicators that are considered in this paper. The following list defines and describes these set:

**Increasing Statement Coverage**:

This indicator calculates the increasing number of statements covered in a given program.

**Increasing Method Coverage**:

This indicator calculates the increasing number of methods covered in a given program.

**Increasing Conditional Coverage**:

This indicator calculates the increasing number of conditions covered in a given program.

**Increasing Composite Coverage**:

This indicator calculates the ratio of sum of increasing number of statements, methods and conditions covered in a given program.

**Adding New Solution Method(s)**:

This indicator is the number of non tested methods in a given program

**Adding New Test Method(s)**:

This indicator depends on the number of test methods, if and only if the number of non-test methods does not increase

**Adding to Existing Tests**:

This indicator is dependent on the tested number of non commented lines of code, if and only if the non tested number of non commented lines of code does not increase.

**Increasing Number of Tests per Method**:

This indicator calculates the ratio of number of tested method methods to the number of non tested methods

**Removing Static Analysis Errors**:

This indicator checks the increase in the score of tools used in the program

**Increasing Correctness**:

This indicator checks the ratio of correctness of the score to the number of elements covered in the program.

Using the above indicators, we need determine which indicators capture useful information about a student's level of effort or adherence to advisable practices, for the purposes of providing formative feedback as the student works on a solution.

# DATASET and IMPLEMENTATION

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*Conference’10*, Month 1–2, 2010, City, State, Country.

Copyright 2010 ACM 1-58113-000-0/00/0010 …$15.00.

## Dataset

For calculating the indicators mentioned in the previous section, the dataset taken is a historical dataset collected from a past programming course and consists of about 20368 rows. The data is bundled under different CRN numbers, different Assignments and different study code. Every study code has four different assignments and each study code represents a student with different number of submissions for a particular assignment. A sample dataset of 250 rows is also considered for evaluation purposes.

## Implementation

To implement the indicators, R programming language, JMP statistical software and MS-EXCEL are used. A script is developed in R to handle and compute all the described progress indicators. The script is written in a way that it is reusable and easy to understand.

The flow of the script consists of functions that represent each of the progress indicators. First, a new column "IncrStmtCov" is added to the existing dataset, which computes the first indicator Increased Statement Coverage as discussed. Then a second column "IncrStmtCov.max4" is added, which computed the maximum of previous four values with respect to "IncrStmtCov". After this , a third column "IncrStmtCov.diff" is created which computes the difference between IncrStmtCov.max4 and IncrStmtCov. Finally, a fourth column "IncrStmtCov.bool" is created which checks for the boolean value of IncrStmtCov.diff. If the value is greater than zero, then the value is set as 1, otherwise the value is set as zero.

This way, we compute for each and every other indicator mentioned in Section 2. Once all the indicators are computed, then new dataset is written to an output csv file.

JMP is used for any analysis we want with the results obtained. Since the dataset is considerably big, JMP is one such software to carry out different analysis, or graphs to see any patterns with regards to the progress indicators. Detailed analysis of the sample data taken will be explained in the evaluation section.

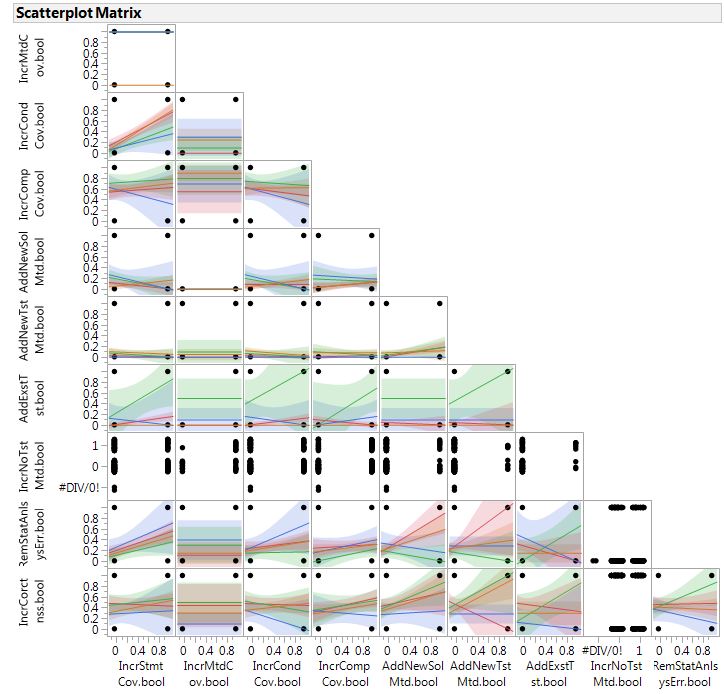
# EVALUATION

To evaluate the student effort and performance through indicators, JMP statistical software is used.

