1-2018

Modeling engagement of programming students using unsupervised machine learning technique

Hua Leong FWA
Singapore Management University, hlfwa@smu.edu.sg

Lindsay MARSHALL

Citation

Modeling engagement of programming students using unsupervised machine learning technique

Hua Leong Fwa
Newcastle University
Newcastle upon Tyne, United Kingdom
h.l.fwa1@newcastle.ac.uk

Dr Lindsay Marshall
Newcastle University
Newcastle upon Tyne, United Kingdom
lindsay.marshall@newcastle.ac.uk

Abstract—**Engagement is instrumental to students' learning and academic achievements. In this study, we model the engagement states of students who are working on programming exercises in an intelligent tutoring system. Head pose, keystrokes and action logs of students automatically captured within the tutoring system are fed into a Hidden Markov Model for inferring the engagement states of students. With the modeling of students' engagement on a moment by moment basis, intervention measures can be initiated automatically by the system when necessary to optimize the students' learning. This study is also one of the few studies that bypass the need for human data labeling by using unsupervised machine learning techniques to model engagement states.**

Keywords—unsupervised, machine learning, engagement, intelligent tutoring, sensors

I. INTRODUCTION

Most would agree that engagement is critical to students' learning and achievement. This correlation between engagement and learning achievement is evidenced by many studies [1-3], and high levels of motivation and engagement have been consistently linked to higher probabilities of student success [4, 5]. Observing the significance of engagement, an increasing number of higher educational institutions routinely conduct engagement surveys so that they can take appropriate action when they identify areas of student engagement which require attention. Some examples of such surveys include the National Survey of Student Engagement (NSSE) [6] and Australasian Survey of Student Engagement (AUSSE) [7], both of which are nation-wide surveys of college students in the United States and Australia respectively. The studies which linked student engagement to desirable learning outcomes and the regular barometric testing of students' engagement in institutions point undeniably to the value of engagement and its role in influencing positive educational outcomes.

Engagement, just like any other affect, is a complex and multi-faceted construct which makes it a challenge to measure. In the literature, one approach to measuring it is through the behavioral perspective. Through this perspective, student engagement is defined as 'the time and effort students devote to educationally purposeful activities' [8] where time-on-task or effort is the amount of time that students actually spend on learning [9].

Motivation engenders engagement [10] and one approach to measuring engagement would be first to define what causes motivation. The ARCS model, proposed by Keller [11], is one such model that encapsulates the behavioral perspective of learning motivation. It defines that the four conditions of Attention, Relevance, Confidence and Satisfaction (ARCS) are prerequisites of motivation. The condition of Attention refers to the gaining and sustaining of students' attention in learning. Relevance points not only to the material need for students to know the relevance of what they are learning but more importantly, whether they perceive what they are learning as important and relevant. This perception of relevance in turn comes from the feeling of enjoyment that results when, for example, people in need of affiliation work in groups. Alternatively, it can be from the enjoyment that people who are high in need of achievement derive when they take responsibility for achieving a moderately challenging task that they personally set. Differences in confidence influence students' persistence and accomplishment while satisfaction results from the intrinsic and extrinsic rewards from learning. Confidence builds up in situations where students experience some level of success when they exert an adequate amount of effort. Thus, from the ARCS model, one can conclude that effort, confidence, satisfaction and attention are important measures of learning motivation and engagement.

Intelligent Tutoring Systems (ITSs) are built with the premise of providing learners with one to one tutoring support automatically and cost effectively by acting as a personal training assistant that assesses one's knowledge continuously. ITSs provide assistance when they detect that one requires help to move on. A criticism of ITSs is that they still underperform when compared to one-to-one human tutoring. This is attributed to the fact that ITSs are impoverished in emotional awareness and empathy as opposed to human tutors who are able to adapt their tutoring based on both the cognitive and affective responses of the learner. This has spawned much research [12-15] that investigates augmenting ITSs with the capability to sense the affective states of the learner. In most of these studies [16, 17], the technique of supervised machine learning is used to map learners' actions or sensor measures into the inferred affect. The supervised machine learning approach, however requires the labeling of the data beforehand by subject matter experts. As this is done manually, the labeling process is tedious and prone to human errors. Moreover, supervised machine learning requires the *a priori*

definition of the classification categories even in situations where we have limited knowledge of what the categories and their boundaries are.

