International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 23
December 2018

An Enhanced Student Action Predicting in automation Environments for Procedural Training

P. Mahalakshmi
Department of CSE, Shri Vishnu Engineering College for Women (A), Vishnupur, Bhimavaram, West Godavari District, Andhra Pradesh.

P.R. Sudha Rani
Associate Professor, *Department of CSE, Shri Vishnu Engineering College for Women (A), Vishnupur, Bhimavaram, West Godavari District, Andhra Pradesh.*

Abstract— This paper focused on improving student performance prediction, based on their personal and academic performance characteristics. Due to the incredible growth in recent technology like social media, it may deter the students from their actual track, and this is one of the reasons for the students to perform poor in academic activities and it even leads to course drop outs. Predicting students' performance will alert the learner to know about their performance and it gives as a chance to improve their performance in future. The research purposes include data about students' performance from the academic and other class room activities in the college during the course time. , Educational Heat map algorithms are used to predict the student performance which is a module in automated intelligent education systems.

Keywords— Predicting, automation, Training, Action, Education.

I. Introduction

In this project, an automated assessment approach applied to divergent courses has been presented. In a controlled study, the scores of the experimental group showed that students gained a more solid grasp of concepts related to debugging, deployment, and versioning. This clearly supports the idea that automatic assessment promotes students' ability also to learn course contents.

Educational data mining has already achieved promising results, for example, with regard to the analysis of student performance or the prediction of student grades, especially in the field of web e-learning. However, there is hardly any research in the literature that has integrated data mining techniques into intelligent tutoring systems (ITSs) for example, to provide customized tutoring for each student.

This project presents a collective student model that has been designed to anticipate the actions that students are likely to take while completing a practical assignment in an educational environment for procedural training. This model is created from activity records or logs collected from students with a similar background that previously completed the same practical assignment. As we will see later, an ITS equipped with this collective student model can use hints to stop students from making certain errors or from floundering with the practical assignment. It is sometimes a good idea to let students make mistakes from which they learn. In other cases, however, it is better to give students the minimum amount of support that they need to progress independently towards problem solving and overcome their zones of proximal development. In this way, each student learns not from his or her mistakes but with a little bit of help. If necessary, the tutor gradually increases the level of support or scaffolding every time the student makes a mistake or gradually reduces the amount of help provided when the student makes progress.

Another reason for helping students not to make mistakes is to prevent student frustration when they fail too often. The proposed collective student model consists of several clusters of students, each of which contains an extended automaton. This automaton is a directed graph adapted for our purposes. As explained later, these clusters will help to provide automatic tutoring adapted to each student type. In order to confirm this claim, we validated the model using student logs collected in a training environment. This validation had two main goals: i) verify that the prediction error is acceptable for tutoring purposes; and ii) check whether clustering methods can classify students into groups that require different tutoring feedback. As we will see later, although students had a lot of freedom of action in this class, the model was reasonably reliable at predicting student actions and provided a useful classification of students into clusters according to their performance.

Computer-based assessment systems allow students to solve knowledge evaluation exercises and to submit their solutions. Robo-Prof and Automated System for the Assessment of Knowledge evaluation (ASES) are two Java-oriented assessment systems. RoboProf presents knowledge evaluation problems within a Web browser and is presented in a series of levels to give the student a view of progress, thus allowing him or her to move onto the next knowledge evaluation activity. The automatic knowledge evaluation assessment element of this tool is integrated with a multiple-choice questions system. The ASES project is a system that automatically assesses programs and provides access to learning materials and tools. An XML document is generated, containing comments on the programs assessed, a description of tests applied, and a final grade. ASES fits into an abstract framework that is known as the Evaluation framework (or ELF). The framework is intended to guide the construction and development of interoperable and reusable software components that can be combined together to meet the requirements of a particular education institution.

Three of the most important Web-based tools that support automatic assessment for several languages are Course Marker, BOSS, and Mooshak. Course Marker is a Web-based assessment system for evaluating Java knowledge evaluation assignments and diagrams. Assignment assessment is done by analyzing the program across a number of criteria: typographic, lexical structure, presence of particular features, program complexity, and execution efficiency. Course Marker also provides administration facilities, statistics reports, and a rich marking interface, allowing students to have their program graded at frequent intervals prior to submission.

The Web based tool to facilitate the online submission and processing of knowledge evaluation assignments. BOSS provides an extensible set of simple program metrics, such as number of comments and percentage of methods declared abstract. It supports online marking and provides an interface for delivering feedback. BOSS incorporates plagiarism detection software and uses a client–server architecture with separate clients for students and for authorized staff for security reasons. The first version of BOSS tested assignments written in the C language. A redesign of BOSS was required to test Java software automatically.