Here, we plot various graphs and charts that could probably give us an insight into various aspects of how the student performed and how much effort the student has put in to achieve the desired result.

As the dataset is large and there are a couple of hundred students, only a sample set of data is chosen for evaluation. The sample dataset contains different study codes, different assignments and different CRNs. For a particular study code, we might want to look into the bool value of every indicator.

We can even consider different cases as to how these values of indicators might be helpful in determining different student factors regarding their effort and performance. The following figure shows a scatter plot with effect of indicators for the assignments.

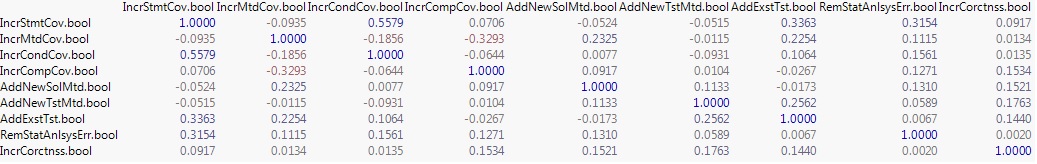


The scatterplot matrix above shows different lines belonging to the assignments of the students. There are certain indicators that students hit the most at the certain value (>0). The first two indicators are the most that students across different assignments are "hit".

This particular matrix evolves from the sample dataset taken. When considered for larger datasets, it allows us to see different patterns for a particular column value, i.e., in this case, how many indicators are hit with most number of assignments. This scatter plot matrix generalizes over student assignments. We can even figure out to a particular student and his/her assignments and see the patterns through it, to determine the number of indicators hit.

Another evaluation would be in the form of a correlation matrix. Correlation matrix determines how these indicators are dependent on each other by using a probability theory to estimate popular parameters.

The following figure shows how the correlation matrix was conceived across different indicators.



The above matrix shows the values on how these indicators are dependent on each other.

The main evaluation would be to see how many indicators a student has hit in order to determine his/her effort. If a student is hitting as many indicators as possible for a certain assignment, then we need to check if there is an increase in the number of indicators for the next assignment. In that case, we might take a call in suggesting that the student has improved his programming ability.

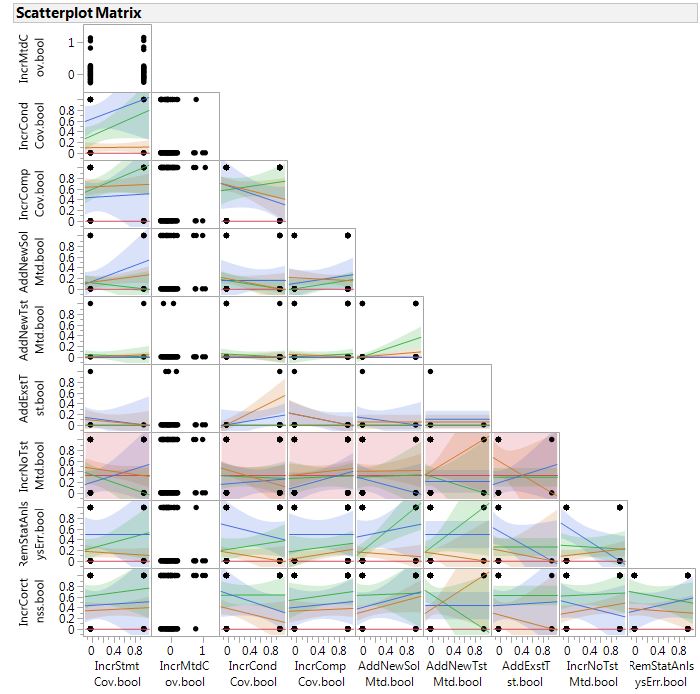
One more thing to consider is, for a particular assignment, to see patterns in the indicators for all the assignments for a particular student.

