A Classification Approach to Recognize On-Task Student's Behavior for Context Aware Recommendations

Lisa Roux, Thierry Nodenot, Patrick Etcheverry, Pantxika Dagorret, Christophe Marquesuzaa, Philippe Lopistéguy

A Classification Approach to Recognize On-Task Student's Behavior for Context Aware Recommendations

Lisa Roux [0000-0003-4221-4421], Thierry Nodenot [0000-0003-0665-9079], Patrick Etcheverry [0000-0002-5624-4624], Pantxika Dagorret [0000-0002-1833-9016], Christophe Marquesuzaa [0000-0002-1834-3648], Philippe Lopisteguy [0000-0002-1670-8105]

Université de Pau et des Pays de l'Adour, E2S UPPA, LIUPPA, Anglet, France
{eroux, thierry.nodenot, patrick.etcheverry, pantxika.dagorret, christophe.marquesuzaa, philippe.lopistegu}@iutbayonne.univ-pau.fr

Abstract. The increasing development of e-learning systems has raised the necessity to apply recommender systems with the aim of guiding learners through the various courses, activities, etc. at their disposal. The learner-oriented approaches allow the recommendations to fit the user's needs as precisely as possible. Nevertheless, due to the multiplicity of possible educational situations and individual particularities, offering adaptive recommendations and diversity is still a major challenge. In order to improve this aspect and provide the learner with recommendations appropriate to both their current specific needs and general profile, we focus on an hybrid system whose knowledge will be augmented through the learner's activity and results. This system will base its analyses and future recommendations according to the evolving student's profile and behaviour during the task. For that purpose, a first step is to categorize the on-task student's behaviour. This paper focuses on this problem and proposes a model, provided by educational sciences, on which the recognition process could be based.

Keywords: Learner behaviour model, E-learning, Supervised Classification, Recommender Systems,

1 Introduction

In recommender systems (RS), two important components need to be considered: the users and the items. The main challenge is to recommend a short selection of appropriate items (i.e. items that fit the user's needs or interests) to a given user, in order to help them to choose from a wide variety of items. In e-learning, the recommendation is mostly based on the content of the items and the collected data that describes the online user's behaviour. Providing the learner with appropriate learning recommendations is a major challenge, since each learner has specific needs and way of learning, depending on their own interests and cognitive involvement, and the learning situation.

In order to provide personalized recommendations taking into account the evolution of the on-task learner's behaviour, we propose in this paper a learner model relying on educational sciences and aiming at describing the cognitive strategies of a learner during a problem-solving activity. This learner model is part of a RS architecture that we detailed in [1]. The learner model presented in this study has been chosen for its genericity: it is expected that it can be trained and used for any problem-based learning [2] in practicum-centered situations (i.e. practical exercises, projects) [3]. Each situation is described as a series of tasks through the guidelines provided to the students, a work environment with the suitable software so that the students can perform the task and possibly collaborate, a set of resources (e.g. numerical documents, work files produced by the students), success criteria for each task, an available support through a LMS comprising aids, resources, tips, components for a solution, course supplements, etc.

This paper focuses on the student's current behaviour classification and is organized as follows: related works on recommendation approaches and e-learning RS is given in Section 2; after having described the cognitive model on which the proposed solution is based and the overall functioning of this solution, Section 3 show the first results that we obtained; Section 4 outlines conclusions of this study.

2 Related works

E-commerce RS are mostly based on matrix factorization [4]. A matrix of *n* users and *m* items is used to represent the data on which the calculations will be done, each matrix cell corresponding to the rating given to item i by the user u. Collaborative and content-based filtering are widely used methods because they are both efficient and easy to implement, but the knowledge based filtering and hybrid approaches can also be employed. The education RS are based on the same filtering methods in a variety of approaches, from matrix factorization (e.g. [5, 6]) to complex classification methods, such as fuzzy-tree [7], convolutional neural networks [8], neuro-fuzzy technics [9, 10], etc. In some works, deep attention has been paid to allow the teacher to implement their own rules [9]. When used, the learner model is mostly based on their knowledge background [9], learning objectives, past activities [11], learning style [12], knowledge level, and sometimes several of them [7, 9, 13]. These models can be categorized as follows: the models based on the student's learning trajectory and those based on their educational needs.

