A Theory-Driven Approach to Predict Frustration in an ITS

Ramkumar Rajendran, Sridhar Iyer, Sahana Murthy, Campbell Wilson, Judithe Sheard

Abstract—The importance of affect in learning has led many Intelligent Tutoring Systems (ITS) to include learners' affective states in their student models. The approaches used to identify affective states include human observation, self-reporting, data from physical sensors, modeling affective states, and mining students' data in log files. Among these, data-mining and modeling affective states offers the most feasible approach in real world settings, which may involve a huge number of students. Systems using data-mining approaches to predict frustration have reported high accuracy, while systems that predict frustration by modeling affective states, not only predict a student's affective state but also the reason for that state. In our approach we combine these approaches. We begin with the theoretical definition of frustration, and operationalize it as a linear regression model by selecting and appropriately combining features from log file data. We illustrate our approach by modeling the learners' frustration in Mindspark, a mathematics ITS with large-scale deployment. We validate our model by independent human observation. Our approach shows comparable results to existing data-mining approaches and also the clear interpretation of the reasons for the learners' frustration.

Index Terms—Intelligent tutoring system, Affective states, Modeling Frustration, Frustration Theory.

1 Introduction

An Intelligent Tutoring System (ITS) provides personalized learning content to students based on their needs and preferences. An ITS consists of the learning content, the student model and the adaptation engine. Student models are constructed from the log files available in the ITS. The students' interaction with ITS, such as responses to questions, number of attempts at a task, and the time taken for various activities (such as responding or reading) are captured in the ITS log file. Student models also typically contain information such as the students' previous knowledge and background [1], from which it is possible to infer the students' cognitive states. However, it is now well established that the learning process involves both cognitive and affective processes [2], [3], and the consideration of affective processes has been shown to achieve higher learning outcomes [4], [2]. The importance of a student's affective component in learning has led ITS to include learners' affective states in their student models. Baker et. al. [5] have suggested that the relevant affective states of students interacting with ITS are boredom, frustration, confusion, delight, engaged concentration and surprise. In this paper we focus on frustration.

To include affective states in the student model, the students' affective states should be identified while they interact with the ITS. Predicting the students' affective states, that is, attempting to determine these states while students interact with the system, is a challenging problem in education research, and is the focus of several current research efforts [6], [7]. Methods that have been implemented in ITS to predict the affective state include human observation [5], [8], [9], learners' self-reported data of their affective state [10], [11], mining the system's log file [12], [13], modeling affective states [11], [14], face-based emotion recognition systems [4], [3], analyzing the data from physical sensors [15], [16], [10], and more recently, sensing devices such as physiological sensors [17], [18]. Advances in these methods look promising in a lab setting. However, they are not yet feasible in a large scale, real-world scenario to predict affective states [9]. The exceptions are data-mining approaches and modeling affective states, which have additional benefits. Existing systems which use data-mining approaches have reported high accuracy in predicting frustration. On the other hand, the advantage of systems which are based on modeling of affective states is that they not only predict the affective state of the learner, but also shed light on the cause for that state.

In this paper, we propose an approach to identify the students' frustration in an ITS, inspired by the better accuracy of data-mining approaches, and the use of theory in modeling affective states. We develop a model that predicts a student's frustration while s/he interacts with an ITS. The model is derived from a theoretical definition of frustration based on the analysis of goal-blocking events, by selecting and appropriately combining the features in the ITS log file. The constructed features are used to form a linear regression model to predict frustration. The model helps us to understand

- *R. Rajendran is with IITB-Monash Research Academy, Indian Institute of Technology, Bombay, India and Monash University, Melbourne, Australia. E-mail: ramkumar.r@iitb.ac.in.*
- *S. Iyer is with Department of Computer Science and Engineering and S. Murthy is with Education Technology, Indian Institute of Technology, Bombay, India. E-mail: sri, sahanamurthy@iitb.ac.in*
- *C. Wilson and J. Sheard are with Faculty of Information Technology, Monash University, Melbourne, Australia. E-mail: campbell.wilson, Judy.Sheard@monash.edu*

Manuscript received on Mon xx, Year;revised Mon xx, Year.

the contribution of each feature towards the cause of a student's frustration. Our approach shows comparable results to existing data-mining approaches.

We illustrate our approach by applying it to an ITS for mathematics, Mindspark, which we describe in Section 2. Related works are reviewed in Section 3. The design decisions of our approach are detailed in Section 4. We then discuss our approach in Section 5. In Section 6, we show how to apply our approach to Mindspark. An independent method to validate our model is detailed in Section 7. The results of our theory-driven approach are shown in Section 8 and its comparison with other approaches on Mindspark data is shown in Section 9.

2 System: Mindspark

Mindspark, a commercial mathematics ITS developed by Educational Initiatives India (EI-India)[1], is used in our research to implement and test our theory-driven approach. Mindspark has been incorporated into the school curriculum for different age groups (grades 3 to 8) of students [19]. Mindspark is currently implemented in sixty schools and is being used by 30,000 students on an average of four sessions per week, with each session ranging from 25 to 30 minutes.

Mindspark is a computer based self-learning system, in which students learn mathematics by answering questions posed by the system. Mindspark provides detailed feedback and an explanation upon receiving the answer from the student. A sample question from Mindspark is shown in Fig. 1. Mindspark consists of a sequence of specially designed learning units (clusters), which contain questions on concepts that make up the topic. Each topic consists of questions of progressively increasing levels of complexity. Mindspark covers a wide range of topics in school level mathematics such as linear inequalities, matrices, quadratic equations, fractions, decimals, and polygons. Mindspark adaptation logic selects the questions to be presented based on a student's answer to a previous question and his/her overall performance in the topic, which allows the student to move at his/her own pace. If a student performs poorly in the current topic (for example, the student does not answer sufficient number of questions correctly), s/he will be moved to a lower complexity level in the same topic.

In Mindspark, if a student answers three consecutive questions correctly, s/he receives a Sparkie (extra motivational points), shown in the Fig. 1 at the top-right corner. If a student answers five consecutive questions correctly, then s/he receives a Challenge Question (tougher question from higher complexity level). If the student answers the Challenge Question correctly, s/he receives five Sparkies. Every week, the highest Sparkie collectors (Sparkie Champ) are identified and their names are published on the Mindspark website[2].

