PUPIL: International Journal of Teaching, Education and Learning
ISSN 2457-0648

Duran-Dominguez et al., 2018

Volume 2 Issue 2, pp. 170-180

Date of Publication: 28th August 2018

DOI-https://dx.doi.org/10.20319/pijtel.2018.22.170180

This paper can be cited as: Duran-Dominguez, A., Gomez-Pulido, J. A., & Pajuelo-Holguera, F. (2018). Virtual Classrooms as Data Sources for Prediction Tools. PUPIL: International Journal of Teaching, Education and Learning, 2(2), 170-180.

VIRTUAL CLASSROOMS AS DATA SOURCES FOR PREDICTION TOOLS

Arturo Duran-Dominguez
On-line Campus, University of Extremadura, Caceres, Spain
arduran@unex.es

Juan A. Gomez-Pulido
School of Technology, University of Extremadura, Caceres, Spain
jangomez@unex.es

Francisco Pajuelo-Holguera
School of Technology, University of Extremadura, Caceres, Spain
fpajueloc@unex.es

Abstract

Nowadays, on-line campus are very important in the learning process of students, since they can access to teacher's resources easily. Moreover, on-line campus provide useful tools for building evaluation processes by teachers. Under this point of view, knowing the strengths and weakness of a student before his evaluation allows to plan better his learning schedule when we analyze the history of the student and his colleagues, in the same tasks. In addition, the instructor can predict the difficulty level of the proposed exercises for determined students, allowing him to adjust better the evaluation tasks. Predicting the student performance can be obtained from Machine Learning tools, specifically Collaborative Filtering techniques based on Recommender Systems. In this paper, we explain how we are using and including these techniques in a Moodle environment, in order to provide several useful resources for both students and teachers. In this context, virtual classrooms provide useful data for predicting purposes.

Keywords

Food Consumption, Healthy Food, Consumer Behavior, Food Market

1. Introduction

Predicting Student Performance (PSP) is a problem tackled by several mathematical techniques based on Machine Learning (ML). The main purposes of this problem are, on the one hand, to acquire knowledge about the learning progress of the students based on the prediction of non-completed tasks, and on the other hand, to know the difficulty level of the evaluation tasks designed by the teachers to be applied in on-line environments.

The application of the Information and Communication Technologies (ICT) to the higher education (Buttar 2016) has promoted many online environments to support the e-learning processes (Gavaldon-Hernandez 2017). Since the most popular on-line environment for students and teachers at university level are the virtual classrooms in on-line campus, we think it is very interesting to add prediction capabilities to the software tools that implement the on-line campus, such as Moodle. Once the corresponding prediction tools are installed, teachers and students can study and predict the learning progress in order to identify the areas where they should apply more effort, not only studying the subjects, but also designing more effective evaluation tasks, always under a custom study for each student.

In this paper, we start explaining the main machine learning tools, specifically the collaborative filtering, since this technique is the basis to design a software solution to sole the PSP problem, which is explained in the third section. After describing the on-line campus features of the University of Extremadura (Spain), we detail a proposal to solve the PSP problem and apply it in the on-line campus, in order to acquire knowledge about the learning progress of the students, as well as other benefits of the proposal. Finally, conclusions and future works are left to the last section.

2. Machine Learning and Collaborative Filtering

Machine Learning (ML) is a research area that provides many tools and methods that extract knowledge from data (Murphy, 2012). This knowledge can be used to predict future behavior or estimate unknown data. For this purpose, ML considers models built from training data, which are validated using test data to optimize performance criteria (Alpaydin, 2010). Machine Learning has high interest nowadays, since many systems are continuously generating many data, according to the concept of Big Data (BD), from which ML can infer useful knowledge or predict future users' behavior.

