Mining educational social networks for student performance classification

Andreu Grimalt Reynes

Abstract

This document is a preliminary report on my final project in computing. The project is concerned with the analysis of log data generated by an online learning platform in order to extract knowledge about student performance. The first section of this report contains a general description of the project as well as an overview of the techniques used. It is followed by a description of the aims and objectives. The next sections discuss the methods employed, the project plan, the progress to date and the planned work. Finally, there is an appendix which contains software code.

Introduction

The growth in the use of VLEs (Virtual learning environments), MOOCs (Massive Online Open Courses) and other online education systems provides an increasing stream of data regarding student behaviour. This new paradigm makes it possible to use data mining algorithms to analyse student habits and try to find correlations between student behaviour and academic performance. Teachers and researchers have begun to explore the possibility of using social media in teaching and learning [10]. In general, the use of social networks generate data sets that accurately describe their users behaviour. When social media is used for learning, the data set generated can be used to parametrise the learning experience and therefore to optimise it. This enables us to think about social networks, education and student behaviour as a classification problem. That is to say, to take the data generated in social media and apply data mining algorithms to it in order to classify students in groups according to their academic performance. Once the algorithms are trained in recognising these groups, they can predict future performance and allow the optimisation of the learning experience.

The data used on this project was generated in a case study involving Music Circle, a social network for sharing and discussing media [4]. Music Circle makes it easy to share media with a community of users and it provides tools that encourage feedback and the discussion of work in progress. Therefore, being able to give specific feedback on the media item (i.e being able to select particular bits on the media and attach comments to them) are key features. Music Circle was developed within the PRAISE research project at the computing department at Goldsmiths [5].

Aims and objectives

Aim

The aim of this project is to use the activity patterns of Music Circle users in order to classify them by their academic performance.

The activity pattern of a user is fully described by mouse events which are logged into the Music Circle database. The log data taken into account is the response after feedback, that is to say what the user does after receiving feedback from other students in the system. The actions a user can perform on the system are the following: Viewing a media item, reading or writing an annotation on a media item and reading or writing a reply on an annotation. These actions can be deduced from the log data which has the following structure:

- user id: The id of the user that performed the action
- time: The time where the user performed the action in milliseconds since the epoch time
- type: The type of mouse action (click, mouse over, mouse out)
- id: The id of the element the action was performed on
- logtype: A custom name used to identify the element the action was performed on

The categories in which the students are classified are simply "successful students" and "struggling students". This implies the assumption that good students have a different behaviour from struggling students and that this difference is represented in the log data. Hopefully, the research will show if this assumption is valid.

Objectives

Setting up the development environment: Virtual Environment, Python scientific tools, MongoDB and PyMongo

The programming language used in this project is Python. The main reason to use Python is that it has a rich library ecosystem in scientific programming and machine learning. It provides tools for doing mathematical operations and plotting as well as a friendly IDE (iPython notebooks), all conveniently sandboxed in a virtual environment.

Music Circle data is stored in a Mongo database. The data is processed in Python using PyMongo which provide the tools to work with MongoDB databases.

Build a research database

The data from Music Circle is stored in a way that makes it efficient for the server to retrieve it. Unfortunately, this way of storing the data is not the most efficient in order to analyse it. Therefore, a database with a new structure needs to be built.

Feature vector construction from the log data

The feature vector describes the behaviour of a particular student at a particular interval of time. There are many ways to construct such a vector but one must take into account that not all possible

feature vector representations will be useful for the classification problem. Therefore, the selection of a suitable feature vector is critical for the success of the project.

Clustering of the feature vectors

Once the feature vectors are constructed they need to be validated. One needs to demonstrate that the vectors describe user behaviours and that they can be used for classification. In order to do this, the first approach is to try to cluster a set of feature vectors. If the vectors represent the behaviour of users correctly and that behaviour is consistent through tscalingime, then they will cluster in n clusters where n is the number of users.

Feature reduction

Not all the components of the feature vectors will contribute the same amount to differentiate users. The objective at this stage is to find which components of the vectors contribute the most to differentiate between users and discard the components that are not relevant.