1. http://www.ei-india.com/
2. http://www.mindspark.in/login/

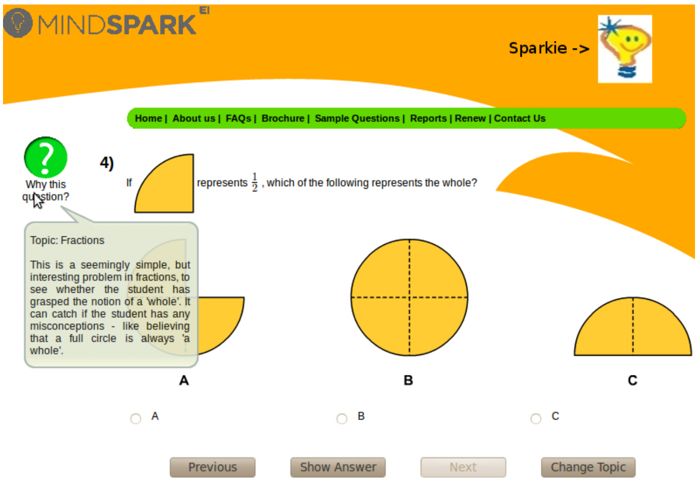


Fig. 1. A sample question from Mindspark on Fractions, Sample image of Sparkie is shown at the top-right corner of the figure

The student's interactions with Mindspark are recorded in a log file. The log file contains the following information: user ID, session ID, date, question number, topic code, the answer provided by the student, time taken to answer, result, and time taken to read the explanation. However, time taken to answer, result, and time taken to read the explanation are the only useful data from Mindspark log file to predict affective states. Hence, the Mindspark dataset contains a huge volume of data with limited information.

3 Related Work

To identify affective states, Baker et. al. [5] suggest three different methods: Human observation, hardware sensors and data-mining techniques using data from student log files. In addition, researchers have modeled affective states of users as they interact with a game or ITS [11], [14]. There are various research studies that identify the learners' affective states using data from physiological signals [20], such as electrocardiogram (ECG) [17], facial electromyography (EMG) [18], and galvanic skin response (GSR) [17]; from sensors, such as, blue eye camera [15], posture analysis seat, pressure mouse and skin conductance bracelet [2]; and from conversation clues [12]. A recent review paper on affective states detection [6] concluded that identification of affective states using sensor signals, facial expression, text analysis and voice analysis are widely researched compared to data-mining approaches.

In the rest of this section, we review existing approaches related to our research on modeling and identifying frustration from ITS log data. We review systems which identify the students' affective states from log files of students' interactions as well as systems which do so by modeling the affective states. We describe three systems, AutoTutor, a programming lab, and Crystal Island, which predict frustration based on data from log file. These systems model the student's affective states using the data form the log file and also using data from the biometrics, like bluej integrated development environment used in a programming lab experiment. However, in this related work section we are interested in data from log file and hence we describe only the data from log file in detail. The systems we describe which model affective states are: Crystal Island, and Prime Climb.

AutoTutor [12] is a dialogue-based tutoring system. In AutoTutor, the students' affective states such as frustration, boredom and confusion are identified based on the features from the log file, such as, response time, number of characters in a student's response, and tutor feedback to a student's response. The feature set is reduced from 17 to 11 by doing correlation analysis with affective states; the features which are significantly correlated are selected. To avoid redundancy among the selected features, dimensionality reduction techniques like principle component analysis (PCA) are used. The selected features are applied to 17 standard classifiers in Weka [21], a data-mining software, and the best results are reported. The maximum classification accuracy reported in this system [12] is 77.7%.

In a programming lab in [13], the students' average frustration in computer programming exercises across different labs is detected based on information from compiler data, such as the average time between compilations, and consecutive pairs of same error. The features are selected based on the Error Quotient (EQ) construct and researchers' knowledge of the system. The feature set is reduced from 11 to 4 by correlation analysis with affective states. The selected features are applied to a linear regression model to predict frustration. The reported regression value of this system is r = 0.3178 [13].

Crystal Island [22] is a task-oriented learning environment. In Crystal Island, the students' frustration is predicted using the data form log file. Based on the appraisal theory, the feature set contains all activities in the learning environment and also the data from physiological signals. This system uses non-sequential modeling techniques such as nave Bayes, decision trees, and support vector machines, and then reports the results. These techniques are implemented using data-mining software, Weka [21]. The best reported performance of this system [22] is accuracy of 88.8%, precision of 88.7% and Recall of 88.9%.

Affective state modeling in [14], creates a Dynamic Bayesian Network (DBN) model. This is then used to capture the users' affective states such as frustration and confusion. The users' affective states are identified when they interact with Crystal Island. The students self-report the affective state during the interaction with Crystal Island. The students' personal attributes such as mastery approach, environmental variables such as goals completed, and worksheet check are considered while creating the DBN model. The accuracy reported in this system [14] for frustration is 28% (for emotion prediction) and 56% (for valence prediction).

The system in [11] created a Dynamic Decision Network (DDN) model to capture the emotions described in OCC theory [23]. This system captures instantaneous emotions like joy/distress and pride/shame of the students when they interact with the educational game, Prime Climb. The users' personality traits and interaction pattern such as move quickly, fall often are considered while creating the model. The DDN model captures emotions like joy, distress and possible causes of it, such as student's goals (learn math, avoid falling, beat partner etc.), and goals satisfied (have fun satisfied, beat partner satisfied etc.). The system [11] reports accuracy to predict Joy is 69%, distress is 70%, reproach is 47%, and Admiration is 66%.

We note that accuracy in data-mining approaches (AutoTutor [12], and Crystal Island [22]) is in the range of 77% to 88%. While accuracy for emotions reported by using DBN [14] and DDN [11] model is comparatively less, 28% to 70%. The approaches used in modeling affective sate (Crystal Island [14], and Prime Clime [11]) captures not only the affective states but also why the user is in that state.

4 Design Decision for Our Approach

In this section we discuss the decision of choosing an affective state and a classifier model to predict frustration. To model frustration we considered different classifiers used in existing approaches, including dynamic Bayesian nets. However, to understand the contribution of each feature towards frustration we started with a linear regression model and later we have tried various other classifier models such as decision tree, support vector machine. And we found that the performance of the linear regression model are comparable with the other models' performance. Hence, we continue using the linear regression model in our approach. However, given the limited information available from Mindspark log data it is not easy to construct Decision Network, as done in [11] and [14]. The linear regression classifier model informs us of the factors contributing to frustration, and also helps us to determine which features contribute most to frustration. Thus the linear regression model can help us to address frustration systematically, and identify potential sources of frustration, thereby helping the students to avoid it.

Classical theories define frustration as an emotion caused by interference, preventing one from achieving a goal. For example, Rosenweig's Frustration Theory (1934) states that frustration is "the occurrence of an obstacle that prevented the satisfaction of a need" [24]. Similarly, according to Frustration Aggression Hypothesis (1939), frustration is the "condition which exists when a goal-response suffers interference" [24]. In recent years, the standard textbook 'Introduction to Psychology' (1986) [25] incorporates this perspective and define it: "Frustration refers to the blocking of a behavior directed towards a goal." Modern theories include sources of frustration other than goal blockage. These include Amsel's frustration theory (1990) [26] and OCC's Appraisal theory (1990) [23], which define frustration as primarily focused on student's goals and event outcomes. The theory based on goal-directed, problem-solving approach by Stein & Levine (1991) [27] explains the different factors which trigger frustration along with goal blockage.