2.1 Types of Machine Learning

Nowadays, we can store and process high amount of data, under the concept of Big Data (BD). This information is collected from many learning processes where many students, professors and subjects are involved. Thus, we can process the data through ML techniques in order to extract useful knowledge. There are three types of ML techniques:

- Supervised Learning (SL): This method learns from the mapping of a set of X inputs to Y outputs, for a given set of N samples of input/output pairs (the training set). The input X represents the features set, whereas the output Y is the response variable. Popular applications of SL are classification, pattern recognition, and regression.
- Non-Supervise Learning (NSL). In this case, the input data X are not labelled and we have not the output Y. The goal is to learn automatically interesting patterns from the input sequences X. This technique can be used to reduce the data dimension or for clustering in those problems where there is not only one error measure. Popular applications of NSL are clustering, dimensionality reduction, and collaborative filtering.
- Reinforcement Learning (RL). This method is applied to solve decision problems, as robot movement, automatic driving, chess games, etc. In these problems, the learning is feed with positive or negative scores, depending on the decision.

2.2 Collaborative Filtering and Recommender Systems

Collaborative Filtering (CL) is a case of NSL, where the past information of the users' activity with regard to several tasks is considered in order to predict the future behaviour of a user in a personalized way. This prediction takes into account the activity of other users in the same task.

Recommender Systems (RS) are a popular CL technique (Jannach et al., 2011) that tries to elaborate recommendations according to the particular behaviour of the user. The algorithm techniques developed for RS are useful for predicting tasks. This is the case of the PSP problem. In this problem, the student performance is predicted for some tasks where their scores are unknown, for example, when the student has not attended them (Thai Nghe et al., 2012). Thus, PSP can be tackled as a RS, where techniques based on matrix factorization are usually applied.

Matrix Factorization (MF) is a popular method applied to build the prediction models in RS (Koren et al., 2009) (Rendle et al., 2008) and, consequently, to solve the PSP problem. The mathematical model consists of two matrices, from which the prediction is calculated multiplying a row in the first matrix and a column in the second one.

3. Building the Prediction Model

The PSP problem tackles the prediction of the student performance for some exercises, and uses the terms given in Table 1.

Table 1: *Notation in PSP problem*

Term	Meaning
S	Number of students.
I	Number of tasks.
P	Matrix (SxP) with the performance scores.
D^{knw}	Known performances.
D^{unk}	Unknown performances.
D^{train}	Training performances, used to build the prediction model.
D^{test}	Test performances, used to validate the model according to the Root Mean Squared Error (*RMSE*) (1) criteria. D^{test} is usually much smaller than D^{train}.
$\hat{P}_{D^{test}}$	Performance predictions of the test data.

$$RMSE=\sqrt{\frac{\sum_{s,i\in D^{test}}(p_{s,i}-\hat{p}_{s,i})^2}{|D^{test}|}} \quad (1)$$

The PSP problem is solved finding the matrix $\hat{P}$ with minimum error. This model is obtained basing on the RS model, particularly considering matrix factorization.

There are K latent factors implicit in the prediction model. The right number of latent factors is difficult to know, since it implies a deep knowledge of the learning process involved when a student solves an academic task. Nevertheless, we can approach this number performing few experiments before.

Matrix Factorization builds *P* as the product $W1\ W2^T$, where *W1* and *W2* are matrices of sizes (*S* x *K*) and (*I* x *K*) respectively. The first matrix describes the *K* features of the student *s*, whereas the second one contains the K features of the tasks *i*. This way, the performance prediction of the student *s* in the task *i* is calculated in (2).

$$\hat{p}_{s,i} = \sum_{k=1}^{K}(w1_{s,k}w2_{i,k}) = w1_s w2_i^T = (\mathrm{W1W2^T})_{\mathrm{s,i}} \quad (2)$$

The prediction methodology follows three steps. First, we use the training dataset to build W1 and W2. In this training phase, the contents of such matrices is optimized by minimizing the differences between real and predicted values. For this purpose, we consider Gradient Descent (GD) algorithm (Bottou 2010). This algorithm takes into account the learning rate (β) and regularization factor (λ). The second step begins after the prediction model is available; then, the test dataset is considered to validate the model by predicting their values and comparing them with the real ones. The third step calculates all the unknown values of the performance matrix, considering the last model found in the training step.

This prediction methodology pursues two goals. First, the model allows us to predict unknown values, as when the students have not performed certain academic tasks. Second, we can recommend to the students some tasks according to their performances, allowing so to improve the learning process.

