Developing a software for analyzing data from online learning quizzes.

Salvador Garcia, s1655274

April 10th 2017

Abstract

In educational literature, different methodologies had been developed to assess student's knowledge and comprehension of a topic. One of them consists in the measure of the performance of a student through high-stakes and low-stakes tests. While the first does not give a stimuli for a constant preparation of the provided material, the second requires that the student study regularly the class material. These frequent-basis tests can provide relevant information about the student understanding of a topic. Furthermore, if these are applied with an online learning tool, extra information can be generated to monitor student's learning. This project consist in the development of an R-package and an user interface for the software package *lanalytics*, which make use of the information generated by these online learning tools and give analysis about the monitoring of the students.

1 Background

(As I understand, I should not write a lot of educational literature, so I just write a pair of paragraphs trying to explain what the package will do, Is that correct? Should I be much more specific in the theoretical background of this type of quizzes?)

The high-stakes tests are those that have a huge reward or penalty for the person that takes it. In the context of a class assessment, these are not in a frequent-basis period, so the student and professor have no way to assess the correct comprehension or knowledge of the topics. In contrast, the high frequency low-stakes tests can benefit the class-taker and the class lecturer with a constant feedback for both parts (cite?). A particular example are the individual and the team Readiness Assurance Tests (iRAT and tRAT respectively), proposed in the team-based learning sequence by (Michaelsen, Knight, & Fink, 2002).

The iRAT and tRAT consist in frequent tests that measures the knowledge and understanding of a topic before each class. In order to obtain the reward (or to obtain no penalty), the students require the comprehension of the topic before class, giving the opportunity that educators detect which concepts and topics should be deeply covered in class. This is call *Just in Time Teaching (JiTT)* , which allows to improve the time efficiency during the class and tries to maximize the student's learning given the constrained class time (cite).

Currently, there are learning software that facilitates the monitoring of low-stakes tests [1], but there is a lack of open-source analysis tools. To address this, a software package called *lanalytics* was created by Melanie I. Stefan (cite?). This package performs relevant analysis using the timestamps, the easiness (as the ratio between correct and total number of questions) and the cognitive level of the each item (measured by the professors into three categories: factual knowledge or terminology, understanding of concept, comprehension of concept).

In particular, the *lanalytics* software package introduces the TEL (Time-Easiness-Level) plots, that combines the Item Response Time with the easiness and cognitive level in one single plot. As stated in (cite), it can be useful for a visual analysis to improve the items quality in future tests.

(The IRT paper draft talks about TEL plots, but in the code of lanalytics there are more codes that will be used. Should I describe the current content of lanalytics? Should I propose more analysis methods in this proposal?)

2 Purpose

The purpose of this project is to publish an open source analysis R-package and interface based on the *lanalytics* software package. This can help to improve the item quality for each course and keep track for different year comparisons. The package and interface could be used by a large community of people that are interested in low-stake tests and the associated benefits of it.

3 Methods

(The IRT paper draft talks about TEL plots, but in the code of lanalytics there are more codes that will be used. Should I describe the current content of lanalytics? Should I propose more analysis methods in this proposal?)

Besides the analysis of the answers, an important factor in the quiz is the quality of the items. For this reason, the project will take as basis the methodology proposed for Melanie I. Stefan (cite?):

[1] For example, the learning catalytics software (cite).

- First, each item of the quiz is classified according to one of the three predefined cognitive level: Factual knowledge (F), understanding of concept (C) or application of concept (A)).
- Second, the easiness for each item is computer as the total number of student that answer correctly divided by the total number of students.
- Third, the Item Response Time (IRT) is computed as the median of the all the times for that answer.
- Forth, for each cognitive level, an scatter plot will be generated. Each point will represent an item and the color the cognitive level. In the x-axis the IRT will be placed and in the y-axis the easiness.

This is called the Time-Easiness-Level (TEL) plot and it helps to divide in four quadrants all the questions:

- easy items and slow IRT (Quadrant 1)
- easy items and fast IRT (Quadrant 2)
- difficult items and fast IRT (Quadrant 3)
- difficult items and slow IRT (Quadrant 4)

Each of these quadrants can have a different intuitive interpretation according to the cognitive level of it. For example, as stated in (cite), easy items that just asks factual knowledge should be in quadrant 2 because if the question is easy, then it should be answered immediately. Similar analysis can be done for each quadrant considering the cognitive level of the question.

4 Evaluation

The input files for this package should work with outputs from learning catalytics, as well as a general input format in a text file. Right now, all of these functions were created by Dr. Melanie I. Stefan, and are available in the existing repository *lanalytics*. For this project, these methodologies along others will be published as an R-package and an user interface will be created.

To evaluate the functions, two datasets will be used. The first includes 29 low-stakes open book quizzes from the molecular biology course at Harvard Medical School (cite?), while the second is a dataset from the Queen Margaret University in Edinburgh (cite?). These datasets had been..

(As far as I understand, the datasets had been anonymity, so we have de-identified data. Should I describe more about the datasets, or it is OK just to mention them?)

Besides that, synthetic dataset should be created to show the functionality and capability of each function, and to help the final user to use this package and interface.

5 Outputs

The projected outcomes for this project are three:

- First, an open-source R-package will be published with the corresponding vignettes and documentation.
- Second, a github page explaining the uses of this analysis tool will be created with Jekyll.
- And third, an user interface created with Shiny-apps will be included.

The proposed technologies seems suitable for the purpose of the project, although others will be considered.

6 Work plan

Table 1: Work plan for dissertation

Week	Dates	Work to do
Week 1	May 26th - June 1st	- Update codes to latest package versions, consider carefully package dependencies. - Read about license types and data protection law. - Create the general structure of the package. What R-object will be created? - Think about the arguments and outputs of functions. - Think about the general format of the input.
Week 2	June 2nd - June 8th	- Look for current requirements for package publication and documentation. - Make a draft of the structure of the R-vignettes and the reference manual. - Search for possible conflicts with other name-spaces or packages. - Research about Jekyll technology to add a page to the repository.
Week 3/4	June 9th - June 22th	- Implement points in the to-do list of the repository. - Generalize methods to work with different years
Week 5	June 23th - June 29th	- Make first version of package. - Make first version of Github page for the repository.
Week 6/7	June 30th - July 13th	- Test usability of the package with friends. - Create synthetic datasets for the package. - Make first draft of R-vignettes and package documentation.
Week 8/9	July 14th - July 27th	- Work on additional features for the package
Week 10	July 28th - August 3rd	- Write dissertation paper
Week 11	August 4th - August 10th	- Final draft of the R-vignettes and package documentation - Final version of Github page for repository and Shiny-app deployment. - Final version of dissertation
Week 12	August 11th - August 18th	- Check and test R-package - Check final R-vignettes and package documentation - Check final github page and Shiny-app deployment - Check final dissertation report

References

Michaelsen, L. K., Knight, A. B., & Fink, L. D. (2002). *Team-based learning: A transformative use of small groups*. Greenwood publishing group.