Developing a software for analyzing data from online learning quizzes.

Salvador Garcia, s1655274
s1655274@sms.ed.ac.uk

April 14th 2017

Introduction

In educational literature different methodologies had been developed to assess student's knowledge and comprehension of a topic. One of them consists in the measurement of the performance of a student through high-stakes and low-stakes tests. While the first does not give a stimulus 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 consists in the development of an R-package and an user interface based in the software package *lanalytics*, which make use of the information generated by online learning tools and give analysis about the test performance of the students.

Background

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 frequent test; in consequence, the student and the professor have no way to assess the correct knowledge and comprehension of the evaluated topics. On the other hand, the high frequency low-stakes tests can benefit the class-taker and the class-lecturer with a constant feedback for both parts. Some particular examples 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 are frequent tests that can measure the knowledge, understanding and comprehension of concepts and topics before each class. Although these test can be applied separately, the *team-based learning sequence* (Michaelsen et al., 2002) propose that the individual RATs should be made first by students, after that, the team RAT should be made to compare answers and receive feedback from their schoolfellows. The educators should determine the level of preparation required for each class and also the rewards given by the tests. With the obtained answers from students, the professor can be aware of misconceptions and difficult topics for the students; in consequence, he can adapt the class to address this problems. This method is called Just in Time Teaching (JiTT) (Novak, 2011).

By analyzing the results of the frequent-basis low-stakes tests and by performing Just in Time Teaching, the educators can obtain insights about the general and the specific performance of the group with respect to each topic. Then they can take a decision about the distribution of the time in the class, choosing between topics that are well understood and topics where there is a general lack of knowledge or understanding. With this relevant information, the educator can try to maximize the student's learning in the constrained amount of class time.

Currently, there are tools that facilitates the monitoring of the answers in low-stakes tests [1], but there is a lack of open-source analysis tools. To address the situation, a software package called *lanalytics* was created by Melanie I. Stefan (Stefan & Liu, 2015). This package performs relevant analysis using the answers given by students and the timestamps of each one, as well as the item easiness (as the ratio between correct and total number of questions) and the item cognitive level (measured by the professors into three categories: factual knowledge or terminology, understanding of concept, comprehension of concept). In addition, some more general ideas had been proposed, as the *Open Learning Analytics* that is a platform that consist in a dashboard and engines that perform analysis about students, researchers, educators and institutions (Siemens et al., 2011).

Purpose

The purpose of this project is to publish an open source analysis R-package and an interface based on the *lanalytics* software package and enhance it with more detailed analysis. This can help to improve the item and test quality for each course and compare different years results. 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.

Methods

The R-package and user interface that will be created should accomplish two main objectives. The first is that it should sufficiently versatile for R-users and Rdevelopers, while the second is that it should be sufficiently user-friendly for people with no background in programming.

To make an R-package we should be sure that the code is reproducible. To do that, R has to follow a package structure and some standards related to the metadata, the namespaces, the vignettes, the documentation and the sample data. These rules help to make a package reproducible, user-friendly

[1] For example, the learning catalytics software (Schell, Lukoff, & Mazur, 2013).

and set the basics for future developments. In this section a brief description of these standards will be given (Wickham, 2015):

- Metadata (`DESCRIPTION`) : Is a general description file, including package name, author, version, package imports (and versions), package dependencies (and versions), license and description.
- Namespaces (`NAMESPACE`): In general, the specification of this file is to avoid conflicts with other packages names.
- Documentation (`man/`): It should be available through the `help()` function and contains the general description and usage of each function of the package.
- Vignettes (`vignettes/`): It describes the purpose of the package and how it uses the functions to achieve it. It also contains examples of usage.
- Sample data (`data/`): It contains usable data to test the functions and see how it works.

The main analysis performed by the *lanalytics* package are represented with plots. As mentioned before, the usability of the package is one of the main purposes, therefore the grammar of graphics will be used to create each one. This framework has many advantages: it builds layered graphics, so the title, annotations, axis, labels, theme and colors are entirely customizable by the user (even after it is created). In R, this can be accomplished with the `ggplot2` package (Wickham, 2009).