However, so far, it appears that none of the existing e-learning RS make recommendations based on both aspects. Moreover, using the methods of e-commerce RS in the education area raises the question of pedagogical efficiency and some ethical issues. Similarly, the selection of the learner model and the choice of basing (or not) the recommendations on a teaching model are crucial stakes that require careful attention. In accordance with the proposals that we have made in previous researches [14] that explored the ethical and epistemological issues of e-learning recommender systems, the learner model presented in this paper is provided by the educative sciences. It has to fulfill two requirements [1]:

1. The classification of students must be meaningful.
2. It must be pedagogically useful and relevant.

3 Proposed solution

3.1 Description of the student's behaviour : the mode of reasoning and the degree of activity

When a learner is working on a pedagogic task, they can be more or less confident, proactive, hesitating, etc. and the teacher generally takes into account the signs of ease when they want to help and guide them: they recommend to read a document, see another similar task, etc. depending on the student's needs, that is, how much they feel comfortable with the issues to be addressed during the task. However, although it is well documented that the learners' emotions when completing a task have a significant impact on their performances and stance (i.e. perseverance, dropping out) [15, 16], we can hardly automatically determine how they feel during the task: one's feelings are internal thus very difficult to assess for an outside observer.
In some papers, the emotion recognition is operated through various technical or technological devices (e.g. computer vision [17, 18]), smart cushion [19], voice recognition [20]) but, since we want our system able to work with reduced hardware costs, we chose to base our system on a learner model that describes their behaviour – thus on an observable learner activity – and that comes from the educational sciences.

In the problem-solving activities, learners alternate between two main modes of reasoning, which are exploration and exploitation.

Exploration: This strategy aims at experimenting with new alternatives or at acting on the environment in order to generate new stimuli [21]. In our experiments, we expect that the learners undertake manipulative actions on tools, environment, available objects, etc., trying and testing different combinations.

Exploitation: This strategy consists in the use of existing knowledge (declarative, procedural) in a given situation. It aims to use it in order to build hypotheses, ideas, and strategies to suit the problem-situation they meet, then get involved in a reality-check. In our experiments, we expect that the learners test their hypotheses/approaches (e.g. their specifications) so that they can validate them or not.

The relationship between the exploitation of old certainties and the exploration of new possibilities is a central concern of studies in adaptive processes [21]. Sometimes learners have to take time to exploit existing knowledge, but they also may have to explore the situation in different ways in order to handle and test the various means at their disposal, so that they can develop new knowledge then raise new solutions [22].

There are two main advantages coming from this model. First, although there can be ambiguous situations difficult to categorize, there are easily observable practice patterns that allow a good class definition for machine learning. Moreover, these behaviours, although objectively observable, can be an expression of the students' ease during the task. For example, whether a learner continues to use the exploitation problem-solving strategy while facing difficulties that they fail to overcome, one possible explanation might be that they lack confidence in exploring new insights, new ways of using the available tools, etc. On the contrary, a learner who would keep trying to explore without success could lack both theoretical and practical knowledge about the environment or the tools. In the former situation, the learner could be encouraged to perform easy tasks in order to build confidence, while in the latter, they could be advised to strengthen the knowledge and skills necessary to complete the task in order to build appropriate hypotheses. In this way, appropriate recommendations can be made to the student, based on the history of their modes of reasoning during the current task, combined with two other information: the amount of time they stay in the same mode and the degree of difficulty encountered. The latter is calculated through a questionnaire.