4. A Proposal to Solve PSP in On-Line Campus

We propose to apply the ML tools to solve the PSP problem in an on-line learning campus corresponding with a university environment, following a methodology implemented in the software tools of that on-line campus.

4.1 On-Line Campus

The University of Extremadura (Spain) developed 10 years ago a virtual environment (CVUEx) to support the on-line learning. This environment is composed of several software tools: web front-ends (Portal), a wide set of virtual classrooms (Avuex) end many virtual spaces (Evuex) oriented to different purposes (conferences, seminars, workshops) for the university community.

Figure 1 shows some usage statistics of CVUEx along 2017 (from 1/january/2017 to 3/december/2017), where the label "others" refers to a set of 8 software services not cited before.

The software services are implemented in the Moodle platform (Wild, 2017), a learning environment widely used in the academic community.

CVUEx is a web application; therefore, the protocol used tipically is *https*. The following data were obtained from the analysis of the activity *log* files of the CVUEx web servers, for the domain "campusvirtual.unex.es" in 2017:

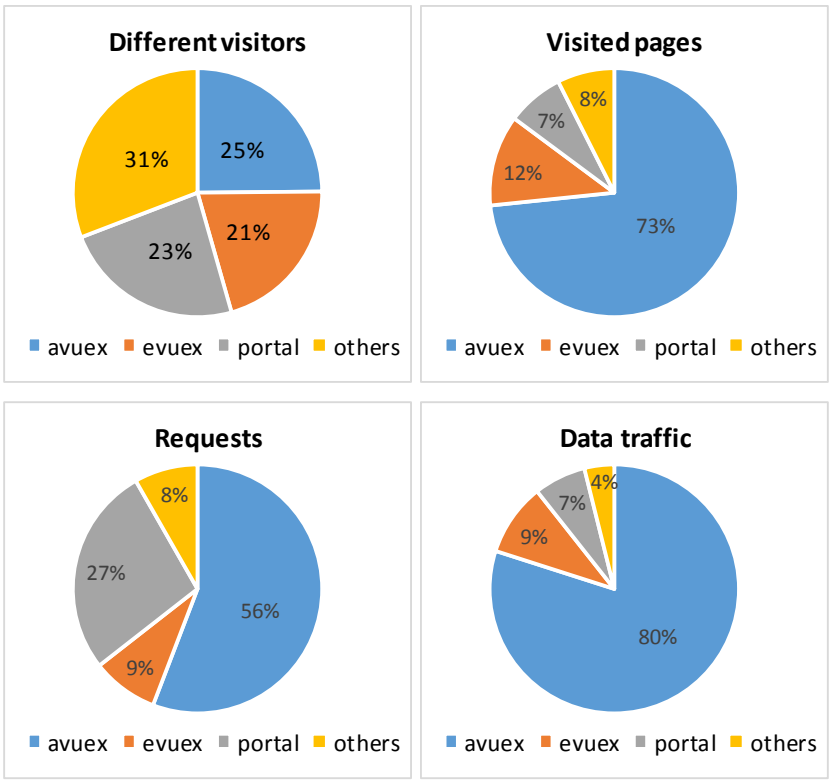

Figure 1: *Some usage statistics of CVUEx along 2017 (from 1/1/2017 tol 3/12/2017).*

- Different visitors: 3,076,371.
- Number of visits: 11,704,040.
- Pages: 218,193,084.
- Requests: 339,327,223.
- Data bandwidth: 14.93 TB.

The following numbers show a general insight of the on-line campus sizes:

- Virtual classrooms: 3,447.
- Users: 71,537.
- Teachers: 4,329.
- Students: 67,208.
- Files duplicated: 1,383,505.
- Files: 3,489,968.
- Hard disk usage: 931 TB.

4.2. Proposal of Prediction Implementation

Figure 2 shows the general architecture proposed in our work. The data with regard to students and tasks for each virtual classroom can be obtained from de databases of the CVUEx servers.