In the research work done by JP Gee's, on learning through games [28], [29], reports that affective states like frustration and confusion are required while learning. Another research reports that, frustration may be less worrisome compared to other affective states such as boredom [5]. However frustration is the reason for the student's disengagement and eventually lead to attrition [30]. Hence, in our research, we consider frustration as a negative emotion, as it interferes with a student's desire to attain a goal. Also we focus on instances of frustration that occur due to goal blockage. Hence we consider the Morgan et. al.'s [25] definition of frustration.

5 Our Approach - Overview

In our approach, the selection and the combination of features from the ITS log file is done via a systematic and generic process based on an analysis of goal-blocking events. The sequence of steps is shown in Fig. 2. Guided by the theoretical definition (Step 1), we first identify the goals of the student with respect to their interaction with the ITS, and select the top n goals (Step 2). Based on information from the student log, a blocking factor, bf, for each of the n goals is identified (Step 3). For example, $goalj.bf$ represents the blocking factor for the $goalj$. We formulate a linear model for F_i, the frustration index at i^{th} question based on the blocking behaviors of student goals (Step 4). We apply a threshold to the frustration index to predict whether the student is frustrated or not. The weights of the linear regression are determined during training process (Step 5) with labeled data from human observation, an independent method to identify affective states. Performance of our model is validated by predicting frustration in test data and comparing results with human observation data (Step 6).

The linear regression model to predict frustration is given below:

$$F_i = \alpha[w_0 + w_1 * goal1.bf + w_2 * goal2.bf + ... \\ +w_n * goaln.bf + w_{n+1} * t_i] + (1-\alpha)[F_{i-1}] \quad (1)$$

In the above equation $w_0, w_1, w_2, ..., w_n$ are weights, which are determined by linear regression analysis, which is explained in Section 7.2. As explained in previous paragraph, the terms $goal1.bf$, $goal2.bf$, . . . , $goaln.bf$, are the blocking factors for goals $goal1$, $goal2$, . . . , $goaln$, respectively. The term t_i, the time spent by the student to answer the question i, is included on the basis of work done by Lazara et. al. [31], in a study to understand the causes of frustration in computer users. The experiment conducted by Lazara et. al. concluded that,

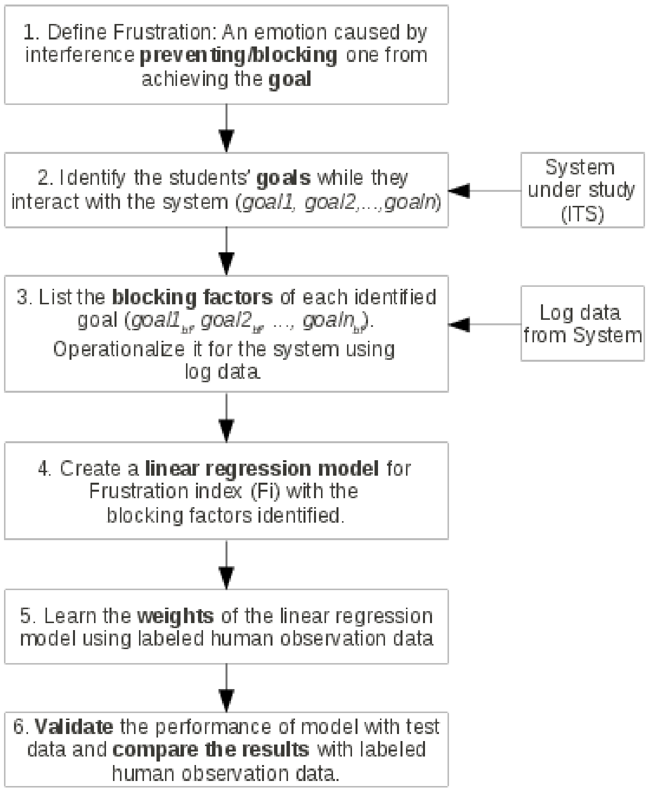


Fig. 2. Steps of theory-driven approach to create frustration model using data from log file

"tasks with higher importance that suffer from larger amounts of wasted time will lead to higher frustration levels". It implies that the time spent to achieve the goal should be considered while predicting frustration, though it is not given explicitly in frustration theories.

The last term in the equation, $(1-\alpha)[F_{i-1}]$ accounts for the cumulative effect of frustration. We include this term on the basis of [4], which states that frustration is cumulative in nature. The value of α, determines the contribution of frustration at $(i-1^{th})$ question to frustration at i^{th} question, α ranges from 0 to 1. We assume that the student is not frustrated at the beginning of their interaction with the ITS, and hence choose $F_i = 0$ for $i = 1, 2, 3$.

We restrict the scope of our approach to identify frustration that occurs due to the students' goal blockage while interacting with the ITS. We do not include frustration that might have occurred due to other factors external to the students' interactions with the ITS. Hence we are primarily concerned with how correct our prediction (precision) is, even though we might not able to find all the frustration instances that exist (recall).

6 Operationalization of Our Approach Applied to Mindspark Log Data

In this section we explain the application of our theory-driven approach to Mindspark data, in order to create the linear regression model (Equation 1).

6.1 Creation of Model for Mindspark

We create the linear regression model based on steps given in Fig. 2.

Step 1. Frustration definition: We begin with the definition of frustration from theory as 'an emotion caused by interference preventing/blocking one from achieving the goal'. The details were previously explained in Section 5.

Step 2. Students' Goals: We identified the four most common goals of students while interacting with Mindspark. To identify these goals, we conducted interviews with the staff of EI-India. Mindspark staff interviewed the students while they interacted with the ITS in their school. We recorded and transcribed the interviews, and analyzed the transcripts to identify student goals.

Step 3. Blocking Factors: The goals $goal1$, $goal2$, $goal3$, $goal4$ and the corresponding blocking factors $goal1.bf$, $goal2.bf$, $goal3.bf$, $goal4.bf$ are given in the Table 1. To model the blocking factor (bf) of each goal, we consider students' response to Mindspark questions, a feature captured in the Mindspark student log file.