Unsupervised machine learning, on the other hand, does not require labeled data. The provided data set consists of just the input features without the corresponding labels or class. Using unsupervised learning, one can either discover groups of similar examples within the data also known as clustering or infers a function that approximates the distribution of data also known as density estimation [18]. To circumvent the need for tedious human labeling, unsupervised machine learning is used in this study to discover and group students with similar learning behaviors and also to associate different levels of engagement to these behavioral groups.

There were previous studies [19, 20] that use unsupervised mining to uncover engagement of students in tutoring systems through the use of only students' action logs within the tutoring systems. The key difference between this study and the previous studies is that this study tracks the engagement of students not in the domain of a mathematics tutoring system but in the domain of a computer programming tutoring system. Intrinsically, solving a programming exercise involves a longer interaction session and the interpretation of syntax and logic errors for the resolution of programming bugs. Another difference is that this study uses not only action logs but also sensor logs such as keystroke logs and head posture logs from web cameras - the use of a combination of different sensors has been advocated as being superior in terms of detection accuracy and availability as compared to using only a single sensor [21].

Although the use of sensors for educational data mining is often associated with issues such as obtrusiveness and difficulty of scaling due to their fragility, high cost and long setup time, the sensors proposed in this study (web cameras and keyboards) are unobtrusive, available at a low cost and easy to set up. With the modeling of students' engagement on a moment by moment basis in a computer programming tutoring system, intervention measures can be initiated automatically by the system when necessary to optimize the learning of students.

II. Setup

The data set used in this study comprises of 14 students from Nanyang Polytechnic, Singapore who were recruited in the academic year 2015 and 2016. The study was conducted in computer labs where students were enrolled to work on programming exercises in a Java programming tutoring software for an average duration of 50 minutes. The tutoring software which was developed by the author has a total of 6 topics and a set of 2 programming exercises per topic, giving a total of 12 exercises to be completed. The topics cover the basics of Java programming and include the use of variables, loops, conditional statements and arrays. A more detailed description of the setup can be found in [22].

The software that was used to track head motion is developed by xLabs [23]. It was used for extracting raw head pose information such as horizontal and vertical head position, head roll, pitch and yaw from the web camera. The students work on the exercises in the tutoring software while their eye gaze, keystrokes and head motion are logged. The eye gaze and head motion are captured by the xLabs software for translating into the raw head pose and eye gaze information. A JavaScript function was developed to send these raw head pose and eye gaze information for logging to physical files on the web server. The logged eye gaze data is however, not used in this study.

III. Modeling

The following sections detail the modeling of a learner's task engagement.

A. Head Pose

The captured head pose information is used for gauging the attention of students. Head pose can be used to approximate the eye gaze direction as proposed by Stiefelhagen and Zhu [24].The head yaw angle is used to track whether a student is looking at the screen or not. If the head yaw angle is more than ten degrees to the left or right, the student is deemed to be not looking at the screen and thus inattentive. In our setup, the web camera is mounted at an average distance of 60 cm from the student and a length of 10.5 cm to the left or right from the center of the computer monitor is considered to be off-screen. This is shown in Figure 1. Using the formula in (1) below, the yaw angle limit θ is calculated to be 10 degrees or 0.175 radians.

$$\theta = \tan^{-1} \frac{10.5}{60} \qquad (1)$$

The head pitch angle measures the vertical angle of the head motion. Although a student who is looking down and not at the screen is considered to be off-screen as well, the head pitch angle is not used (only head yaw angle is used). The reason is that, from observations, some student participants looked down at the keyboard while typing. A student who is typing program code and looking down at the keyboard while typing should not be considered as being inattentive. As such, using the pitch angle as part of the criteria for determining
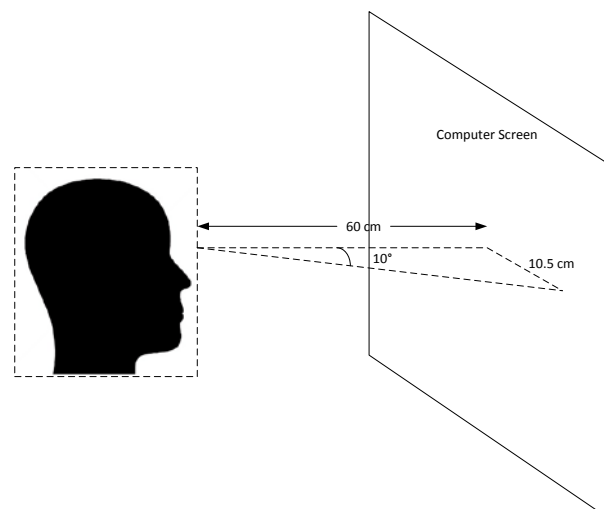