In the sample data taken, data of one student is considered across all the assignments. Through this, we can see the patterns corresponding to each and every indicator. We can also see specific lines where we can see how many indicators the student has hit.

Judging by the number of indicators hit, we could probably determine the usage of particular corresponding indicator in the effort of the student.

Another way of evaluating the data present at hand, is to plot different data considering the stydycodes for every assignment.

The following figure demonstrates the scatter plot for a particular student where it shows the all the assignments over the indicators. This can help us to see what indicators were actually hit by the student and what indicators were not hit.



From the above figure, we can see that most of the indicators have value greater than zero, meaning most of the indicators were hit by the student. The different color lines specify different assignments. The least hit indicator here is the InCrMethod. This shows that this particular indicator was least hit. But overall, as the student is able to hit most of the indicators, we might see that there is an increase in the efoort of the student.

Similarly, scatter plot matrix tells us for different assignments, how much of the indicators have been utilized. That means, performance in the improvement of the student can be seen. Apart from this we can also consider the final scores of each and every student and can see the different increases. If there is a possible increase, then we can tell that a student is actually improving on the scores based on the indicators.

# CONCLUSION

In this paper, we have conducted a preliminary study of evaluating a number of student progress indicator indicators of student progress. These were applied on partial results taken from a historic data of programming assignments.

The study of indicators was conducted to see the progress of a student in improving his programming capability, to foster his growth mindset.

The evaluation of the indicators can give us an insight into how much effort the student has put in or whether a student is really able to progress well or not even after putting effort.

The evaluation is done a sample data to see different patterns pertaining to the indicators of every assignment for a particular student. We can see that there are cases where certain indicators were hit a lot and also the certain cases where a few indicators were not hit.

From this we can plan to estimate the effort a student has put in terms of programming ability. Evaluations could be done in many ways to see different patterns pertaining to every student. We have considered only a sample data as only a preliminary study is being conducted now. In future, probably efficiency in terms of indicators can be achieved, by considering increasing the number of indicators and also the by taking live data instead of a historic data into consideration.

The main purpose of these progress indicators are aimed at reinforcing or fostering the growth mindset of a student rather than a fixed mindset.

# FUTURE WORK

The future work of this paper can be extended by increasing the number of evaluations in different ways.

One way would be to increase the number of possible student progress indicators. This would help us evaluate better for determining various parameters regarding student's effort. Increasing the number of progress indicators also means that we can test on various student assignment data as well.

One more way is to do a multiple regression prediction equation of the final scores obtained by the student for each assignment. This helps us analyze whether there is increase in the final scores that helps us determine, through the indicators, whether a student has actually increased his performance in writing code, not just the solution.

Performing various statistical analysis on these indicators helps us evaluate the student effort and persistence in writing code.

So far, the entire evaluation was time-independent. We could also utilize the submission times of a student for a particular student, reason being, we can check the number of progress indicators that a student hits within a specific time, helps us determine certain factors related to the students effort.

Apart from this, we can also evaluate the study by considering the indicators that have bool value of most 1s. Then, we can choose the indicators with values of most 0s. By plotting the two sets of values in a graph and to see the number of students with respect to the graph, can make us imprvoise on the usage of the indicators.

Overall, a lot of emphasis can be made from evaluating the progress indicators to check the growth of mindset of a student with regards to computer programming.

# REFERENCES

1. W. N. Venables, D. M. Smith and the R Core Team, An Introduction to R - Notes on R: A Programming Environment for Data Analysis and Graphics, Version 3.2.0 (2015-04-16)

<http://cran.r-project.org/doc/manuals/R-intro.pdf>

1. Interviews with Carol Dweck, mindset works, Carol Dweck

<https://www.mindsetworks.com/webnav/dr-dweck-interviews.aspx>

1. What is Mindset - <http://mindsetonline.com/whatisit/about/>
2. JMP Sofware - <http://www.jmp.com/en_us/software.html>