TABLE 1
Student Goals and Blocking Factors for Mindspark

Student Goal	Blocking factor
$goal1$: To get the correct answer to the current question	$goal1.bf$: Answer to the current question is wrong
$goal2$: To get a Sparkie (answer 3 consecutive questions correctly)	$goal2a.bf$: Answers to previous two questions are correct and to current question is wrong $goal2b.bf$: Answer to previous question is correct and to current question is wrong
$goal3$: To reach the Challenge Question (answer 5 consecutive question correctly)	$goal3a.bf$: Answers to previous four questions are correct and to current question is wrong $goal3b.bf$: Answers to previous three questions are correct and to current question is wrong
$goal4$: To get the correct answer to the Challenge Question	$goal4.bf$: Answer to the Challenge Question is wrong

For $goal1$ of "*to get the correct answer to the current question*" the blocking factor is getting the wrong answer to the current question. We use a_i to represent the answer to the current question; $a_i = 1$ if correct, $a_i = 0$ if wrong. The blocking factor of $goal1$ is captured using:

$$goal1.bf = (1 - a_i) \tag{2}$$

For $goal2$, "*to get a Sparkie*" the student should answer three consecutive questions correctly. This goal can be blocked, if a student gets any question wrong in a sequence of three questions. Since the blocking factor by getting the wrong answer to current question is already addressed in $goal1.bf$, we consider only the blocking factor by getting the second and third answer wrong, in a sequence of three questions. Hence $goal2.bf$ has two components. One way by which $goal2$ can get blocked is, if the student answers the first two questions correctly in a sequence of three questions, and the third question wrongly. This is captured by blocking factor $goal2a.bf$:

$$goal2a.bf = (a_{i-2} * a_{i-1} * (1 - a_i)) \tag{3a}$$

The second way in which $goal2$ can get blocked is, if the student answers only the first question correctly in a sequence of three, and the second question wrongly. This is captured by blocking factor $goal2b.bf$:

$$goal2b.bf = a_{i-1} * (1 - a_i) \tag{3b}$$

The blocking factor of $goal2$ is:

$$goal2.bf = goal2a.bf + goal2b.bf \tag{4}$$

For $goal3$, "*to reach the Challenge Question*", the student should answer five consecutive questions, correctly. This goal can be blocked, if the student gets any question wrong in a sequence of five questions. Since the blocking factor obtained by getting the wrong answer to first, second and third question in a sequence of five questions is already addressed in $goal1.bf$, and $goal2.bf$, we consider only the blocking factor obtained by getting the fourth and fifth answer wrong in a sequence of five questions. Hence, $goal3.bf$ has two components. A way by which $goal3$ can get blocked is, if the student answers only the first four questions correctly in a sequence of five questions, and the fifth question wrongly. This is captured by $goal3a.bf$:

$$goal3a.bf = (a_{i-4} * a_{i-3} * a_{i-2} * a_{i-1} * (1 - a_i)) \tag{5a}$$

The second way in which $goal3$ can get blocked is, if the student answers only the first three questions correctly in a sequence of five questions, and the fourth question wrongly. This is captured by $goal3b.bf$:

$$goal3b.bf = (a_{i-3} * a_{i-2} * a_{i-1} * (1 - a_i)) \tag{5b}$$

The blocking factor of $goal3$ is:

$$goal3.bf = goal3a.bf + goal3b.bf \tag{6}$$

For $goal4$: "*To get the correct answer to the Challenge Question*" the blocking factor is getting the answer to the Challenge Question wrong. The blocking factor of $goal4$ is captured using:

$$goal4.bf = I * (1 - a_i) \tag{7}$$

where, I is the indicator for whether the current question is the Challenge Question or not. $I = 0$ for normal question and $I = 1$ for Challenge Question.

Step 4. Linear Regression Model: The mathematical model to predict frustration for Mindspark data is given in Equation 8, with the individual terms $goal1.bf$, $goal2.bf$,...,$goal4.bf$, being defined in Equations 2-7:

$$F_i = \alpha[w_0 + w_1 * goal1.bf + w_2 * goal2.bf + w_3 * goal3.bf + w_4 * goal4.bf + w_5 * t_i] + (1 - \alpha)[F_{i-1}] \quad (8)$$

Step 5 and Step 6 of Fig. 2 are explained in Section 7.2 and Section 8, respectively.

6.2 Feature Selection and Combination in our Theory-Driven Approach

The features (that is, the goal-blocking factors) in our linear regression model for the frustration index F_i, are not the features directly available in the log file. Instead, they are created by selecting and appropriately combining the available features from the Mindspark log file (Table 2) using theory. We illustrate the advantages of selecting features by applying a goal-blocking based theory, by comparing a feature with a simple combination of data from log file. Both methods start with the same raw data - features from the log file which indicate the students' response to the current question (a_i), the previous question (a_{i-1}), and two questions before the current question (a_{i-2}).

In our approach, we use a complex combination of the previous three responses ($goal2.bf$ in this example). This combination, $goal2_bf$, is calculated using equation (4). If $a_i = 1$, this indicates that the current question is correct and $a_i = 0$ indicates that the current question is wrong. When the goal of achieving a Sparkie (goal2), is blocked near the goal (the student answers first two questions correctly, in a sequence of three questions), then the blocking factor $goal2.bf$, which contributes to the frustration index is high. Similarly, when the goal is blocked midway to the goal (the student answers only the first of a sequence of three questions correctly) then the value of $goal2.bf$ is medium. Finally, when the goal is blocked far from the goal (the student answers the first of a sequence of three questions wrongly), $goal2.bf$ has a low value. The difference between goal-blocking near the goal and far from the goal is captured in our approach. This is not possible in a simple combination of the data from log file (the last column in Table 2). Since the causes of frustration are identified in finer detail, it may help us in addressing frustration in finer detail and also may help us to prevent frustration instances.

7 Experimental Methodology

In this section, we discuss the independent method we used to identify frustration: human observation. We conducted human observations of students interacting with Mindspark to validate our frustration model and to determine the weights of our model. We describe the observation details, labeling of the observations and the metrics used to compare the results of our frustration model with those from the human observation method.

TABLE 2
An example to illustrate the advantages between selecting features by applying goal-blocking based theory and by simple combination of the data from log file when applied to Mindspark

			Theory-Driven Approach			**Simple Combination of Data from Log File**
a_{i-2}	a_{i-1}	a_i	Eq 3a	Eq 3b	$goall2.bf$ Eq 4	Sum of last 3 responses
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	1	1	1
0	1	1	0	0	0	2
1	0	0	0	0	0	1
1	0	1	0	0	0	2
1	1	0	1	1	2	2
1	1	1	0	0	0	3

7.1 Human Observation

The goal of human observation was to observe the students' facial expressions and label them as frustrated or non-frustrated. In the human observation method, the observer observes a student's facial expression while s/he interacts with the ITS. In this section, we discuss the expressions observed and affective states identified from facial expressions.

7.1.1 Sample

We recorded video of facial expressions of 27 students (13 female, 14 male). 6 of the students were from one school in Mumbai, India and 21 students from another school from Ahmedabad, India (both urban areas in India). The students are fifth or sixth standard students with the age range of 10-12. We recorded the video of students' expressions while they interacted with a Mindspark session in their school. Each video was 25-30 minutes long, and on an average, contained a student's facial expressions over 30 to 40 questions.

