International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 4 | Issue 2 | ISSN : 2456-3307

Forecasting Student Actions in A Practical Guidance Setting

D. Vamsi Kumar Reddy [1] S.A.Md. Noorulla Baig, Mscit,Mtech [2]

[1]student, Department Of Computer Applications, Rayalaseema Institute Of Information And Management Sciences, Bairagipatteda, Tirupati, Andhra Pradesh, India

[2]associate Professor, Department Of Computer Applications, Rayalaseema Institute Of Information And Management Sciences, Bairagipatteda, Tirupati, ndhra Pradesh, India

ABSTRACT

Data mining is known to have a potential for anticipating client execution. Nonetheless, there are few investigations that investigate its potential for anticipating understudy conduct in a procedural preparing condition. This paper shows an aggregate understudy demonstrate, which is worked from past understudy logs. These logs are firstly gathered into groups. At that point an expanded machine is made for each bunch in view of the groupings of occasions found in the group logs. The primary target of this model is to foresee the activities of new understudies for enhancing the mentoring input gave by an astute coaching framework. The proposed demonstrate has been approved utilizing understudy logs gathered in a 3D virtual research center for educating biotechnology. Because of this approval, we presumed that the model can give sensibly great forecasts and can bolster mentoring input that is better adjusted to every understudy compose.

Keywords: Educational Data Mining, e-learning, Procedural Training, Intelligent Tutoring Systems.

I. INTRODUCTION

Educational information mining has just accomplished promising outcomes, for instance, with respect to the investigation of understudy execution or the expectation of understudy grades, particularly in the field of web e-learning. Nonetheless, there is not really any exploration in the writing that has coordinated information mining methods into canny coaching frameworks (ITSs), for instance, to give tweaked mentoring to every understudy.

This paper 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 foundering 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 3D virtual laboratory for teaching biotechnology. 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 virtual laboratory, the model was reasonably reliable at predicting student actions and provided a useful classification of students into clusters according to their performance. The structure of the remainder of the paper is as follows. It shows relevant works in the field of educational data mining. Section describes the proposed ITS architecture, which would be able to leverage the collective student model detailed later in Section. Section reports model validation detailing the method followed in this study and discussing its results. Finally, Section outlines the conclusions of this research and some future work.

II. RELATED WORK

The related work is divided into two sections. It briefly presents the main goals of educational data mining and mentions some of the key results with respect to web based e-learning systems. Section focuses on systems for procedural training equipped with ITSs, whose student logs have been analyzed by means of data mining.

Educational Data Mining (EDM):

EDM tries to use data sourced from the repositories of different types of learning environments to better understand learners and learning. Some general applications of EDM: impart understudy exercises and use of online courses to instructive partners; help with course upkeep and change by breaking down utilization information; examine how well the area is organized by understudy execution forecast; produce suggestions for understudies; foresee understudy evaluations and learning results; and model understudies. Given the extent of our examination, the writing survey will center around the last three EDM applications. Some researchers use data mining to provide clues, criticism or proposals about which content is best for every understudy. Some of these utilizationan ITS. The most every now and again utilized information digging systems for this reason for existing are affiliation, sequencing, classification and bunching. Different scientists attempt to anticipate various types of understudy learning results, for example, final evaluations or dropouts. The most successive information mining strategies utilized as a part of this gathering are affiliation, classification and grouping. Understudy demonstrating has a few applications, for example, the recognition of understudy conduct or learning issues. This gathering most as often as possible uses similar information mining procedures as above, in addition to measurable examinations, Bayes systems, psychometric models and support learning. One vital paper in the last gathering forms Moodle logs to find a specific understudy conduct demonstrate. They separate these logs into understudy bunches with comparative attributes utilizing a grouping technique and afterward apply process mining to each bunch to make a model (spoke to by a coordinated non-cyclic chart) that demonstrates the most regular arrangements of understudy activities. A fascinating finish of this paper, which is applicable for our exploration, is that charts, models or visual portrayals are less demanding to grasp. Educators and understudies find this condensed data more open. Along these lines, this data could be exceptionally valuable for observing the learning procedure and giving criticism. As we find from the works referenced in this segment, most research in EDM has concentrated on contemplating information or logs enlisted by web e-learning frameworks, as Moodle or MOOCs, or information gathered from understudy educational program. To the best of our insight, there is no some other proposition of prescient model in the writing to help procedural preparing conditions that depends on information mining. Subsequently, next segment will center around a shut related region where we have really possessed the capacity to find some intriguing commitments, procedural preparing situations outfitted with ITSs, whose understudy logs have been handled through information mining.