3.2 Learning steps

Our system requires two intertwined learning steps: the former to determine the classification rules of the student's current reasoning mode, the latter to define the function used to identify their current activity level given their reasoning mode. We composed six indicators, which have been selected because they fulfil three requirements:

1. They can be obtained by aggregating data describing the mouse and keyboard actions of the student
2. They have to participate to describe both the student's current reasoning mode and activity degree
3. They have to be generic (independent to the task), because the final recommender system is expected to work for any problem-solving task.

In our proposal, the student's behaviour is decribed by a 7-dimensional vector :

Frequency of document shift: This indicator shows whether the student regularly shifts back and forth across the available pedagogical and work documents. It is the average of the number of times the student has launched or clicked on another document than the active one in five minutes.

Frequency of the work verification: This indicator shows whether the student regularly checks the validity of their work. In the case of a programming task, it is calculated by dividing the number of times the student has launched the compilation operation by the number of complete structures in their program. This is an optional indicator, since not all software programs can offer such a functionality

Rate of activity: This indicator assesses whether the student intensively uses the keyboard and the mouse (e.g. clicking, rolling the mouse wheel, using the scrollbar), in comparison with the average students' rate of activity (number of actions per minute).

Rate of writing: It indicates how much the student writes, on average, in comparison with the average students' rate of writing (number of written character per minute).

Frequency of erasure: This assesses the student's certainty about what they write. It is defined as the number of times the student presses an erase-key per minute.

Rate of the working document elasticity: This information is described by two indicators. The frequency of erasure is insufficient to evaluate adequately the student's production progress. Indeed, when we detect that they have pressed an erase-key, we do not know whether they have erased only one character or a bigger selected extract of the text. This indicator reflects the degree of variability of the size of the working document S_i. S_i is measured at regular time intervals (30 sec). The indicators are calculated as follows, based on the average size of a document and its standard deviation :

$$\mathrm{m} = \frac{\sum_{i=1}^{n} |S_i - S_{i-1}|}{t} \tag{1}$$

$$R_e = \sum_{i=1}^{n} \left| \frac{|S_i - S_{i-1}|}{t_i - t_{i-1}} - m \right| \times \frac{t_i - t_{i-1}}{n} \quad (2)$$

3.3 Classification and evaluation

Data collection ranges from the acquisition, at runtime, of the different sources of traces (collection per student's workstation) to the gathering of these traces into an object-model that can further be exploited (queries, mining). At the end of that process, described in [23], in average 500 macro-interactions per student were collected. We obtained the archives of 60 students' actions. In order to classify them, we labelled manually these data, by watching the 60 students working on a programing task.

Several classification strategies have been investigated in our work. Since the student's activity depends on their current problem-solving strategy, we considered the cascade hybridization method, which performs the problem-solving strategy classification followed by a regression of the activity.

Concerning the classification step, due to the specificities of our data sample (i.e. small, based on surveys carried out by human observers), only classification methods that can be applied on small samples and robust to noisy measurements may be used. For the first step, which consists in classifying the data into two classes (i.e. exploitation and exploration), three different classification methods were therefore considered: Random Forrest (RF), Naïve Bayes (NB), KNeighbors (KNN). Indeed, KNN is mostly used for small samples with few explanatory variables, since the algorithm speed slows as the number of observations increases and the results are less accurate with many characteristics. NB does not require huge amount of data neither, and is not sensitive to irrelevant characteristics. Although it is efficient on large dataset, RF can also deal with little ones. Moreover, it reduces overfitting, which is particularly harmful for noisy datasets. We used 75% of the original dataset to train the model and 25% to test it. Due to the scattered data distribution, we used KNN with k=2,3,4,5. We calculated the error rate, which computes how much items are wrongly classified among the testing data set, and the f1-score for each classification. The results are reported in fig. 1 and show that the classification is better when k=3. Basically, the f1-score is the harmonic mean between precision and recall [24, 25], which are respectively how many of a class is found over the whole number of elements of this class and how many are correctly classified among that class:

$$\text{Recall} = \frac{\text{number of correctly recommended items}}{\text{number of interesting items}} \quad (3)$$

$$\text{Precision} = \frac{\text{number of correctly recommended items}}{\text{number of recommended items}} \quad (4)$$

$$\text{f1-score} = \frac{2 \times (\text{Recall}+\text{Precision})}{\text{Recall}+\text{Precision}} \quad (5)$$