II. PROBLEM STATEMENT

A model that can predict student actions in procedural training environments. Additionally, this project explains how this model is integrated into an ITS architecture and how it can be used to improve the tutoring feedback by anticipating student errors as long as this is pedagogically convenient. The collective student model is created from student logs by clustering logs and computing an extended automaton for each resulting cluster. We should highlight that there are few ITSs in the literature that rely on data mining techniques to enhance their tutoring feedback. As a result of this validation, we concluded that the model can provide reasonably good predictions and support tutoring feedback that is more adapted to each student type. An application that displays the collective student model would be very useful for facilitating the definition of the tutoring strategy. In this way, the instructor could visualize when students make more mistakes or which part of the practical assignment students find easier. Based on this information, the instructor could decide where and what tutoring feedback the ITS should provide. Additionally, this could also help the instructor to improve his or her own teaching. Another line of future work will be to validate an ITS built upon the proposed model in order to evaluate the tutoring feedback induced by the proposed model

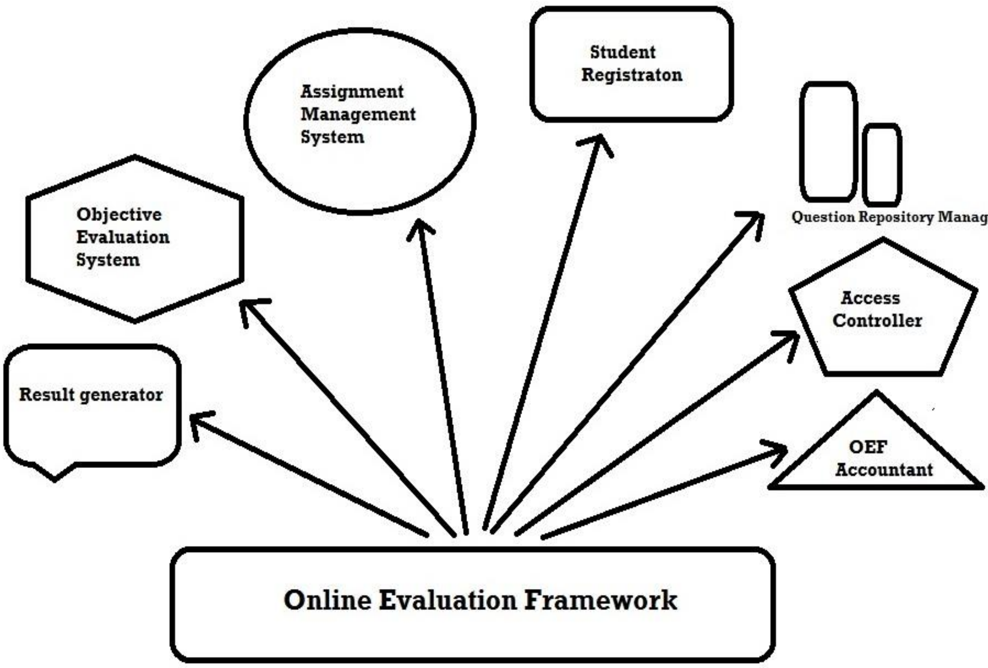
III. PROPOSED SYSTEM

This project presents a collective student model that has been designed to anticipate the actions that students are likely to take while completing a practical assignment in an educational environment for procedural training. This model is created from activity records or logs collected from students with a similar background that previously completed the same practical assignment. As we will see later, an ITS equipped with this collective student model can use to intelligently assess individual abilities of students. It is sometimes a good idea to let students make mistakes from which they learn. In other cases, however, it is better to give students the minimum amount of support that they need to progress independently towards problem solving and overcome their zones of proximal development. To obtain this sort of intelligence we propose the following two algorithms

In this way, each student learns not from his or her mistakes but with a little bit of help. If necessary, the tutor gradually increases the level of support or scaffolding every time the student makes a mistake or gradually reduces the amount of help provided when the student makes progress. Another reason for helping students not to make mistakes is to prevent student frustration when they fail too often and hence development of such a system propels student performance better. Instant Notifications to concerned parties. Querying users experiences flexibilities while initiating usages. In a controlled study, the scores of the experimental group showed that students gained a more solid grasp of concepts related. This clearly supports the idea that intelligent learning tools promote students' ability also.

IV. ARCHITECTURE

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 23
December 2018



Architecture diagram represents mainly flow of requests between users and Teacher usages of Evaluation resources, ASES Web Server. In this scenario overall system is designed in n- tires separately, specifically using three layers called client layer, Test Details Specification , Processing and Retrievals Instance layer provided using php scripts sources and Evaluation Specification layer provided with PHP sources. This project was implemented using n-tire architecture and a sample flow is represented here.