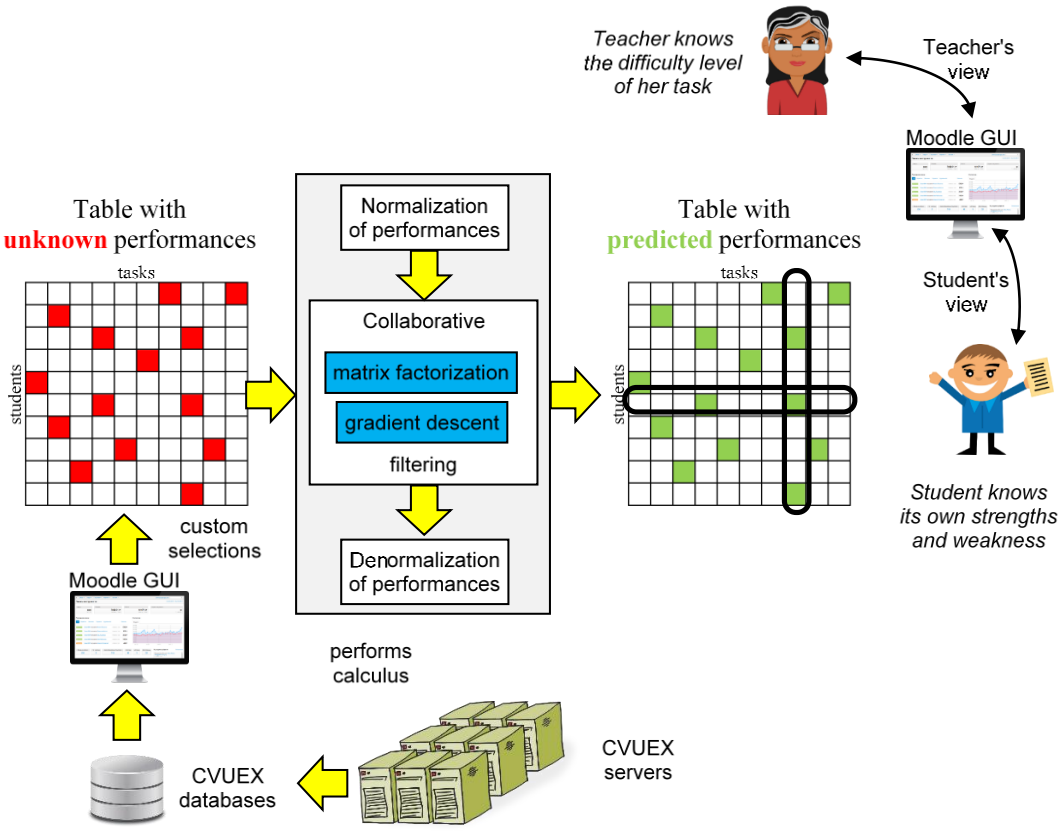

Figure 2: *General architecture of our research proposal, where the prediction tools for the student performance are included in on-line campus software environment based in Moodle.*

From these databases, we are working on a method to allow teachers to generate custom tables of the relationship "student - performs - task", with the corresponding performance scores, using the Graphical User Interfaz (GUI) of Moodle. This method follows a particular protocol:

- Teachers or students can build the performance tables only for particular cases.
- Those tasks without enough activity by the students in the learning process are automatically removed. This affects to the student profile in the Moodle's GUI.
- Those students without enough activity for particular evaluation tasks are automatically removed. This affects to the teacher profile in the Moodle's GUI.

  The table generated by this protocol can present empty cells to be predicted;

- The prediction can be automatically done once the user accesses to the virtual classroom.
- The prediction is performed by collaborative filtering algorithms running in the servers.

The prediction results (full table) is shown in the Moodle's GUI according to the corresponding profile (student or teacher):

- The student can view the prediction of his incomplete/unfinished tasks. Any prediction result corresponding with other students can be shown in no case.
- The teacher can view the prediction of all the evaluation tasks or exercises, and the corresponding students in his virtual classroom.

4.3. Example of Prediction

The following example shows the application of the prediction methodology:

- Subject: "Introduction to Computers".
- Students: 207.
- Year: 2017.