Figure 1. Calculation of head yaw angle for off-screen

whether the student is looking off-screen and possibly

inattentive would not be accurate. Thus, only the yaw angle is used for determining whether the student is looking off-screen.

B.   Other features

Other than the head motion logs, the keystrokes, the compilation error details and the number of exercises completed are also logged. The features derived from these logs include the number of keystrokes, the maximum keystroke pause duration, the number of compilation attempts, the number of compilation errors and the off-screen duration from head motion. These features are then aggregated into time window slice width of 60 seconds. They are then accumulated into a sequence of feature vectors, yielding a total of 1345 rows of feature vectors. These feature vectors are then normalized using the formula $(X - \bar{X}) / \sigma$ where $\bar{X}$ is the mean and σ is the standard deviation. The number of keystrokes, maximum keystroke pause duration and the number of compilation attempts serve as a surrogate measure for the effort put in by the student, the number of compilation errors as a surrogate measure for the obstacles encountered and confidence in tackling these obstacles and lastly, the off-screen duration as a surrogate measure for the attention level of the student. An intuition is that if the student is confident in resolving the compilation errors, it will be reflected as a decrease in the number of compilation errors in the next time period.

The system will log each instance in which a key is depressed as well as the time at which it is depressed. The pause duration between each key is calculated as the time difference between the depressing of 2 consecutive keys and the maximum keystroke pause duration is the maximum of these pause durations within an entire time window slice of 60 seconds.

C. Hidden Markov Model

The temporal-sequential aspect of learning data is an important aspect in the modeling of learning [20]. The actions of students in a tutoring system are usually time ordered and the sequence of these actions does offer a valuable trove of information into modeling the learning process. For example, comparing two students who solved a programming exercise in the same amount of time, the learning mileage will differ depending on the trajectory of resolving the problem. One student may request hints for every failed compilation attempt with no effort put into deciphering the compilation errors while another may solve the problem by carefully deciphering the compilation errors and resolving them without the use of hints.

This temporal sequential aspect is best modeled by the Hidden Markov Model (HMM). The HMM used for this study was first popularized by Rabiner and Juang [25]. It has since been used in diverse applications such as speech recognition [26], financial stock market prediction [27] and human action recognition [28]. HMMs consist of stochastic Markov chains based on a set of hidden states whose values cannot be directly observed with the relationship between a hidden state and the actual observations being modeled with a probability distribution. HMMs adhere to the Markov property which states that the state of a model at time t is only dependent on

the state of the model at time t-1 and not on other prior states such that $P(S_j^{t+1} \mid S_i^t) = P(S_j^{t+1} \mid S_i^t, S_m^{t-1}...S_n^0)$.

A HMM is described by the tuple $\{S, O, A, B, \Pi\}$ where

N : number of hidden states S

$S = \{S_1, S_2, ..., S_N\}$

M: number of observation symbols O

$O = \{O_1, O_2, ..., O_M\}$

$A = \{a_{ij}\}$, where $a_{ij} = P(S_j^{t+1}, S_i^t); i, j = 1, ..., N$

$B = \{b_i(k)\}$, where $b_i(k) = P(O_k \mid S_i)$

The model parameters are valid probabilities that must satisfy the following constraints:

$$\sum_j^N a_{ij} = 1, \quad \sum_k^M b_j(k) = 1, \quad \sum_j^N a_{ij} = 1, \quad \sum_k^M b_j(k) = 1$$

The probability of being in state i at time t=0 is given by $\Pi = \{\Pi_i\}$, where $\Pi_i = P(S_i^0)$.