7.1.2 Video Recording Procedure

We contacted two schools which were already using Mindspark in their curriculum. We explained our objectives to students and their parents and requested their consent to record the students' facial expressions. The recording of the facial expressions and storing of videos adhered to ethics committee guidelines. The students' facial expressions were recorded using a webcam, and the students' interaction with the computer was recorded using Camstudio[3]. The collected videos were used for human observation. The students' facial expressions were coded, after they received feedback to the response they submitted for each question. Our goal was to capture the students' expressions at the point where

3. www.camstudio.org

they learn whether their answer to Mindspark question was correct or wrong.

7.1.3 Instrument

We used an observation protocol based on a facial analysis coding system [32], [33]. The data collection sheet contained the following information: Student ID (Mindspark login ID), Question Number and observation made by the observer. A sample observation that records a student's expressions is given in Table 3.

7.1.4 Observation Procedure

The observers were Ph.D. students in Educational Technology at the Indian Institute of Technology (IIT), Bombay. All observers had taken the 'Research Methodology in Education' course, and hence had an understanding of the observation data collection method. All the observers had practiced facial expression coding [32], [33], during a pilot observation, prior to the actual study. After the pilot study, the observations were checked for inter-observer reliability. Observers agreed 80% of the time with the facial expression coding of other observers and Cohen's κ was found to be 0.74, a substantial agreement in the pilot study. In our actual study, the observers observed the students' facial expression from the video whenever the student submitted an answer to a question. The recorded videos helped the observers to pause the video and note down the expression. The observations were recorded on the data collection sheet. In our study, only one observer out of four is involved in building the model. Moreover, the students' interaction with the computer is used only to identify when the facial expressions are to be observed, and are not shown to the observer during the observation process. Hence, the observation process is truly an independent method of identifying frustration.

7.1.5 Labeling Technique

The observers classified the students' facial expressions into frustration and non-frustration, based on the guidelines given in [15] and [8]. The key behaviors to express frustration are:

- Outer brow raise
- Inner brow raise
- Pulling hair
- Statements like "what", "this is annoying", "arey (Hindi word for expressing disappointment)"
- Banging on the keyboard or the mouse
- Cursing

We show labeling of the students' affective states along with sample observation data in Table 3.

Facial emotions have been reported by Ekman to be universal [34]. The study by Hillary et. al. [35] reviews 87 articles, related to cross-cultural facial expression, and reports that "Emotions may be more accurately understood when they are judged by members of same national, ethnic or regional group that had expressed the emotion". The study by M.K. Mandal et. al. [36], of India, analyses the observer-expressor culture differences. The study includes 43 university students from Canada and 43 university students from India. The results support the universality of facial emotions. Based on the research results reported in the above articles we conclude that, (i) facial expressions are universal for frustration, and (ii) emotion recognition from facial expression is done better if the observer-expressor are from the same culture/nation (which is the case in our study). Hence, our facial expression observation is a valid independent analysis to validate our model.

TABLE 3
Sample Human Observation Sheet to Record Students' Facial Observations and to Label it as Frustration (Frus) or Non-Frustration (Non-Frus)

Question Number	Observation	Classification
1	Mouth open, lower lip down	Non-Frus
2	Reading aloud	Non-Frus
3	Making noise (cha, arey), resting forehead with fingers	Frus
4	Lips tightening	Non-Frus
5	Two hands raised up to to chest	Non-Frus
6	Hands clamped	Non-Frus
7	Reading	Non-Frus
8	Head lean towards screen immediately	Non-Frus
Challenge Question	Inner brow up, eyes wide open, noise (hindi cursing words)	Frus
9	No expression	Non-Frus
10	Mouth little open, eyes wide	Non-Frus
11	Inner eye brows lowered (shrink)	Non-Frus

7.2 Analysis Procedure

We recorded 932 observations from 27 students. Among those, 137 observations were classified as frustration (Frus) and remaining as non-frustration (Non-Frus). The unbalance in the data can led to classification bias. To avoid the classifier bias towards the majority class (Non-Frus), we can up-sample the non majority class (Frus) data or down-sample the majority class (Non-Frus) data. However, in real-world scenario, number of non-frustration instances are more compared to frustration instances. Hence, we use the data obtained from Mindspark log file, as it is, without up-sampling or down-sampling to replicate the real-world scenario.

The dataset is stratified at student level. We represent the values obtained from human observation as B_i at the i^{th} instance, $B_i = 0$ for non-frustration and $B_i = 1$ for frustration.

To validate our model, we need to calculate the Frustration Index, F_i (using Equation 8), for the students whose facial expressions were observed. In order to calculate F_i, the weights of the frustration model need to be learned. The procedure to learn the weights of our frustration model, corresponding to **Step 5** in Fig. 2, is given below:

1) To maintain a uniform scale among all the features in the frustration model, we apply normalization to all the features. We used the following normalization equation.

$$X_{new} = \frac{X - Mean(X)}{Max(X) - Min(X)}$$

Here, X_{new} is the normalized value of feature X. We used the range $(Max(X) - Min(X))$ of the feature in the denominator, instead of standard deviation. This is because the feature "response time" had a wide range of data (from 1 second to 200 seconds) compared to other features. Hence standard deviation in the denominator will not normalize all the features in an uniform scale.
2) We used cross-validation technique [37] to validate our frustration model and to check how generalizable our frustration model is, when applied to independent data. In this paper, we used the tenfold cross-validation technique [37] for all the experiments.
3) We use linear regression analysis to identify the values for weight by assigning, 0 and 1 to represent non-Frus and Frus respectively in the dataset.
4) We apply threshold to F_i to classify the frustration index values as frustration and non-frustration, we call this value as predicted value P_i.
5) We use the trained model (weights learned from linear regression analysis), to predict the frustration on test dataset and compared our prediction with human observation.

7.3 Metrics to Validate Frustration Model

The metrics used to measure the performance and to validate our frustration model are discussed in this subsection. Based on the feature from the log file, we predict the students' affective state at a given instant, as frustration or non-frustration, hence we consider it as a binary classification problem. In binary classification, most of the evaluation metrics are based on a contingency table [38], [39]. The 2x2 contingency table for our research is shown in Table 4:

TABLE 4
Contingency Table

	Actual Frus	Actual Non-Frus
Pred Frus	True Positive (TP)	False Positive (FP)
Pred Non-Frus	False Negative (FN)	True Negative (TN)

where, Actual Frus and Actual Non-Frus are results from human observation based on the students' facial expressions while they interact with the ITS. Pred Frus and Pred Non-Frus are predicted values of frustration and non-frustration respectively, using our frustration model on the data from the Mindspark log file.