Statistical distribution comparison

Once the principal components are found, one can construct a statistical distribution of the feature vectors for each user and then compare those distributions. The hypothesis is that students from different categories will have different statistical distributions, by associating the distributions to the ground truth data one should be able to classify the students.

The ground truth data consists of the grades from quizzes and peer assessments that students completed during the case study.

Methods

This section describes the methods used to implement the objectives.

Virtual Environment, Python scientific tools, MongoDB and PyMongo

The software stack used in this project consists in Virtual Environment, iPython notebook, SciPy, PyMongo and Scikit-learn.

Virtual Environment is a tool to keep the dependencies required by different projects in separate places, by creating virtual Python environments for them [9]. iPython notebooks are a web-based interactive computational environment where you can combine code execution, text, mathematics, plots and rich media into a single document [2]. SciPy is a Python-based ecosystem of open-source software for mathematics, science, and engineering [8]. PyMongo is a Python distribution containing tools for working with MongoDB [6]. Finally, Scikit-learn is a machine learning library in Python [7].

Build a research database

The Music Circle database is approximately 19Gb in size with some collections containing an order of 10^6 elements. In order to process the data in a reasonable amount of time, one needs to program in a way that maximises the CPU use.

MongoDB response to large database processing is sharding [3]. Sharding main idea is to split the database across different machines or processor cores in order to allow scaling as well as parallel computing. Such a system has the name of sharded cluster.

A sharded cluster has three main components: Shards, configuration servers and routers.
Shards contain the data in the database (each shard contains a subset of the data in the database). The configuration servers contain metadata about the cluster which map to data in the shards. Finally, the routers route the reads and writes to the shards.

There are some parameters that need to be chosen carefully in order to maximise the benefits of sharding. These parameters are: Number of shards, number of routers and chunk size.
Ideally, the number of shards should match the number of CPUs available. This way MongoDB can access the database using more than one CPU thus allowing parallel computing. Technically only one router is needed, however, in real world scenarios it is recommended to use at least two routers in the case one fails. The size of the chunks will determine how the data gets distributed in the shards, one must aim for an equal distribution of data between the shards.

Feature vector selection

The feature vector selection is crucial for the success of the project. In general, the feature vectors will be determined by the log data available. At the same time, there are many ways of combining the log data in order to construct the vectors, this means that once built the feature vectors need to be validated in order to demonstrate that they characterise correctly user behaviour. These validation process consists in clustering the vectors. To do so, a set of feature vectors corresponding to different users and different periods of time is selected. If these vectors successfully represent users' behaviour and this behaviour is consistent over time, the vectors corresponding to each user should cluster together. In ideal conditions, the clustering process would produce n clusters where n is the number of users.

Feature reduction and statistical comparison

The previous step will hopefully find a set of users with consistent behaviour through time. Some of the components of the feature vector will contribute more than others to differentiate between user behaviour.
Feature reduction techniques allow to find the principal components of the feature vector. This dimensional reduction process is necessary in order to ease the analysis and avoid redundant information that could contribute to false results [11].

In such scenario, one can build a statistical distribution of the feature vectors for each user and then compare the distributions using Kullback-Leibler divergence [12]. Making the hypothesis that the statistical distribution of feature vectors between successful and non successful students are different and linking the distributions to the ground truth data, this method should classify students in binary categories of performance. With this last step the project can be validated by trying to cluster different distributions. The number of clusters should be equal to two with one cluster representing successful students and the other representing struggling students. In order to evaluate the project, the clustering can be compared to the ground truth data and measure how accurate the classification is.

Project plan

The software development method used in this project is agile development [1]. It consists in iterative implementations, face-to-face communication between members of the team, short feedback loop and quality focus.

The project is divided into the objectives described in the "Aims and Objectives" section. Each of the objectives is further divided into small incremental tasks which are set at a meetings held weekly. Every week a set of tasks have to be implemented and on the following meeting those implemented tasks are reviewed.
Supposing that the implementation has been successful a new set of tasks is set. Otherwise, the failures on the implementation are analysed and solutions are proposed.

There is not a long term planning of the project but a succession of incremental and adaptive iterations over short time frames which get set in weekly meetings.