HMMs can be used in unsupervised learning. In unsupervised learning, the model parameters are learned by maximizing $\sum_S \log(P(O^S \mid H))$, the sum of the posterior log-likelihoods of each training sequence O^S using a form of expectation maximization (EM) called the Baum-Welch algorithm.

The observations in this study are continuous. Although it is possible to discretize the continuous valued features, it will result in a loss of information. In this model, the model parameter B cannot be represented as a matrix of point probabilities but will have to be represented as a probability distribution function (pdf) over the continuous observation space for each state. The pdf used is the Gaussian Mixture Model [29].

In this study, the engagement levels of students are modeled as hidden states of the HMM using the HMM toolbox [30]. The HMM is then fitted to the action and sensor logs of students.

D. Unsupervised Learning

In the supervised learning mode of the HMM, the hidden states need to be annotated. The annotated states serve as the ground truth for inferring the parameters of the model. As such, using the HMM in supervised learning mode requires substantial human effort to annotate the data set. In addition, in cases where it is infeasible to infer the hidden states through human observation, supervised learning will not be possible. In consideration of these, we use the unsupervised learning mode of HMM to model the engagement levels of students in this study.

Each student observation is made up of the tuple < number of keystrokes, the maximum keystroke pause duration, the number of compilation attempts, the number of compilation errors and the off-screen duration>. The unsupervised learning method requires that one determines the number of hidden states that the model should have through model selection [31]. One model selection technique is the use of Leave-One-Out Cross-Validation (LOOCV). The algorithm to determine the ideal number of hidden states is listed below.

Algorithm 1 To determine ideal number of hidden states

Input: data_set

Output: log_likelihood_state

1. **for** states = 1 to 5 do

2. **for** validatedStudent=1 to 14 do

3. testing_data=data_set(validatedStudent);

4. training_data=data_set(~validatedStudent);

5. **for** runs=1 to 10 do

6. hmm_params_set = hmm_em_learn(training_data);

7. log_likelihood_run(states, validatedStudent,runs) = hmm_log_probability(test_data, hmm_params_set);

8. **end for**

9. log_likelihood_student(states, validatedStudent) = max_over_runs(log_likelihood_run(states, validatedStudent));

10. **end for**

11. log_likelihood_state(states) = mean_over_students(log_likelihood_student(states));

12. **end for**

13. **return** log_likelihood_state;

hmm_em_learn is the procedure used to learn the hmm model parameters using Baum_Welch learning technique.

hmm_log_probability is the procedure that calculates the log probability of the data given the hmm model.

max_over_runs is the procedure that calculates the maximum of log likelihoods over the various runs.

mean_over_students is the procedure that calculates the mean of log likelihood over all the students.

Models ranging from 1 to 5 hidden states were trained over 10 runs using the Baum Welch algorithm (for 30 iterations) with different random initial model parameters. The outer loop performs the LOOCV by leaving a different student's data out of training each time. The left out student's data set is then used as the test data set for evaluation of the model's log likelihood. Algorithm 1 returns the log likelihood values for the various number of hidden states. We then derived the Akaike Information Criteria (AIC) from the log-likelihood values and the number of parameters in the model. The AIC provides an estimate of the measure of fit of the model [32]. By plotting the AIC values against the number of hidden states, we determined the optimal number of hidden states to be 3 (the point with the lowest AIC value).

IV. RESULTS AND DISCUSSION

The mean values of the learned model's features are shown in Table I. As seen in Table I, State 1 is characterized by a moderate number of compilations, high number of keystrokes, low number of compilation errors and a low off-screen duration. This suggests that students in State 1 are engaged as they are actively trying to compile and solve the programming exercises (high effort as evidenced by the low maximum keystrokes pause duration and high number of keystrokes) and are maintaining a high level of attention on the screen (as evidenced by the low off-screen duration). Students in State 2 are characterized by zero number of compilations, high maximum keystroke pause duration and close to zero number of keystrokes but yet low off-screen duration. It can be inferred that students in State 2 are just starting out working on the exercises. It could also be probable that students in State 2 are clueless and do not know how to start on the exercises if they have been working on the exercises for a while. State 3 is characterized by a low number of keystrokes, high maximum keystroke pause duration, a moderate number of compilations and high off-screen duration. It seems that the students in State 3 are disengaged and not actively working on the exercises as evidenced by their high off-screen duration (low attention), low number of keystrokes and high maximum keystroke pause duration (low effort).