V. ALGORITHM

A. Algorithm: Heat Map Algorithm

1.For the input HM, extract the *n* largest peak points and use the locations of these peak points as the key points in later alignment steps: (***P1, P2, P3, ... Pn***), where ***Pi***=[*xi*, *yi*] is the location of the *i*-th key point with *xi* and *yi* being its *x* and *y* coordinates in the HM.

2. Organize the key points ***Pi*** for each input HM in an descending order according to their heat values in HM (i.e., *H*(***P****i*)>*H*(***P****j*) for *i* > *j*).

3. Shift the key points (***P1, P2, P3, ... Pn***) such that the gravity center of these points is in the center of the HM.

4. Scale the key points (***P1, P2, P3, ... Pn***)

5. Align the key points of the input HM with the target HM such that $\arg_T[\min[\sum_{i=0}^{n} |\mathrm{Gi} - \mathrm{Pi.T}|^2]]$, where ***T*** is a 2×2 matrix for aligning the key points of ***Pi***, and ***Gi*** are the key points for the target HM. And ***T*** can be achieved by linear regression.

6. Apply the final shift, rotation, and scaling operation derived from 2-4 on the entire input HM for achieving the final aligned version.

For Heat Map with only one peak, we will add an additional key point for alignment. That is, we first pick up the points whose heat values are half that of the peak point, and then the one which is farthest to the peak will be selected as the second key point for alignment. Since the direction from the peak to the additional key point represents the slowest-descending slope of the Heat Map surface, the HMs can then be suitably aligned by matching this slope. The key points are shifted, scaled, and rotated coherently (i.e., by the same parameter) in order to keep the overall shape of Heat Map during alignment. The key points ***Gi*** of the target Heat Map are assumed to be already shifted and scaled properly. In our Heat Map algorithm, we perform clustering on the Heat Map s in the training data and perform alignment within each cluster. After that, the mean of the aligned Heat Maps in each cluster is used as the standard surface (i.e., the target Heat Map) for representing the cluster during recognition.

VI. RESULT ANALYSIS

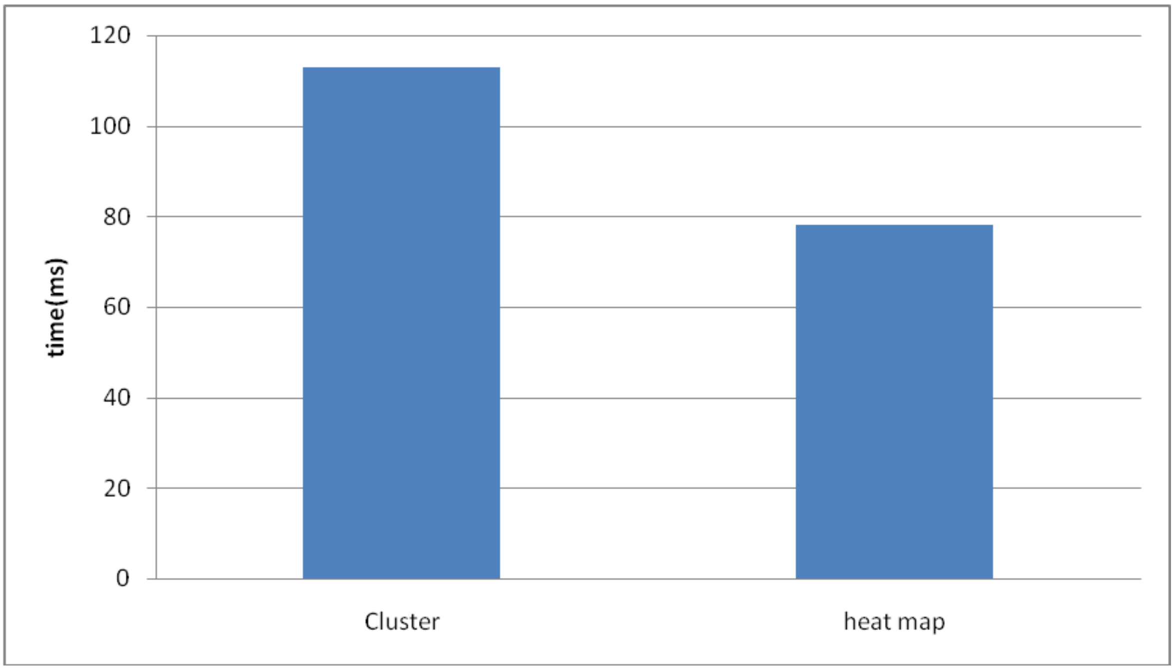


Fig 1: Cluster Vs Heat Map

TABLE I
CLUSTER VS. HEAT MAP

	Cluster	Heat Map
Time(ms)	113	78

A set of data and separating it into subgroups where the elements of each subgroup are more similar to each other than they are to elements not in the subgroup has been extensively studied through the methods of Cluster Analysis. This method can separate students into groups that can be recognized and characterized by common traits in their answers, without any prior knowledge of what form those groups would take (unbiased classification). In this paper we start from a detailed analysis of the data coding needed in Cluster Analysis, in order to discuss the meaning and the limits of the interpretation of quantitative results. Heat maps are a popular conversion optimization tool, but are they really any ... have enough sample size per page/screen before you act on results. Heat maps help you get an instant feel for an area by grouping places into categories and displaying their density visually.