Intelligent Tutoring Systems with EDM in Procedural Training:

Two well-known ITSs that employ EDM are Assistment and Cognitive Tutor Algebra. Both are web tools that guide students through the process of solving math exercises. There are several data mining studies using data collected from these two environments, but they do not report whether or not the results of these studies have been used to improve the tutoring services. For example, data from Assistment were utilized to make a model to anticipate when an understudy is going to request a clue. EDM is connected in Cognitive Tutor Algebra I to make a model that distinguishes understudy mentalities/sentiments, for example, engagement, focus, perplexity, dissatisfaction, and weariness exclusively from understudy associations inside the guide . To the best of our insight, there is just a single learning condition for procedural preparing furnished with an ITS that depends on EDM. It is called CanadarmTutor. It reenacts the Canadarm2 mechanical arm utilized as a part of the International Space Station. This ITS gives help to clients on the best way to play out a right grouping of arm tasks to achieve an objective. To do this, it coordinates a subjective model to evaluate abilities and spatial thinking, and a specialist framework that naturally creates arrangement ways. As a result of the not well defined attributes of the critical thinking method, the ITS uses information mining systems to remove a halfway undertaking model from past client arrangements. Utilizing this model the ITS can recognize the learner plan and provides assistance based on the forecast of the client's next activity. In spite of the exploration that has just been led around there, the group is missing all the more by and large pertinent outcomes, for instance, prescient models that can be utilized as a part of in excess of one distinctive setting. There is likewise a momentous deficiency of smart instructive frameworks that exploit models created by EDM. The exploration displayed in this paper speaks to a stage forward towards the improvement of an ITS that use an aggregate understudy show figured by methods for EDM to offer better mentoring criticism. Additionally, this model is expected for procedural preparing in learning conditions and is space autonomous.

III. PROPOSED SYSTEM ALGORITHM

In order to leverage the presented collective student model, we propose an extension of a previous ITS architecture, the MAEVIF architecture, which is depicted in Figure 1. MAEVIF is a multi-agent architecture that is an adaptation of the classic ITS architecture for learning environments specialized in training. Within the extension of MAEVIF, this model encompasses a new agent, called Collective Student Agent.

Of the MAEVIF operators, let us center around the Student Modeling Agent. The Student Modeling Agent contains an adaptive, extensible and reusable student model that derives the understudy learning state utilizing a pedagogic cognitive conclusion with non-monotonic thinking capacities. The motivation behind this operator is to find every understudy's learning status, that is, the thing that he or she does or does not think about the subject. This can fill in as help for the customized programmed coaching of every understudy. Along these lines, if the understudy display contains enough data on a specific understudy, it will give great expectations of his/her conduct. For instance, if the understudy display realizes that an understudy as of late played out an undertaking effectively, it is likely that this understudy will play out the same (or a fundamentally the same as) assignment accurately once more.

One disadvantage of this student model is, however, that, if queried about the attainment level of a particular learning objective, it will need a lot of background information about the student with respect to that learning objective in order to give a reliable enough response, and this information will often not be available. For example, this may be the case if it is the student's first attempt at an exercise. This may constitute a problem when the tutoring agent needs to predict the student's next actions, because if the student modeling agent is not confident enough that the student knows which actions to take next, it will not be able to provide a good prediction.