The state transition matrix is shown in Table II. From the state transition matrix, students in State 1 (the engaged state) and State 2 (the starting out state) are more likely to persist in their current states than to transit to other states. This can be seen from the higher probabilities of persisting in the original state (e.g. students in States 1 and 2 will tend to persist in States 1 and 2 with a probability of 0.72 and 0.68 respectively). This suggests that a student who is engaged will likely stay engaged. A student who is in State 3 (the disengaged state) is slightly more likely to transit to State 1 (with a probability of 0.40) than to transit to State 2 (with a probability of 0.29). The student in State 3 is also less likely to persist in the disengaged state as seen from the probability of 0.31. The high probability of a student in the starting out state persisting in the same state

TABLE I. MEAN VALUES OF FEATURES FOR THE VARIOUS HIDDEN STATES

Features	Hidden States		
	State 1	**State 2**	**State 3**
No. of compilations	1.35	0.28	1.47
No. of keystrokes	32.31	1.82	7.22
Max pause duration	10.57	56.39	46.95
No. of errors	0.42	0.88	0.65
Off-screen duration (in milliseconds)	285.90	283.23	8140.42

TABLE II. PROBABILITIES OF STATE TRANSITIONS

Transition Matrix	State1	State 2	State 3
State 1	0.72	0.12	0.16
State 2	0.23	0.68	0.09
State 3	0.40	0.29	0.31

as well as the one-third chance that a disengaged student will stay disengaged suggest opportunities for pedagogical interventions to shift the student into the more desirable and potentially sticky engaged state. Although there is only a one third chance that a student who is disengaged will stay disengaged, it would be beneficial if we can lower this probability further with the provision of tutorial support.

V. CONCLUSION

In this study, unsupervised learning using HMM is employed to infer the engagement states of students from both students' actions within the tutoring system as well as from sensors such as web cameras (for head postures) and keyboard (for keystrokes). As engagement is critical to students' learning and achievement, the inference of engagement states allow us to intervene at opportune moments to sustain the students' engagement. The use of HMM allows for the time sequenced students' responses comprising of their actions and head postures to be translated into hidden engagement states and their transition probabilities.

The results show that the engagement states of students can be clustered into 3 distinct states – the engaged state (State 1), the starting out state (State 2) and the disengaged state (State 3). In addition, students who are engaged tend to remain engaged while students who are disengaged are more likely to switch to the other states. These suggest the potential for the use of learning interventions such as hints or scaffolding to shift the students into the more enduring engaged state. This hypothesis will however require further experiments to validate.

REFERENCES

[1] G. D. Kuh, S. Hu, and N. Vesper, ""They shall be known by what they do": An activities-based typology of college students," Journal of College Student Development, vol. 41, pp. 228-244, 2000.

[2] A. W. Astin, Preventing students from dropping out: San Francisco: Jossey-Bass, 1975.

[3] A. W. Astin, What matters in college?: Four critical years revisited vol. 1: Jossey-Bass San Francisco, 1993.

[4] W. E. Blank and S. Harwell, "Promising Practices for Connecting High School to the Real World," pp. 15-21, 1997.

[5] P. C. Dev, "Intrinsic Motivation and Academic Achievement What Does Their Relationship Imply for the Classroom Teacher?," Remedial and Special Education, vol. 18, pp. 12-19, 1997.

[6] G. D. Kuh, "Assessing what really matters to student learning inside the national survey of student engagement," Change: The Magazine of Higher Learning, vol. 33, pp. 10-17, 2001.

[7] H. Coates, "Development of the Australasian survey of student engagement (AUSSE)," Higher Education, vol. 60, pp. 1-17, 2010.

[8] A. Radloff and H. Coates, "Doing more for learning: enhancing engagement and outcomes: Australasian Survey of Student Engagement: Australasian Student Engagement Report," 2010.

[9] R. E. Slavin and N. Davis, "Educational psychology: Theory and practice," 2006.

[10] S. Saeed and D. Zyngier, "How motivation influences student engagement: A qualitative case study," Journal of Education and Learning, vol. 1, p. 252, 2012.