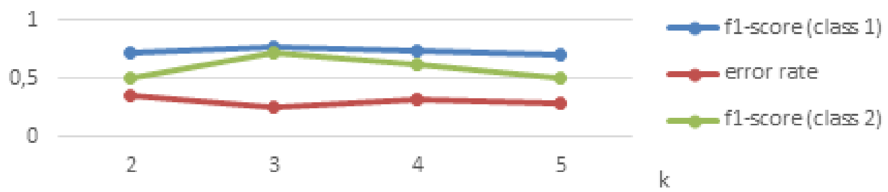

Fig.1 Evaluation of k-NN classification with k=2,3,4,5

We compared the results by using those three classification methods, that is 3-NN, RF, NB. As Figure 2 shows, results are better with 3-NN, although they look quite satisfying in RF and NB, with a quite little error rate. In order to confirm these observations, we used SHAP, which is a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions [26].

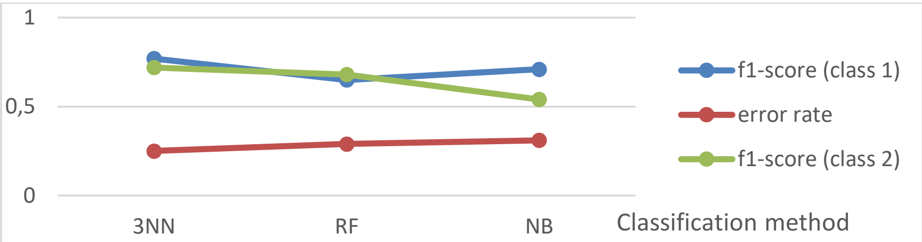

Fig.2 Evaluation of 3-NN, RF, and NB classifications

The results obtained using SHAP shed new light on these first results. Fig. 3 shows that the characteristics used to classify data are not really decisive for 3-NN, whereas the distinction of classes based on the same characteristics seem relevant for RF (fig. 4).