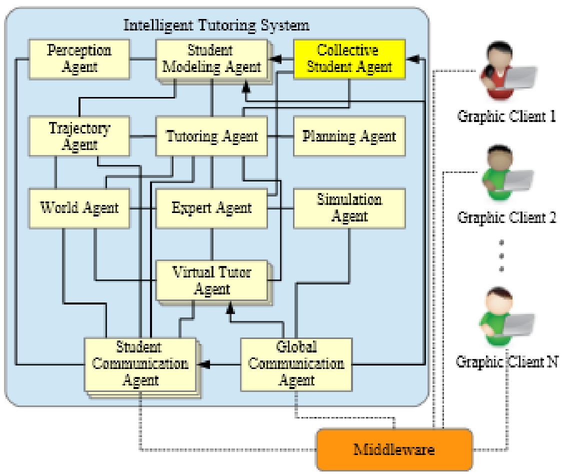

Figure 1. MAEVIF architecture with the new agent

As this paper shows, if the student model does not possess enough information on a particular student, the collective student model will be a reasonably good alternative. The collective student model comprises summarized data on past student action events that are used to predict the actions that a student under supervision is most likely to take next. The premise for creating this model is that the behavior of past students doing a practical assignment should be similar to current students with the same training completing the same practical assignment.

DESCRIPTION OF THE COLLECTIVE STUDENT MODEL

This model is created using historical data from past students and is continually refined with the actions from students under supervision. Our model can be considered as the result of the models/patterns discovery phase of the knowledge discovery in databases process adapted to EDM as formulated by Romero and Ventura Building this model was roused by our encounters assessing the 3D virtual lab for biotechnology. Amid display outline, we watched the conduct of understudies in the virtual world after the ethnographic technique. Consequently, as suggested by Mostow et al, the understudy logs were broke down by hand to recognize intriguing wonders. One of the conclusions drawn from this investigation was that understudies tend to fall into various gatherings relying upon their execution in the down to earth Moreover, succession models can be seen in information mining as chart based models, and they could be portrayed as coordinated diagrams for finding continuous occasions.

Hence the proposed model(seeFigure2)consists of several groups of understudies, every one of which contains an expanded machine, which is a coordinated chart fit for our motivations. As talked about later, these bunches will give programmed mentoring adjusted to every understudy write. Show creation has an indistinguishable principle stages from the procedure proposed by Bogar'ın et al. It is executed when the mentor is propelled. To make the model, it is important to get to understudy log occasions put away in the understudy demonstrate philosophy. Right off the bat, understudy logs are bunched in light of any of the clustering methods detailed in Sectio.Then, for each group, a robot is worked from the understudy logs of this group. Next, the model is refreshed with each new understudy activity at preparing time. Along these lines, the model can a dapt to the students under super vision better and therefore manage contrasts of conduct between current understudies and past understudies.

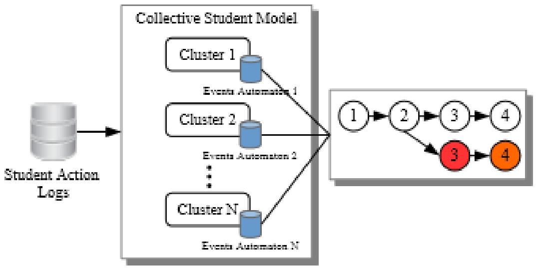

Figure 2. Collective Student Model

Extended Automaton Definition:

States are spoken to by circles and advances by bolts as appeared in Figure 3. Moreover, states are assembled into three zones: Correct Flow Zone, Irrelevant Errors Zone and Relevant Errors Zone. Advances signify occasions created by understudies all through an activity, for example, activities or endeavored activities that past understudies have performed up until this point and new understudies may rehash later on. Inside the robot, an occasion speaks to one of the accompanying circumstances: • avalid action for an exercise(do X event in Figure 3); • an endeavored activity hindered by the smart mentor (attempt X occasion in Figure 4), since this activity isn't right and the coaching system has been configured to keep understudies from playing out this activity; or • a blunder identified by the clever guide at the season of approving an erroneous activity that has not been blocked (come up short X occasion in Figure 3). A wrong activity may include in excess of one mistake, every one of which will be considered as an occasion.