The data obtained from CVUEx were processed to remove students and tasks without significant activity; otherwise, these data could invalidate the prediction process. For this purpose, we applied several filters:

- 27 tasks without any activity were removed.
- Tasks with less than 50% scores were removed.
- 72 students with activity under 25% were removed.
- Once applied the above filter, we recalculated the activity of the available tasks, finding the task with minimum activity of 67%. Therefore, we decided remove two additional tasks representing less than 80% of activity.
- After removing some duplicated or unclear tasks, we obtained eight final tasks corresponding to six evaluations of laboratory, and two exams with theoretical contents.
- Once recalculated the students' activity again, we removed 28 students with less than 75% of activity in the eight tasks obtained before.

The final performance matrix consists of 8 columns and 107 rows, corresponding with the tasks and students left after filtering the data.

The performance matrix represents an academic environment where there is not any student with less than 88% of activity in the eight evaluation tasks, and the task with the minimum students' activity has 80% of participation. The final data (Figure 3.a) consists of:

- S = 107 students.
- I = 8 evaluation tasks.
- D^{knw} = 800 known scores.
- D^{unk} = 56 unknown scores.
- D^{train} = 800 training scores (we chose as training all the known scores: $D^{train} = D^{knw}$).
- D^{test} = 102 scores (we chose as test scores one for each student following consecutive tasks; in case of coincidence with unknown score, we skip to the next student).

Figure 3.a shows some students and their corresponding performances: yellow cells show unknown values and blue cells are test data. The scores go from 0 to 10.

We build the prediction model considering 64 latent factors and a learning rate of 0.8, obtaining RMSE = 0.37. Figure 3.b shows the results (the unknown scores predicted).

We can analyse the behaviour of students and tasks comparing both tables of Figure 3, allowing us to detect the strengths and weakness of the learning process.

	Teo1	Teo5	Lab1	Lab2	Lab3	Lab4	Lab5	Lab6
	1	2	3	4	5	6	7	8
1	9.50	10.00	8.21	7.90	8.89	8.75	10.00	10.00
2	6.00	5.69	6.61	5.81	10.00	6.38	7.50	9.50
3	7.85	8.03	8.33	8.21	9.38	7.25	10.00	10.00
4	5.38	7.06	5.00	4.53	6.35	6.69		2.00
5	8.30		8.33	7.91	9.26	2.88	9.30	5.00
6	9.50	8.92	7.56	6.49	8.63	9.25	2.00	10.00
7	6.95		6.79	6.72	8.17	8.75	9.00	10.00
8	8.83	7.67	7.98	7.93	9.26	4.75	6.50	3.00
9	8.70	7.78	4.52	1.41	5.00	4.25	7.00	6.00
10		3.64	5.71	1.69	8.26	4.63	3.00	3.00
11	7.85	7.14	3.10	1.30	7.50	8.13	8.00	10.00
12	7.80	9.42	5.12	5.76	8.82	7.56	5.00	9.00
13	8.68		6.96	5.78	9.44	6.25	8.00	2.00
14	8.29	7.69	4.58	3.16	5.00	7.13		7.50
15	6.55	5.83	6.01	5.74	10.00	5.88	8.50	3.00
16	9.20	8.94	7.14	8.84	8.89	9.13	7.00	3.00
17	6.75	8.33	7.98	7.04	6.46	4.63	2.00	
18	9.05		6.73	5.49	2.94	9.25	9.00	8.00

(a)