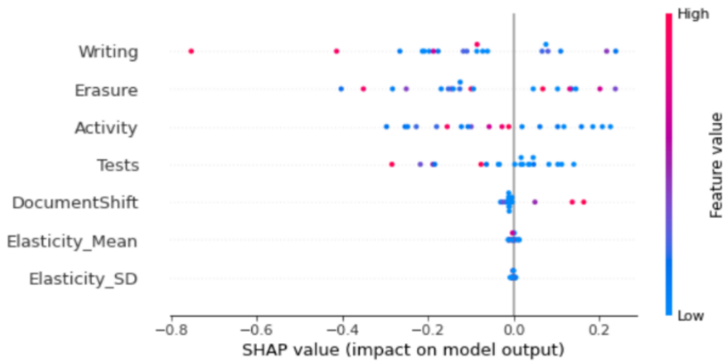

Fig 3. SHAP on 3-NN classification

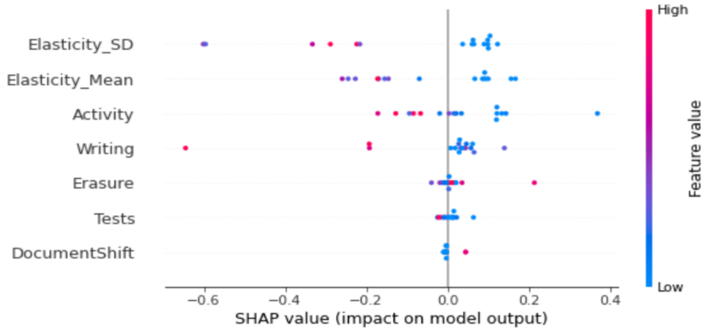

Fig 4. SHAP on Random Forest classification

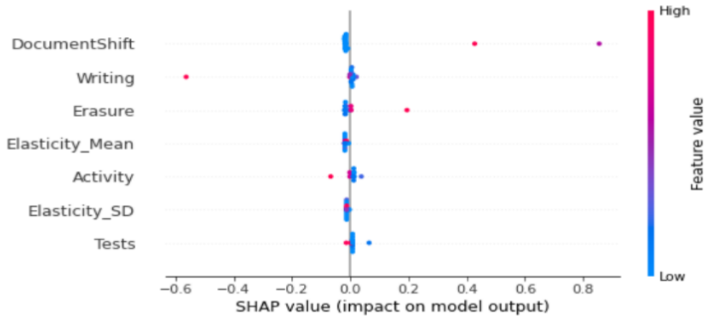

Fig 5. SHAP on Naïve Bayes classification

The classification provided by NB does not appear to be very conclusive (fig. 5), since the characteristics seem not discriminating, except for a very reduced number of individuals. All these results lead us to think that our model is quite promising, especially using 3-NN and RF, but has to be enhanced, notably through improvements to the dataset, in particular by increasing the number of observations, and testing classifications using multiple versions of our existing characteristics. Indeed, we can use some descriptors that, although repetitive, will describe data from different perspectives. Then, an important task will be to select the most appropriate information. For example, in these first tests, the calculation of the working document elasticity is based on the number of lines of the work document. Yet, this information can be expressed in bytes, number of words, number of lines, number of meaningful blocks (e.g. in the context of a programming task, in could be a variable declaration, a condition block, a function). Although we have the intuition that the number of meaningful blocks would be more significant, we have to ensure our hypothesis through some data feature selection methods. Similarly, the student’s activity can be expressed in either an atomic (e.g. clicks, rolling, writing a character) or a larger level with meaningful actions (e.g. document shift, move in the document, sequence writing).

4 Conclusion and perspectives

In this article, we proposed an approach to recognize a learner's behaviour during a task. It is based on a student model that comes from the educational sciences. In relation with this model, we presented the characteristics used to describe and classify the learner's behaviour and applied three different classification methods (K-NN, Random Forest, Naïve Bayes) to examine whether the chosen model and characteristics can provide a sound foundation for classifying the students' behaviours and recognize them. The results are quite encouraging, but the datasets are still very small and other tests have to be performed with larger ones. Thus the next step is to collect vast amounts of data.

For that purpose, an automatic labelling tool appears to be necessary, since manual categorization is extremely time-consuming. Our idea is to ask the learners to answer, each 15 minutes, a little questionnaire of five questions. The automatic labelling will be based on their replies. In this way, once we have collected enough data, we will be able to select the most appropriate classification method and the definitive characteristics. Indeed, combined with our process of automatic collection and fusion of traces, this labelling tool will contribute to feed our recommender system with rich and numerous traces of students. By processing the keyboard and mouse actions of the students, features of their professional gestures, indicators on the content of the files that they have produced or modified, etc., we think that we can aim for a classification, then a RS intelligible for the teacher and their students. The coming months will allow us to confirm these ambitions and first results, firstly for programming task. Then we will extend our range by applying these methods for students of others educational fields involved in problem-solving situations and working on specific software.