[11] J. M. Keller, "Development and use of the ARCS model of instructional design," Journal of instructional development, vol. 10, pp. 2-10, 1987.

[12] S. K. D'Mello and A. Graesser, "Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features," User Modelling and User-Adapted Interaction, vol. 20, pp. 147-187, 2010.

[13] S. Alexander, A. Sarrafzadeh, and S. Hill, "Easy with Eve: a functional affective tutoring system," presented at the Workshop on Motivational and Affective Issues in ITS. 8th International Conference on ITS, 2006.

[14] N. Jaques, C. Conati, J. M. Harley, and R. Azevedo, "Predicting affect from gaze data during interaction with an intelligent tutoring system," in International Conference on Intelligent Tutoring Systems, 2014, pp. 29-38.

[15] C. Conati and H. Maclaren, "Empirically building and evaluating a probabilistic model of user affect," User Modeling and User-Adapted Interaction, vol. 19, pp. 267-303, 2009.

[16] C. Merten and C. Conati, "Eye-tracking to model and adapt to user meta-cognition in intelligent learning environments," in Proceedings of the 11th international conference on Intelligent user interfaces, 2006, pp. 39-46.

[17] S. W. Mcquiggan, S. Lee, and J. C. Lester, "Early prediction of student frustration," in Affective Computing and Intelligent Interaction, ed: Springer, 2007, pp. 698-709.

[18] C. Bishop, "Pattern Recognition and Machine Learning (Information Science and Statistics), 1st edn. 2006. corr. 2nd printing edn," ed: Springer, New York, 2007.

[19] C. Beal, S. Mitra, and P. R. Cohen, "Modeling learning patterns of students with a tutoring system using Hidden Markov Models," presented at the Proceedings of the 2007 conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work, 2007.

[20] B. Shih, K. R. Koedinger, and R. Scheines, "Discovery of Student Strategies using Hidden Markov Model Clustering," in the Proceedings of the 6th International Conference on Educational Data Mining, 2010.

[21] S. K. D'mello and J. Kory, "A review and meta-analysis of multimodal affect detection systems," ACM Computing Surveys (CSUR), vol. 47, p. 43, 2015.

[22] F. H. Leong, "Automatic detection of frustration of novice programmers from contextual and keystroke logs," in Computer Science & Education (ICCSE), 2015 10th International Conference on, 2015, pp. 373-377.

[23] (2016). xLabs eye, gaze and head tracking via webcam. Available: https://xlabsgaze.com/

[24] R. Stiefelhagen and J. Zhu, "Head orientation and gaze direction in meetings," in CHI'02 Extended Abstracts on Human Factors in Computing Systems, 2002, pp. 858-859.

[25] L. Rabiner and B. Juang, "An introduction to hidden Markov models," ieee assp magazine, vol. 3, pp. 4-16, 1986.

[26] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proceedings of the IEEE, vol. 77, pp. 257-286, 1989.

[27] M. R. Hassan and B. Nath, "Stock market forecasting using hidden Markov model: a new approach," in 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), 2005, pp. 192-196.

[28] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden markov model," in Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on, 1992, pp. 379-385.

[29] R. C. Rose and D. A. Reynolds, "Text independent speaker identification using automatic acoustic segmentation," in Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on, 1990, pp. 293-296.

[30] (2016). Hidden Markov Model (HMM) Toolbox for Matlab. Available: http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html

[31] G. Celeux and J.-B. Durand, "Selecting hidden Markov model state number with cross-validated likelihood," Computational Statistics, vol. 23, pp. 541-564, 2008.

[32] H. Akaike, "A new look at the statistical model identification," IEEE transactions on automatic control, vol. 19, pp. 716-723, 1974.

Hua Leong Fwa is a Senior Lecturer with Nanyang Polytechnic Singapore. He is also concurrently a PhD candidate with the Department of Computing Science in Newcastle University, Newcastle upon Tyne, United Kingdom. His research interests include affective computing, learning analytics and the use of machine learning techniques for inferring complex learning states.

Lindsay Marshall is a Senior Lecturer with the Newcastle University, Newcastle upon Tyne, United Kingdom. He completed his PhD in Computing Science from the same university in 1980. His research interests include pedagogies such as problem based learning and active learning as well as assessment techniques in computer science education.