Student logs can also contain irrelevant actions. This type of student actions does not have any influence on the development of the practical assignment. Depending on the pedagogical value of these actions, they will be considered for creating the collective student model (and treated as right actions) or will be discarded. Accordingly, states represent the different situations derived from the events generated by students. Each state s contains the number of students whose logged sequences of events have passed through that state, which is denoted by γ(s). The support of a state s is defined as the rate of γ(s) with respect to the total number of students in the same cluster. Likewise, each transition t also contains its student frequency, denoted by φ(t), which is the number of logged sequences of events have passed through that transition. From this frequency, the confidence of the transition t is defined as the rate of φ(t) with respect to γ(s1) where s1 is the source state in transition t. Figure 3 denotes support and confidence as percentages in the states and the transitions, respectively.

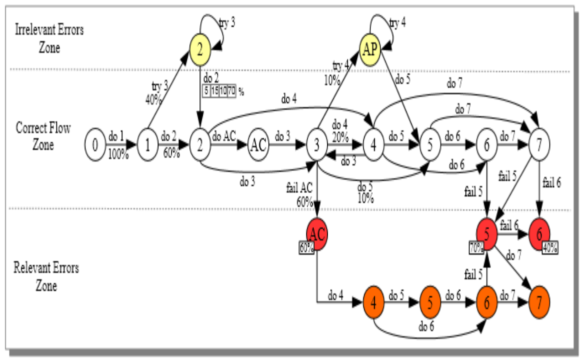

Figure 3. Example automaton

Correct Flow Zone:

This area includes the states that constitute the valid sequences of actions for an exercise, which logically end up with a satisfactory final state. These states are depicted by white circles in Figure 3.

Irrelevant Errors Zone :

This area groups states derived from error events that do not influence the final result. These error events are associated with attempted actions blocked by the tutor. States in this zone are depicted by yellow circles in Figure 3. As a student may attempt the same wrong action more than once consecutively, vector transitions are employed for outgoing transitions from states of this zone to represent event frequencies. Therefore, the first position of the vector contains the number of students that exit the state without ever repeating the wrong action; the second position contains the number of students that exit the state after repeating the wrong action once and so forth. For example, Figure 3 illustrates the vector transition from yellow state2 to white state2 for students that reached state 2 by attempted action 3, where 5% exited that state without repeating the attempted action “try 3” and 15% exited that state after repeating the attempted action “try 3” once

Relevant Errors Zone

This area comprises states derived from error events that actually influence the final result, i.e., if an event of this type occurs the final result will be wrong, unless are pair action is taken. If the error is not repaired, it will be propagated to the subsequent states, which will also be considered erroneous, no matter what the student does afterwards (unless it is a repair action). The states derived directly from relevant errors are depicted by red circles, and the subsequent states that are the result of applying a right action to a state in this zone are denoted by orange circles.

Example Automaton:

