Lista de exercícios sobre filtros de software

Cristiano Silva Júnior

23 de Junho de 2017

1 Filtro de média

1.1 Mostrar a derivação da forma recursiva.

A partir da definição da média

$$\bar{x_k} = \frac{1}{k} \sum_{i=1}^{k} x_i$$

Podemos fazer a seguinte manipulação

$$\bar{x_k} = \left(\frac{1}{k-1} \sum_{i=1}^{k-1} x_i\right) \frac{k-1}{k-1} + \frac{x_k}{k}$$

$$\bar{x_k} = \left(\frac{1}{k-1} \sum_{i=1}^{k-1} x_i\right) \frac{k-1}{k} + \frac{x_k}{k}$$

Como

$$\bar{x_{k-1}} = \frac{1}{k-1} \sum_{i=1}^{k-1} x_i$$

Então

$$\bar{x_k} = \bar{x_{k-1}} \frac{k-1}{k} + \frac{x_k}{k}$$

2 Methods

Developed the C# programming language and available for free under the MIT license, Scalemate can be downloaded from GitHub, and everyone is invited to contribute to it.

The software is divided into two parts: a library and the graphical user interface. The library holds the logic of the engine, mostly implemented on a Tester class. Separating the library from the rest of the program enables the tester to implement Scalemate however they wish to. Two implementations are provided, though: one based on the Windows Forms technology, and another on the Telegram messaging application.

2.1 Library

The library consist of a DLL file containing the logic of the test, and two folders: *assets* and *results*.

The *assets* folder is the place to store the tests per se. There is a file called "kinds.txt" to list every test that must be shown in the program. Each line of this file must follow the following organization:

- The first word is the the name of the folder containing a test.

- The following words are the test label, and will appear on the main screen for easier selection.

For each line in the "kinds.txt" file, there must a corresponding folder, containing the needed information for the test.

For each test, there must be at least two files inside its folder:

- A "inventory.txt" file containing the questions and answers for the test. The first line in this file must be the number N of possible answers in the test. The following lines must be organized in groups of N+1 lines: the first one is the question that will appear on the top of the test screen, while the following N lines are the possible answers. They must be placed according to their respective scoring: the first answer is worth 0 points, the second answer, 1 point; the third answer, 2 points; and so forth. If the researcher intends to revert this scoring logic, they can add a '*' character before the question sentence. The application will understand this as a command and this character will not appear in the questionnaire but will influence its final result.

- A "results.txt" file which will describe how to score this test. Each line must contain three itens: the first one is a number, representing the lowerbound of this specific result; the second item is another number, defining the upperbound; and the rest of the line is the result to be written in the test result. These must separated by space characters. Scalemate will score a test application by summing the subject's points and checking if their score is equal or greater than the lowerbound and less than the upperbound. When that is the case, the engine will consider that as the correct result. When designing a test, the researcher must be sure that there is no overlapping between each result boundary and that every possible score is described in this file.

There are two other files that can appear inside the test's folder, but they are optional. They serve to make the test's application easier. These files are:

- A "information.txt" file with a list of questions to be asked for before the test begins, so the researcher can survey some other information about the subject before the assessment actually starts.

- A "instructions.txt" file with a text to appear before the test's beginning to instruct the subject on how to proceed or to give whatever information the researcher considers important or useful for that procedure. If omitted, Scalemate will go straight to the test's first question.

The *results* folder on Scalemate is used to store the users' results, as saved and processed by the engine. After each test completion, the application will generate a tab-separated values file, unbundling the subject's information, their result on the test and each given answer. Each file is named as {*subject name*}_{*test folder*}.*csv* and is organized as follows:

- The first line contains the information needed to identify the subject. It reveals the test's folder name, the subject name, their score and the result given by Scalemate, in this order.

- If the "information.txt" file is present, then the following line contain the answers to the initial survey.

- The following line has the answers to each question on the test, for later analysis. Each answer represents the index minus one of the selected item, that is, if the user chose the first option, then the number 0 will be stored to represent that answer; if they chose the second option, then the number 1 will appear, and so forth.

2.2 Graphical User Interface

Two implementations for the GUI were made. The first one was written with the Windows Forms technology from Microsoft Visual Studio. It consists of a compressed folder which, once extracted, contains an executable, made to run on computers running Microsoft Windows Vista onwards, and the library structure. This version aims to help those that perform simple question-and-answer based research on places where internet is not readily available. This is an lightweight application, consuming as much as 20MB of RAM memory, depending on how big the tests are.

The second one is a Telegram bot that can be run on any Windows® computer available as a server. This version is directed towards testers that want to reach bigger populations using an open source smartphone application.

3 Results

In order to demonstrate Scalemate's data collection abilities, we conducted a small research based on a paper by Clarry Lay [1] on procastination. We applied his questionnaire using this program and developed a small MATLAB application to analyze the generated data. We applied his test on students from the University of Brasilia from many differents graduation programs and backgrounds to find a correlation between their average academic performance and their scores on Lay's scale. We chose the Lay's scale due to its simplicity of application: it fits Scalemate's test model really well and the collected data can easily be assessed from the students. The results are summarized on figure 1.

The research was conducted using the Telegram bot implementation of the Scalemate engine from March 8th to March 15th in the year of 2017 on students from the University of Brasilia, and collected surveys from 33 students, whose 20 were rended valid. These students came from the Electrical Engineering, Control and Automation Engineering, Computer Science, and Psychology graduation programs.

A small MATLAB® script was written to process the data and generate the output plot. Using MATLAB's **corrcoef** function, we were able to calculate the correlation coefficient between the students' scale results and academic performance. From the collected data, we got a correlation coefficient $C = 0.251390$, indicating that procastication and academic performance are not correlated, thus repeating Lay's thesis on his paper.

4 Future Works

Just as it is available, Scalemate can only be used for simple tests, and still require the research to open and deal with many files. Therefore the most obvious steps are:

- Enable multimedia content inside the tests, like images and sounds. This will expand the number of fields in which Scalemate can be used;

- Create a user-friendly test editor whithin the program, so the researcher does not need to leave the application to deal with test data or text files.

Another possible improvement can be done on the way Scalemate process the collected data. Right now, it just uses a simple score based grade system, whereas some other more complex analysis could be done on the collected information. Even though Scalemate's scope covers mostly how the researcher will apply their tests, it can also come really handy to have an inside tool to help with further data mining and processing.

5 Bibliography

References

[1] Clarry Lay, At last my research article on procrastination, Journal of research in personality 20, 1986.