VII. CONCLUSIONS

Our proposal achieves an automatic tutoring in procedural training more adapted to each type of student by applying methods of extraction and analysis of data, which can anticipate possible errors depending on its configuration. The principal application of the presented predictive model is to help students with preventing messages. For this, we have designed an ITS, presented above, which leverages the predictive model to provide that kind of tutoring. We consider that the advice of an expert educator or teacher of the subject is essential at design time, despite this ITS may become very independent once its tutoring strategy is configured. This is because the resulting predictive model needs to be analyzed for refining the tutoring strategy. In order to facilitate this task, it will be necessary to develop an application that displays the model to the expert or professor. In this way, he/she could visualize where students make more mistakes or where the practice is easier for them, and with this information he/she could decide where and what tutoring feedback is needed. Additionally, this could also help teacher to improve his/her own teaching.

REFERENCES

[1] C. Romero and S. Ventura, "Educational data mining: A survey from 1995 to 2005," Expert Systems with Applications, vol. 33, no. 1, pp. 135–146, 2007.
[2] C. Romero and S. Ventura, "Educational Data Mining: A Review of the State of the Art," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 40, no. 6, pp. 601– 618, 2010.
[3] R. S. Baker, "Educational Data Mining: An Advance for Intelligent Systems in Education," Intelligent Systems, IEEE, vol. 29, no. 3, pp. 78–82, 2014.
[4] L. S. Vygotsky, Mind in society: The development of higher psychological processes. Harvard university press, 1978.
[5] A. M. Olney, "Scaffolding Made Visible," in Design Recommendations for Intelligent Tutoring Systems. Orlando: U.S. Army Research Laboratory, 2014, ch. 26, pp. 327–340.
[6] H. K. Holden and A. M. Sinatra, "A Guide to Scaffolding and Guided Instructional Strategies for ITSs," in Design Recommendations for Intelligent Tutoring Systems. Orlando: U.S. Army Research Laboratory, 2014, ch. 22, pp. 265–281.
[7] C. Romero, S. Ventura, M. Pechenizkiy, and R. S. Baker, Handbook of Educational Data Mining. CRC Press, 2010.
[8] D. Perera, J. Kay, I. Koprinska, K. Yacef, and O. R. Zaïane, "Clustering and Sequential Pattern Mining of Online Collaborative Learning Data," pp. 759–772, 2009. [9] T. Y. Tang and G. McCalla, "Smart Recommendation for an Evolving E-Learning System: Architecture and Experiment," International Journal on ELearning, vol. 4, no. 1, pp. 105–129, 2005.
[10] D. Godoy and A. Amandi, "Link Recommendation in E-Learning Systems Based on Content-Based Student Profiles," in Handbook of Educational Data Mining. CRC Press, 2010, ch. 19, pp. 273–286.
[11] P. Fournier-Viger, R. Nkambou, E. M. Nguifo, A. Mayers, and U. Faghihi, "A Multiparadigm Intelligent Tutoring System for Robotic Arm Training," Learning Technologies, IEEE Transactions on, vol. 6, no. 4, pp. 364–377, 2013.
[12] G. W. Dekker, M. Pechenizkiy, and J. M. Vleeshouwers, "Predicting Students Drop Out: A Case Study." in International Working Group on Educational Data Mining. ERIC, 2009.
[13] C. Romero, P. G. Espejo, A. Zafra, J. R. Romero, and S. Ventura, "Web usage mining for predicting final marks of students that use Model courses," Computer Applications in Engineering Education, vol. 21, no. 1, pp. 135–146, mar 2013.
[14] J. A. Lara, D. Lozano, M. A. Mart´ınez, J. Pezos, and T. Riera, "A system for knowledge discovery in e-learning environments within the European Higher Education Area Application to student data from Open University of Madrid, UDIMA," Computers & Education, vol. 72, no. 0, pp. 23–36, mar 2014.
[15] C. Attunes, "Acquiring Background Knowledge for Intelligent Tutoring Systems." in EDM, 2008, pp. 18–27.
[16] A. Herskovits and R. Nachmias, "Learning about online learning processes and students' motivation through Web usage mining," Interdisciplinary Journal of Knowledge and Learning Objects, pp. 197– 214, 2009.