	Teo1	Teo5	Lab1	Lab2	Lab3	Lab4	Lab5	Lab6
	1	2	3	4	5	6	7	8
1	9.50	10.00	8.21	7.90	8.89	8.75	10.00	10.00
2	6.00	5.69	6.61	5.81	10.00	6.38	7.50	9.50
3	7.85	8.03	8.33	8.21	9.38	7.25	10.00	10.00
4	5.38	7.06	5.00	4.53	6.35	6.69	3.43	2.00
5	8.30	3.37	8.33	7.91	9.26	2.88	9.30	5.00
6	9.50	8.92	7.56	6.49	8.63	9.25	2.00	10.00
7	6.95	3.45	6.79	6.72	8.17	8.75	9.00	10.00
8	8.83	7.67	7.98	7.93	9.26	4.75	6.50	3.00
9	8.70	7.78	4.52	1.41	5.00	4.25	7.00	6.00
10	4.03	3.64	5.71	1.69	8.26	4.63	3.00	3.00
11	7.85	7.14	3.10	1.30	7.50	8.13	8.00	10.00
12	7.80	9.42	5.12	5.76	8.82	7.56	5.00	9.00
13	8.68	3.33	6.96	5.78	9.44	6.25	8.00	2.00
14	8.29	7.69	4.58	3.16	5.00	7.13	4.66	7.50
15	6.55	5.83	6.01	5.74	10.00	5.88	8.50	3.00
16	9.20	8.94	7.14	8.84	8.89	9.13	7.00	3.00
17	6.75	8.33	7.98	7.04	6.46	4.63	2.00	5.06
18	9.05	3.61	6.73	5.49	2.94	9.25	9.00	8.00

(b)

Figure 3: *Performance matrices before (a) and after (b) prediction.*

It is important to note that the prediction of an unknown task for a particular student considers not only the performance of this student in the remaining evaluation tasks, but considering also the performance of the remaining students for the same task for which we are predicting the unknown score of the student.

5. Conclusions and Future Works

The prediction based on collaborative filtering is very useful for students and teachers when it is analysed under a learning point of view:

- The student knows the prediction of his incomplete task. Since this prediction is based on its own performances in other tasks and the performances of his colleagues in the same task, this information can be very useful for him in order to identify the main strengths or weakness in the learning process of the corresponding subject.
- The teacher can analyze the level of difficulty of a particular task taking into account the performance of all the students, including those students whose performances were predicted because they did not complete or attend the evaluation exercise.

As future work, we want to design a methodology to automatically extract data from the virtual classrooms databases and filter the students and tasks not interesting for the prediction analysis. This research line is key to implement the proposed prediction methodology with success in the software tools of the on-line campus.

Acknowledgments

This work was partially funded by the Government of Extremadura under the project IB16002, and by the AEI (State Research Agency, Spain) and the ERDF (European Regional Development Fund, EU) under the contract TIN2016-76259-P.

References

Alpaydin, E. (2014). Introduction to Machine Learning Second Edition The Massachusetts Institute of Technology Press Cambridge, Massachusetts, USA.

Bottou, L. (2010). Large-Scale Machine Learning with Stochastic Gradient Descent. Proc. of 19th International Conference on Computational Statistics, G S. Y. Lechevallier, ed., Springer, 177-186. https://doi.org/10.1007/978-3-7908-2604-3_16

Buttar, S. (2016). ICT in Higher Education. People: International Journal of Social Sciences, 2 (1), 1686-1696.

Gavaldon-Hernandez, G. Azqueta, D. (2017). E-Learning, Virtual Learning and Social Capital. People: International Journal of Social Sciences, 3 (2), 1298-1308. https://doi.org/10.20319/pijss.2017.32.12981308

Jannach, D., Zanker, M., Felfernig, A., Friedrich, G. (2011). Recommender Systems. An Introduction: Cambridge University Press.

Koren, Y., Bell, R., Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. Computer, 42 (8), 30-37. https://doi.org/10.1109/MC.2009.263

Murphy, K. P. (2012). Machine Learning. A Probabilistic Perspective". The Massachusetts Institute of Technology Press. Cambridge, Massachusetts, USA.

Rendle, S., Schmidt-Thieme, L. (2008). Online-updating regularized kernel matrix factorization models for large-scale recommender systems. 2008 ACM Conference on Recommender systems, Lausanne, Switzerland, 251-258. https://doi.org/10.1145/1454008.1454047

Thai-Nghe, N., et al. (2012). Factorization Techniques for Predicting Student Performance. Educational Recommender Systems and Technologies: Practices and Challenges, IGI-Global, 129-153. https://doi.org/10.4018/978-1-61350-489-5.ch006

Wild, I. (2017) Moodle 3.x Developer's Guide. Packt Publishing.