Progress to date

The progress up to date relates to setting up the development environment, building the research database and finding a suitable feature vector.

Most of the time was spent in building the research database. Because of the size of the Music Circle database, finding a method to import the log data efficiently (described in "Build a research database" section) was essential.
The research database contains two collections: Actors and Media.
Each element in the Actors collection contains all the activities of a particular user on the system. The Media collection contains data about the media items uploaded to the system.

This is the schema of the research database:

```
Actors: [
    {
        _id: element id,
        activities: [
            {
                at: activity type
                id: id of the element the action took place on,
                ts: timestamp
            }
        ],
        idx: user id on the Music Circle database,
        name: name oft heuser,

    }
],
```

Where the possible values for 'at' are: 0=upload, 1=view, 2=comment, 3=reply, 4=login, 5=play, 6=log data (mouse events).

```
Media: [
    {
        _id: element id,
        fmt: media format,
        owner: userid,
        title: title of the media item,
        ts: timestamp,
    }
]
```

Where the possible values for 'fmt' are: 0=video, 1=audio.

This schema was initially proposed by Chris Kiefer who previously did research on the same dataset.

With the research database running on a shared cluster, the Music Circle data was imported in parallel thus reducing the computation time by a factor of 4.

At the moment all the time is invested in constructing a useful feature vector. The main difficulty encountered is in dealing with the time nature of the data. This dependancy makes it difficult to unambiguously establish the causes of a given action on the system. The problem is that the user actions after feedback depend on how the data is quantised in time. In general, one can not be sure for how long a particular action affects the subsequent user behaviour in the system. At the moment feedback actions are considered to be characterised by user clicks on a particular element. The quantisation of the data is made by establishing a time threshold after a user action (a click). The hypothesis is that all the log events under the threshold are caused by that particular click. This approach is very subjective and another alternative should be found. The success of the project depends heavily on how well this issue can be solved.

Planned work

On a chronological order the planned work consists in the feature vector construction, the clustering of the feature vectors, the principal component analysis of feature vectors and the probability distribution comparison.

As described in the previous section, the construction of the feature vector presents some challenging problems.

The clustering of the feature vectors will consist in the application of different clustering algorithms and the selection of the most accurate result.

The feature vectors dimensionality is expected to be low (5 dimensions approximately). In order to find the relevant components (feature reduction), the first approach is to calculate all the possible combinations of components in the vector ($n^2 - 1$ where n is the dimensionality of the vector). With these new set of feature vectors one can build a probability distribution of them for each user. The hypothesis considers that the distributions between successful and struggling students are different. Therefore, one can select the combination of components that maximise the distance between two distributions belonging to users from different categories (a successful student and a struggling student). This metric should allow the selection of the optimal

components of the feature vectors.
Once the principal components are selected and the distributions calculated one can proceed to evaluate the project as described in the "Methods" section.

References

[1] Agile manifesto. http://agilemanifesto.org/. Accessed: 2015-02-20.

[2] iPhython. http://ipython.org/. Accessed: 2015-02-20.

[3] MongoDB sharding. http://docs.mongodb.org/manual/sharding/. Accessed: 2015-02-20.

[4] Music Circle. https://goldsmiths.musiccircleproject.com. Accessed: 2015-02-20.

[5] PRAISE project. http://www.iiia.csic.es/praise. Accessed: 2015-02-20.

[6] PyMongo. http://api.mongodb.org/python/current/. Accessed: 2015-02-20.

[7] SciKit-learn. http://scikit-learn.org/stable/. Accessed: 2015-02-20.

[8] SciPy. http://www.scipy.org/. Accessed: 2015-02-20.

[9] Virtual Environment. http://docs.python-guide.org/en/latest/dev/virtualenvs/. Accessed: 2015-02-20.

[10] Félix Castro, Alfredo Vellido, Àngela Nebot, and Francisco Mugica. Applying data mining techniques to e-learning problems. In *Evolution of teaching and learning paradigms in intelligent environment*, pages 183–221. Springer, 2007.

[11] Imola K Fodor. A survey of dimension reduction techniques, 2002.

[12] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03 1951.