The most common metric used in a classification problem based on the contingency table is accuracy [40]. Accuracy measures the ratio of correctly predicted instances (TP + TN) to total instances.

Other standard measures in classification problems are precision and recall based on the contingency table. Precision measures the ratio of correctly predicted frustration instances (TP) to total number of predicted frustration instances (TP + FP). Recall measures the ratio of correctly predicted frustration instances (TP) to actual number frustration instances identified from human observation (TP + FN). Since our dataset is an unbalanced distribution of frustration and non-frustration, we calculate the F1 score and Cohen's kappa to measure the performance of our model compared to random guess.

As we discussed already, it is not feasible to predict all types of frustration experienced by a student from all sources, especially those extrinsic in nature. In our research, we are interested in predicting the students' frustration arising from their interactions with the ITS. Hence, we report precision and recall of predicting frustration (instead of predicting non-frustration). Our goal is to ensure the correctness of our prediction of frustration, instead of being able to predict all frustration instances encountered by students while interacting with the ITS. Hence, high precision, and not high recall, is the important metric in our research.

8 Performance of Our Approach on Data from Mindspark log File

The threshold value to determine frustration is chosen by averaging the values we used to represent frustration and non-frustration in the training dataset. The threshold value used is 0.5 (average of 0 and 1). Since we use the linear regression classifier, we consider the mid-value as the threshold. In linear regression analysis, our goal is to minimize the difference in predicted frustration P_i and corresponding human observation values B_i, $(P_i - B_i)$

$$min(P_i - B_i)^2$$

$$\text{by varying } w_0, w_1, w_2, w_3, w_4, w_5$$

To solve the above problem, we used GNU Octave[4], an open source numerical computation software.

In our experiments, to train the weights to optimum value, we used the gradient descent algorithm with step size = 0.001. Our approach leads to a converged set of weights after 70000 iterations as seen in Fig. 3. We manually vary α, which represents the proportion of frustration from previous questions, from 0.1 to 1 in steps of 0.1, and the alpha value which gave best accuracy on training data was selected. The selected $\alpha = 0.8$.

4. http://www.gnu.org/software/octave/

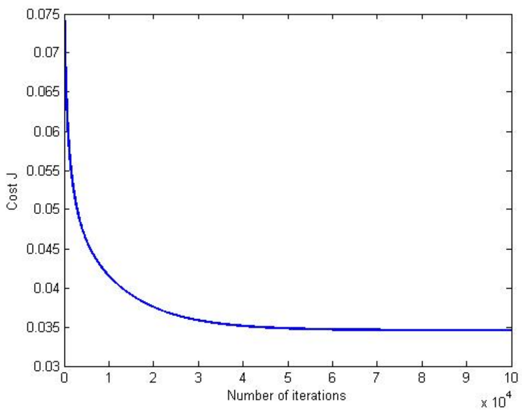


Fig. 3. Convergence of Weights of our Linear Regression Frustration Model using Gradient Descent Algorithm

The performance of our frustration model on Mindspark data, using tenfold cross-validation, compared to human observation is given in Table 5.

TABLE 5
Contingency Table of Our Approach when Applied to Mindspark Log Data

		Human Observation	
		Frustrated	Non-Frustrated
Pred	Frustrated	45	12
Result	Non-Frustrated	92	783

The values from Table 5 are used to calculate the metrics mentioned in the Section 7.3, the results are given in Table 6.

TABLE 6
Performance of our Approach Shown Using Various Metrics when Applied to Mindspark Log Data

Metrics	Results
Accuracy	88.84%
Precision	78.94%
Recall	32.85%
Cohen's kappa	0.41
F1 Score	0.46

From the above results, the accuracy and precision of our frustration model are high compared to recall. As we discussed already, we are interested in how correct our prediction of frustration is (measured by Precision) instead of predicting all frustration instances encountered by students (reflected by Recall). Thus, the results are aligned with our research goals. The Cohens kappa (0.41) and F1 measure (0.46) are found and it is an acceptable value. We have also calculated ROC values to measure performance of our model, and the result (0.01, 0.32) lies far above from the middle line (random guess) which indicates that the selected threshold is valid. And the selected threshold gives a better balance of precision and recall compared to other threshold values.

9 Performance of Different Approaches Applied to the Data from Mindspark Log File

While our theory-driven approach gave high accuracy and precision for predicting frustration, we would like to compare how the theory-driven approach compares with data-mining approaches. In this section, we compare the results of our theory-driven approach to predict frustration, along with some existing data-mining approaches, applied to the data from Mindspark log file. We consider the approaches used in AutoTutor [12], Crystal Island [22], and approach in [13], and apply them to the same Mindspark data.

We identified the students' frustration using the data from the Mindspark log file by applying the approaches in [12], [22] and [13]. We identified 14 features from the Mindspark log file related to the students' responses, time spent to answer and time spent on reading the explanation. We captured these features, after students answer each question in the session. As per the approaches in AutoTutor [12] and programming lab [13], we did a correlation analysis of these 14 features with the observed affective state to select those features that correlated with observed frustration. We identified 10 such features that were correlated with observed frustration, and omitted the remaining uncorrelated features. To avoid redundancy and to reduce the number of features, we did correlation analysis among features and removed the strongly correlated features (Pearson's $r > 0.7$) as suggested in [12]. If two features are highly correlated then the feature which has a higher correlation with the affective state is preserved [12].

After this analysis, seven features were selected: i) response to the question (whether the response provided by a student is correct or not), ii) response time to answer the question, iii) time spent on explanation of the answer, iv) response to the Challenge Question, v) sum of responses to the previous two questions, vi) sum of responses to the previous four questions and vii) average response time to answer the previous three questions. We used these seven features to follow a similar approach done in [12] and [13]. We applied the data to all the classifiers used in AutoTutor [12], to identify frustration from the log file. We report only the results of best classifiers in each category in Table 7.

Similarly, we used all 14 features to follow the similar approach done in Crystal Island [22], and we did experiments on Naive Bayes, SVM and Decision Tree classifiers as mentioned in [22]. The results along with theory-driven approach are shown in Table 7. We used the tenfold cross-validation method in all our analyses.