As an illustrative case of this robot, we take an extract from the biotechnology virtual research center convention, which is appeared in Figure 4. Figure 4 demonstrates a few activities, all of which, aside from two, must be executed in a specific arrange. These two activities can be performed in any request, however both should dependably be executed before activity 5 and after activity 2. Figure 3 demonstrates a case of the broadened robot portrayed previously. This machine has been made from a subset of the logs of understudies who have played out the procedure appeared in Figure 4 inside the virtual research facility for educating biotechnology. Thus, it reflects every one of the activities and blunders that understudies have performed in this piece of the reasonable task (we have excluded the frequencies of all occasions for clearness). Figure 3 indicates how a few occasions related with right activities recorded in understudy logs deliver rectify states. For example, event “do 1” (drop the glass on the shaker) leads to state 1, and event “do 3” leads to state 3. In contrast, the event that leads to yellow state 2 does not represent a right action, but the error event of trying to do action 3 instead of action 2. Likewise, yellow state AP (already performed) is the result of another irrelevant error, i.e.,trying to do action 4 when it has already been performed previously, because some students performed do 4 → do 3 from white state 2. The “do AC” event, which is not illustrated in Figure 4, represents the action of Adding Casein (a protein) to the mix. This action is considered incompatible with the other actions, but, for pedagogical reasons, the laboratory makes more chemical products available to students than required for the mix. Hence, as soon as the next chemical product is added to the mix (action 3 is performed), this event is validated as in correct producing a “failAC” event that leads to a red AC state. As Figure 3 shows, some students completed the remainder of the practical assignment correctly after performing “do AC” (path of orange states to orange state 7) As regards relevant errors, red state 5 is caused by the fail event of not doing the right action 5 next, because some students performed action 6 or action 7 too early. For example, the automaton represents the fact that some students mistakenly performed action 6 instead of action 5 by the path: do 6 → fail 5. As explained above, a wrong action can cause more than one error event at the same time in a student log. For example, if some students mistakenly perform action 7 instead of action 5, it will be represented by the path: do 7 → fail 5 → fail 6, because students have skipped two consecutive right actions. The frequencies in Figure 3 show that many students forgot to turn on the shaker before adding substances to the mix (try 3 event labelled with 40%), or that most students forgot to adjust the mix pH (red state 5 labelled with 70%).In view of the error rates in these two cases, it might be a good idea to set hints at state 1 and/or the two states 4 if the instructor wants to stop most students from making these mistakes. As explained previously, a different automaton is built for each cluster. Therefore, the frequencies of this example may not be the same as for the automaton of another cluster. Namely, red state 5 may not have such a high frequency or may not even exist. In this way, the tutoring feedback for students of different clusters may differ.

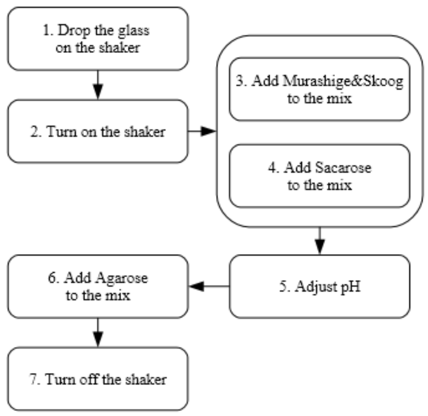

Figure 4. Example of biotechnology lab process

IV. CONCLUSIONS

This paper introduces a model that can foresee understudy actions in procedural training environments. Additionally, this paper clarifies how this model is coordinated into an ITS engineering and how it can be utilized to enhance the mentoring criticism by suspecting understudy blunders as long as this is academically advantageous. The collective student model is created from student logs by bunching logs and registering a broadened robot for each subsequent group. We should feature that there are few ITSs in the literature that rely on data mining techniques to improve their mentoring criticism. The proposed demonstrate has been approved utilizing the understudy logs gathered in a 3D virtual research facility for instructing biotechnology. Because of this approval, we presumed that the model can give sensibly great forecasts and support coaching input that is more adjusted to every understudy compose. An application that displays the collective student model would be extremely valuable for encouraging the definition of the coaching system. Along these lines, the teacher could imagine when understudies commit more errors or which part of the down to earth task understudies find simpler. In view of this data, the teacher could choose where and what coaching input the ITS ought to give. Moreover, this could likewise help the educator to enhance his or her own particular instructing. A different line of future work will be to approve an ITS based upon the proposed show with a specific end goal to assess the mentoring input incited by the proposed display.

V. 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]. J. Clemente, "Una Propuesta de Modelado del EstudianteBasadaen Ontologıas y DiagnosticoPedagogico-Cognitivo no Monotono," Ph.D. dissertation, Universidad Politecnica de Madrid, 2011.

[6]. M. Rico, J. Ramırez, D. Riofrıo, M. Berrocal-Lobo, and A. De Antonio, "An architecture for virtual labs in engineering education," in Global Engineering Education Conference (EDUCON), 2012 IEEE, 2012, pp. 1–5.

[7]. 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.