References

1. Roux, L., Dagorret, P., *et al.* A Multi-layer architecture for an e-learning hybrid recommender system, *18th International Conference on Cognition and Exploratory Learning in Digital Age* (2021).
2. Wood, D. F. Problem based learning. *Bmj* vol. 326, no 7384, p. 328-330 (2003).
3. Gary, K. Project-based learning. *Computer,* vol. 48, no 9, p. 98-100 (2015).
4. Bell, R., Koren, Y., *et al*. Matrix factorization techniques for recommender systems. IEEE Computer, 42(8), 30–3 (2009).
5. Thai-Nghe, N., Thai-Nghe, N., *et al*. *Predicting Student Performance in an Intelligent Tutoring System* (Doctoral dissertation, University of Hildesheim) (2012).
6. Salehi, M. Application of implicit and explicit attribute based collaborative filtering and BIDE for learning resource recommendation, *Data & Knowledge Engineering*, vol. 87, p. 130-145 (2013).
7. Wu, D. Lu, J., *et al*. A Fuzzy Tree Matching-Based Personalized E-Learning Recommender System, *IEEE Trans. Fuzzy Syst.*, vol. 23, nº 6, p. 2412-2426 (2015).
8. Shu, J., Shen, X., *et al*. A content-based recommendation algorithm for learning resources, *Multimedia Systems*, vol. 24, nº 2, p. 163-173 (2018)
9. Sevarac, Z., Devedzic, V., *et al*. Adaptive neuro-fuzzy pedagogical recommender, *Expert Systems with Applications*, vol. 39, nº 10, p. 9797-9806, (2012).

10. Kinshuk, A., Nikov, A., *et al*. Adaptive tutoring in business education using fuzzy backpropagation approach. *Proceedings of the ninth international conference on human–computer interaction*,Vol. 1, pp. 465–468 (2001)
11. Khribi, M. K., Jemni, M. *et al*, « Automatic Recommendations for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval », in *2008 Eighth IEEE International Conference on Advanced Learning Technologies*, p. 241-245 (2008).
12. Qomariyah, N.N., and Fajar, A.N. Recommender System for e-Learning based on Personal Learning Style, in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, p. 563-567 (2019).
13. Dwivedi, P., and Bharadwaj, K.K. Effective Trust-aware E-learning Recommender System based on Learning Styles and Knowledge Levels, *Journal of Educational Technology & Society*, vol. 16, nº 4, p. 201-216 (2013).
14. Roux, L., Recommender Systems in e-learning: axiological, epistemic, and practical implications of the designing choices, *Bonn International Symposium*, Bonn, December 9th– 10th (2021).
15. Pekrun, R., and al. Academic emotions in students' self-regulated learning and achievement: A program of qualitative and quantitative research. *Educational psychologist*, vol. 37, no 2, p. 91-105 (2002).
16. Pekrun, R., Linnenbrink-Garcia, L. Academic emotions and student engagement. *Handbook of research on student engagement*. Springer, Boston, MA, p. 259-282, 2012.
17. Bosch, N., D'mello, S., *et al.* Detecting student emotions in computer-enabled classrooms. *IJCAI*, p. 4125-4129 (2016).
18. Tonguç, G., and Ozkara, B.O. Automatic recognition of student emotions from facial expressions during a lecture. *Computers & Education*, vol. 148, p. 103797 (2020).
19. Gravina, R., Li, Q. Emotion-relevant activity recognition based on smart cushion using multi-sensor fusion. *Information Fusion, 48*, 1-10 (2019).
20. Mannepalli, K., Sastry, P. N., *et al*. Emotion recognition in speech signals using optimization based multi-SVNN classifier. *Journal of King Saud University-Computer and Information Sciences* (2018).
21. March, J. G. Exploration and exploitation in organizational learning. Organization science, 2(1), 71-87(1991).
22. Axtell, C. M., Holman, D. J., *et al.* Shopfloor innovation: Facilitating the suggestion and implementation of ideas. *Journal of occupational and organizational psychology*, 73(3), 265-285 (2000)
23. Roux, L., Nodenot, T., *et al.* Supervised classifications and process to detect exploitation and exploration student's behaviours in higher vocational Education, *22nd IEEE International Conference on Advanced Learning Technologies* (2022).
24. Salton, G., McGill, M.J. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York (1983).
25. Van Rijsbergen, C.J. *Information Retrieval*, Second ed. Butterworths, London (1979)
26. Strumbelj, E., and Kononenko, I. "Explaining prediction models and individual predictions with feature contributions." *Knowledge and information systems 41.3*, 647-665 (2014).