TABLE 7
Comparison of our Approach with Existing Data-Mining Approaches Applied to the Data from Mindspark Log File

System	Classifiers	Accuracy in %	Precision in	Recall in %
AutoTutor approach [12] (Selected Features)	Naive Bayes	82.83	40.94	37.95
	MLP	86.59	55.76	42.33
	K*	87.02	56.89	48.17
	Bagging Pred	87.55	57.89	56.20
	Logistic Model Tree	**88.63**	**65.97**	**46.71**
	PART	87.23	60.97	36.49
Crystal Island approach [22] (All Features)	Naive Bayes	81.12	38.72	48.90
	Decision Tree	**86.05**	**52.63**	**51.09**
Introductory programming lab approach [13]	Selected features used to form a linear regression model	r = 0.583		
Our Approach of this paper	**Linear Regression Model**	**88.84**	**78.94**	**32.85**

Bold – Best results obtained in each approach

10 Discussion

In Table 7, the best results by each approach are highlighted. Within the AutoTutor [12] approach, the Logistic Model tree performs comparatively better than the other classifiers on the Mindspark dataset with an accuracy of 88.63%, a precision of 65.97%, and a recall of 46.71%. In Crystal Island [22] approach, we observed that Decision Tree classifier gives maximum accuracy of 86.05%, precision of 52.63%, and a recall of 51.09% as compared to other classifiers.

In our research, what is important is how correctly the frustration instances are predicted, rather than predicting all frustration instances encountered by students. Our theory-driven approach performed comparatively better than other approaches in precision of 78.94% (best result from data-mining approach is a precision of 65.97%) and comparatively equal in an accuracy of 88.84% (best result from data-mining approach is accuracy of 88.63%). Hence, our goal of achieving best precision is achieved in our theory-driven approach. However, our theory-driven approach performed poorly in recall of 32.85% (best result in data-mining approach is recall of 56.2%). The reason for better precision and poor recall could be that the features are selected based only on goal-blocking type of frustration and hence other types of frustration might have been missed. A significant advantage of our theory-driven approach is that the features identified give a clear interpretation of the reasons for the students' frustration and can be useful for informed adaptation. This knowledge can give information on which variables to control while doing an adaptation to mitigate frustration.

10.1 Performance of other Classifier Models

To test if different models for frustration (other than a linear model) perform better, we applied our features to second order, third order polynomial models and logistic regression classifier. We also applied our data to non-linear classifiers in Weka [21] The results of the other classifiers are comparatively equal to the linear regression classifier. Hence, for the ease of understanding of the causes of frustration, we use a linear regression model instead of higher order models.

11 Conclusion

We proposed a theory-driven approach to predict frustration of a student working with an ITS to understand the causes of frustration. The results of our approach are relatively equal compared to existing approaches. Our approach performs better in precision compared to other approaches of selecting features from the ITS log data. By using our frustration model, the cause of frustration is clearly seen, since we can infer which feature contributes more towards frustration. Since the cause of frustration is clear, it can lead to an informed adaptation in addressing frustration. Hence, we recommend our approach to those ITS developers, who are interested in not only detecting frustration but also in identifying its cause, thereby, enabling them to perform an informed adaptation.

In order to apply our theory-driven approach to other systems, careful thought is required to operationalize the blocking factors of goals. The goals of the students when they interact with the system should be captured, this is a limitation in the scalability of our approach. The results of the theory-driven approach are dependent on how well the goals are captured and how well the blocking factors of the goals are operationalized.

In our future work, we propose to address the frustration (predicted by our model), in real-time, in Mindspark's production environment. We are in the process of integrating our model with the Mindspark code to: (i) check the student's frustration index after every response, and (ii) trigger an appropriate adaptation whenever the index is above a threshold value (0.5). We also propose to perform experiments to determine the appropriate setting for the threshold value for different age groups of students. We are in the process of developing theory-driven adaptation strategies to mitigate frustration in a timely manner, during a student's interaction with the ITS.

ACKNOWLEDGMENTS

The authors would like to thank all the students who agreed to participate in our study. We would also like to thank the staff and the Managing Director of Education Initiatives for their support while we conducted our research.

REFERENCES

[1] P. Brusilovsky and E. Milln. User models for adaptive hypermedia and adaptive educational systems. In *The adaptive web: methods and strategies of web personalization*, pages 3–53. Springer-Verlag, 2007.

[2] B. Woolf, W. Burleson, I. Arroyo, T. Dragon, D. Cooper, and R. Picard. Affect-aware tutors: Recognising and responding to student affect. *International Journal of Learning Technology*, 4(3/4):129–164, 2009.

[3] S. K. D'Mello, S. D. Craig, J. Sullins, and A. C. Graesser. Predicting affective states expressed through an emote-aloud procedure from autotutor's mixed-initiative dialogue. *International Journal of Artificial Intelligence in Education*, 16(1):3–28, 2006.

[4] J. Klein, Y. Moon, and R. Picard. This computer responds to user frustration: Theory, design, and results. *Interacting with Computers*, 14(2):119–140, 2002.

[5] R. S. J. d. Baker, S. K. D'Mello, M. M. T. Rodrigo, and A. C. Graesser. Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive-affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies*, 68(4):223 – 241, 2010.

[6] R. A. Calvo and S. K. D'Mello. Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing*, 1(1):18–37, 2010.

[7] Z. Zeng, M. Pantic, G.I. Roisman, and T.S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):39–58, 2009.

[8] M. M. T. Rodrigo, R. S. J. d. Baker, M. C. V. Lagud, S. A. L. Lim, A. F. Macapanpan, S. A. M. S. Pascua, J. Q. Santillano, L. R. S. Sevilla, J. O. Sugay, S. Tep, and N. J. B. Viehland. Affect and usage choices in simulation problem-solving environments. In *International Conference on Artificial Intelligence in Education*, pages 145–152, 2007.

[9] M. M. T. Rodrigo, R. S. J. d. Baker, J. Agapito, J. Nabos, M. C. Repalam, S. S. Reyes, and M. O. C. Z. S. Pedro. The effects of an interactive software agent on student affective dynamics while using an intelligent tutoring system. *IEEE Transactions on Affective Computing*, 3(2):224 –236, 2012.

[10] D. Cooper, I. Arroyo, B. Woolf, K. Muldner, W. Burleson, and R. Christopherson. Sensors model student self concept in the classroom. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 30–41. 2009.

[11] C. Conati and H. Maclaren. Empirically building and evaluating a probabilistic model of user affect. *User Modeling and User-Adapted Interaction*, 19(3):267–303, 2009.

[12] S. K. D'Mello, S. D. Craig, A. Witherspoon, B. Mcdaniel, and A. C. Graesser. Automatic detection of learner's affect from conversational cues. *User Modeling and User-Adapted Interaction*, 18(1-2):45–80, 2008.

[13] M. M. T. Rodrigo and R. S. J. d. Baker. Coarse-grained detection of student frustration in an introductory programming course. In *International Workshop on Computing Education Research*, pages 75–80, 2009.

[14] Jennifer Sabourin, Bradford Mott, and James C. Lester. Modeling learner affect with theoretically grounded dynamic bayesian networks. In *International Conference on Affective Computing and Intelligent Interaction*, pages 286–295, 2011.

[15] S. K. D'Mello, S. D. Craig, B. Gholson, and S. Franklin. Integrating affect sensors in an intelligent tutoring system. In *International Conference on Intelligent User Interfaces*, pages 7–13, 2005.