The user interface will be done as a Shiny app, a web application framework for R (RStudio, Inc, 2013). Shiny has the big advantage that is written in R code and is open-source, so it will have full compatibility with the complementary R-package. Technically, a Shiny app consist in two files that correspond to the backend and the frontend components. The backend is written in a file called *server.R*, where all the analysis codes will be run, while the frontend is a file called *ui.R* that correspond to the user interface interactive component.

Besides the technical component of the implementation and deployment of the package, its content should be analyzed. As said before, the methods that will be included in the R-package are based on the *lanalytics* package, in addition with additional features that will help in the student's assessment. The functions included in this software package analyze the relationships between variables and variable transformations. This way, relevant analysis about the invested time to answer each item, the total invested time for the test, the order of item answering, the easiness of the questions or even the cognitive level of each item can be made in order to obtain valuable information. Some of the current implemented capabilities are:

- Compute invested time per question. Computes the time that each student spent in each question.
- Item Response Time calculation. Finds the median time that students spent to answer an item.

- Compute total time invested in the test.
- Detect possible incidences of guessing and cheating.
- Calculates item discrimination.
- Computes the total scores per student.
- TEL (Time-Easiness-Level) plots that analyses the Item Response Time with the easiness and cognitive level in one single plot.
- Order of answering of the items.

The *lanalytics* package can be extended for more analysis. Depending on the insights that will be found on the datasets, new features and capabilities will be added. Some other ideas for analysis that can be tried are (among others):

- Find the relationship between time to deadline and invested time on the quiz
- Find the relationship between time to deadline and total score per student
- Analyze the student progression through the semester (The student is performing above or below average?).
- Track the improvement of items in the quizzes and compare it with the performance of students in different generations.
- Compare student performance in the low-stakes tests and the student performance in the final high-stakes test. Is there a relevant relationship between them? (if information is available).

Evaluation

At the end, the package should work with outputs from the learning catalytics software (Schell et al., 2013), also with general text files that contains the same information in a text file. In addition, the codes should manage several date and hour formats. Finally, the documentation, vignettes and Github page should contain clear instructions and the motivations for the package.

To test the correctness and performance of the R-package and the Shiny-app, they will be tested with two real-world datasets. The first will be a 29 low-stakes open book quizzes from the Molecular and Cell Biology course at Harvard Medical School (2013, 2014), while the second is from the Cell Biology course at Queen Margaret University (2017). The datasets had been anonymized. All the functions will be tested and will be used as an example to produce several synthetic datasets. These will be created to show the functionality and capability of each function, as well as the necessary code for each one. Finally, the shiny-app should have full integrability with the R-package in order that the posterior developments does not need major changes in the appearance of the shiny-app.

Outputs

The main outcomes for this project are three:

- An open-source R-package that will be published with the corresponding vignettes, documentation and synthetic datasets.
- A Shiny-app that given a text file as an input, it show the selected graphs analysis.
- A Github page explaining the uses of this analysis tool and that host the open source repository.

The proposed technologies and implementations seems suitable for the purpose of the project, as they provide reproducible, usable and easy to download software. This is relevant because it will provide automatic analysis and by doing so it allows to improve the workflow of the analysis tools for educators interested in this topic.

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.

Novak, G. M. (2011). Just-in-time teaching. *New Directions for Teaching and Learning*, *2011*(128), 63–73. Retrieved from `http://dx.doi.org/10.1002/tl.469` doi: 10.1002/tl.469

RStudio, Inc. (2013). Easy web applications in r. [Computer software manual]. (URL: `http://www.rstudio.com/shiny/`)

Schell, J., Lukoff, B., & Mazur, E. (2013). Increasing student engagement and retention using classroom technologies: Classroom response systems and mediated discourse technologies. In C. Wankel (Ed.), (pp. 233–261). Emerald.

Siemens, G., Gasevic, D., Haythornthwaite, C., Dawson, S., Shum, S. B., Ferguson, R., ... Verbert, K. (2011). *Open learning analytics: an integrated & modularized platform* (Unpublished doctoral dissertation).

Stefan, M. I., & Liu, Y. (2015). lanalytics. *GitHub repository*. GitHub repository. Retrieved from `https://github.com/MelanieIStefan/lanalytics`

Wickham, H. (2009). *ggplot2: Elegant graphics for data analysis.* Springer-Verlag New York. Retrieved from `http://ggplot2.org`

Wickham, H. (2015). *R packages.* " O'Reilly Media, Inc.".