[16] T. Dragon, I. Arroyo, B. Woolf, W. Burleson, R. Kaliouby, and H. Eydgahi. Viewing student affect and learning through classroom observation and physical sensors. In *Intelligent Tutoring Systems*, pages 29–39, 2008.

[17] K. Brawner and B. Goldberg. Real-time monitoring of ecg and gsr signals during computer-based training. In *Intelligent Tutoring Systems*, pages 72–77, 2012.

[18] M. Hussain, H. Monkaresi, and R. Calvo. Categorical vs. dimensional representations in multimodal affect detection during learning. In *International Conference on Intelligent Tutoring Systems*, pages 78–83, 2012.

[19] S. Srinivas, M. Bagadia, and A. Gupta. Mining information from tutor data to improve pedagogical content knowledge. In *Educational Data Mining*, pages 275–276, 2010.

[20] M. S. Hussain, O. AlZoubi, R. A. Calvo, and S. K. D'Mello. Affect detection from multichannel physiology during learning sessions with autotutor. In *International Conference on Artificial Intelligence in Education*, pages 131–138, 2011.

[21] M. Hall., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11(1):10–18, 2009.

[22] S. W. Mcquiggan, S. Lee, and J. C. Lester. Early prediction of student frustration. In *International Conference on Affective Computing and Intelligent Interaction*, pages 698–709, 2007.

[23] A. Ortony, G. L. Clore, and A. Collins. *The cognitive structure of emotions*. Cambridge university press, 1990.

[24] R. Lawson. *Frustration: the development of a scientific concept*. Critical issues in psychology series. Macmillan, 1965.

[25] C. T. Morgan, R. A. King, J. R. Weisz, and J. Schopler. *Introduction to Psychology*. McGraw-Hill Book Company, seventh edition, 1986.

[26] A. Amsel. Arousal, suppression, and persistence: Frustration theory, attention, and its disorders. *Cognition & Emotion*, 4(3):239–268, 1990.

[27] N. L. Stein and L. J. Levine. Making sense out of emotion. In W. Kessen, A. Ortony, and F. Kraik, editors, *Memories, thoughts, and emotions: Essays in honor of George Mandle*, pages 295–322. 1991.

[28] James Paul Gee. What video games have to teach us about learning and literacy. *Computers in Entertainment (CIE)*, 1(1):20–23, 2003.

[29] James Paul Gee. Learning and games. *The ecology of games: Connecting youth, games, and learning*, 3:21–40, 2008.

[30] Ashish Kapoor, Winslow Burleson, and Rosalind W. Picard. Automatic prediction of frustration. *Int. J. Hum.-Comput. Stud.*, 65:724–736, 2007.

[31] J. Lazar, A. Jones, M. Hackley, and B. Shneiderman. Severity and impact of computer user frustration: A comparison of student and workplace users. *Interacting with Computers*, 18(2):187–207, 2006.

[32] T. Kanade, Y. Tian, and J. F. Cohn. Comprehensive database for facial expression analysis. In *International Conference on Automatic Face and Gesture Recognition*, pages 46–53, 2000.

[33] Y. Tong, W. Liao, and Q. Ji. Inferring facial action units with causal relations. In *Computer Vision and Pattern Recognition*, pages 1623–1630, 2006.

[34] P. Ekman. Strong evidence for universals in facial expressions: a reply to Russell's mistaken critique. *Psychology Bulletin*, 115(2):268–287, 1994.

[35] H. A. Elfenbein and N. Ambady. On the universality and cultural specificity of emotion recognition: A meta-analysis. *Psychology Bulletin*, 128(2):203–235, 2002.

[36] M. K. Mandal, M. P. Bryden, and M. B. Bulman-Fleming. Similarities and variations in facial expressions of emotions: Cross-cultural evidence. *International Journal of Psychology*, 31(1):49–58, 1996.

[37] S. Geisser. *Predictive Inference: An Introduction*. Monographs on Statistics and Applied Probability. Chapman & Hall, 1993.

[38] G. Salton. *Automatic Text Processing; the Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989.

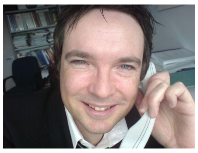
[39] Y. Yang. An evaluation of statistical approaches to text categorization. *Information retrieval*, 1(1):69–90, 1999.

[40] D. Michie, D.J. Spiegelhalter, C.C. Taylor, and J. Campbell. *Machine learning, neural and statistical classification*. Ellis Horwood London, 1994.

Ramkumar Rajendran received the BE degree from University of Madras, and the ME from Anna University. He is currently pursuing the PhD degree in the Dept of Computer Science & Engineering from IITB-Monash Research Academy, a joint research program between Indian Institute of Technology, Bombay and Monash University, Melbourne. His research interest include learning analytics, intelligent tutoring system, affective state recognition and adaptive learning.

Sridhar Iyer is presently an Associate Professor in the Dept of Computer Science & Engineering at Indian Institute of Technology (IIT) Bombay. Prior to this, s/he has been a faculty member in the Dept of Computer Science & Engineering at IIT Guwahati. His research interests include: eLearning technologies, networking protocols and multimedia tools for distance education, wireless networking and applications, and some areas in program/protocol verification. Sridhar Iyer received his BTech, MTech and PhD from the Dept of Computer Science & Engineering at IIT Bombay.

Sahana Murthy is an Assistant Professor in the Educational Technology Inter-Disciplinary Programme at the Indian Institute of Technology (IIT) Bombay, India. She has worked in the area of physics education research and the integration of ICT in education. Her current research interests lie in the role of computer-based visualizations in teaching and learning, and the development of scientific abilities among college students. Sahana Murthy received her Ph.D. in physics from Rutgers University, USA, M.Sc. in physics from IIT Bombay and B.Sc. in physics from the University of Mumbai, India.



Dr Wilson is a Senior Lecturer at the Caulfield School of Information Technology, Faculty of Information Technology at Monash University. His interests include databases, information retrieval and XML data management. He is also a member of the Centre for Distributed Systems and Software Engineering (DSSE) in the Faculty of Information Technology at Monash University.

Judithe Sheard is an Associate Professor in the Faculty of Information Technology and is the Head of the Caulfield School of Information Technology, Monash University. She has leadership roles both nationally and internationally in the computing education research community. Associate Professor Sheard is co-director of the Computing Education Research Group at Monash and immediate past chair of the Australasian chapter of the ACM special interest group in computing education (SIGCSE). She has published more than 100 peer-reviewed publications on IT education and educational technology. She has over 20 years teaching experience in IT. Together with three other colleagues s/he was awarded the 2002 Monash Vice Chancellor's Award for Team-based Educational Development. Associate Professor Sheard established and is director of the Monash Museum of Computing History.