Participation-based Student Final Performance Prediction Model through Interpretable Genetic Programming: Integrating Learning Analytics, Educational Data Mining and Theory

ABSTRACT

Building a student performance prediction model that is both practical and understandable for users is a challenging task fraught with confounding factors to collect and measure. Traditionally, most prediction models are difficult for teachers without a significant background in probability to interpret. This poses significant problems for model use (e.g. personalizing education and intervention) as well as model evaluation. In this paper, we synthesize learning analytics approaches, educational data mining (EDM) and HCI theory to explore the development of more usable prediction models and prediction model representations using data from a collaborative geometry problem solving environment: Virtual Math Teams with Geogebra (VMTwG). First, based on theory proposed by Hrastinski (2009) establishing online learning as online participation, we operationalized activity theory to holistically quantify students' participation in the CSCL (Computer-supported Collaborative Learning) course. As a result, 6 variables, *Subject, Rules, Tools, Division of Labor, Community,* and *Object*, are constructed. Unlike some traditional blunt instruments (feature selection, Ad-hoc guesswork etc.), this step diminishes data dimensionality and systematically contextualizes data in a semantic background. Secondly, an advanced modeling technique, Genetic Programming (GP), is coded to develop the prediction model. We demonstrate how connecting the structure of VMTwG trace data to a theoretical framework and processing that data using the GP algorithmic approach outperforms traditional models in prediction rate and interpretability. Theoretical and practical implications are then discussed.

Keywords

Learning Analytics; Educational Data Mining; Prediction; CSCL; Activity Theory; Genetic Programming

1. Introduction

The ability to predict a student's final performance has gained increased emphasis in education (Baker & Yacef, 2009; Romero & Ventura, 2010). One of the practical applications of student performance prediction is for instructors to monitor students' progress and identify at-risk students in order to provide timely interventions (Bienkonwski et al., 2012). It is already difficult to detect at-risk students in a regular classroom, not to mention when classes are much larger and learning happens online, e.g. MOOCs (Gunnarsson & Alterman, 2012). It would be desirable to expand beyond the at-risk students to predict the future performance of all students, allowing a feedback process to enhance learning and awareness for a greater number of students during the course (Zafra & Ventura, 2009). As an automated method, student performance prediction has the potential to decrease teachers' duty in assessment.

The objective of performance prediction is to estimate an unknown value – the final performance of the student. In order to accomplish this goal, a training set of previously labeled data instances is used to guide the learning process (Espejo et al, 2010), while another set of correctly labeled instances, named the "test set", is employed to measure the quality of the prediction model obtained (Márquez-Vera, Cano & Romero, 2013). Previous studies that have documented student performance prediction models have focused on statistical modeling and data mining techniques (Wolff et al., 2013; Gunnarsson & Alterman, 2012; Thomas & Galambos, 2004). These traditional modeling methods have their own limitations. From the perspective of Educational Data Mining (EDM), which focuses on model and algorithm development to improve predictions of learning outcomes (Siemens & Baker, 2012), existing statistical and data mining methods usually do not have an established paradigm to optimize performance prediction. For example, statistical models such as linear regression or logistic regression have requirements related to the distribution of data and a-priori regression function structures. Poor estimation and inaccurate inferences would be generated if the basic premises of the regression models are breached (Harrell, 2001); and it is difficult for end users to detect when such breaches occur. In addition, there is a strong tradition in the domain of education of employing linear or quadratic models, limiting exploration of potentially more useful models for predicting student performance.

Approaching educational performance prediction through learning analytics is to generate actionable intelligence for teachers and students to improve learning, which deals with the interpretation and contextualization of data (Agudo-Peregrina et al., 2014). Model interpretability in performance prediction is important for two primary reasons (Henery, 1994): first, the constructed model is usually assumed to support decisions made by human users- in our context, to facilitate teachers to provide individualized suggestions to students. If the discovered model is a black-box, which renders predictions without explanation or justification, people or teachers may not have confidence in it. Second, if the model is not understandable, users may not be able to validate it, hindering the interactive aspect of knowledge validation and refinement. Unfortunately, traditional prediction models (e.g. support vector machines, neural networks etc.) require a sophisticated understanding of computation that most teachers do not possess (Romero & Ventura, 2010; Siemens & Baker, 2012). Lacking readability for teachers results in difficulty for them to provide meaningful feedback to students. For instance, Campbell, Deblois & Oblinger (2007) employed logistic regression, neural networks and other models to search for students that are at-risk of failing and alert instructors to potential issues. While automatic alert messages enable teachers to quickly identify struggling students, the generation of a risk signal is unable to convey enough information to enable personalized interventions for students (Essa & Ayad, 2012). From an application perspective, typical data mining algorithms usually work as black boxes, and as a result, it is difficult to identify the relationship between student performance and the various factors affecting performance. In turn, these models are difficult to implement practically, demanding far more time and computing resources. Moreover, most previous studies stopped at the level of predicting failure and success of a student in a course or a program (e.g. Romero et al., 2013, Zafra & Ventura, 2009, Hamalainen & Vinni, 2006), while few went further to predict student performance at more granular levels. With focus put solely on low performing students, interventions have the risk of becoming a tool only for punitive means (Mintrop & Sunderman, 2009).

Moreover, previous research in forecasting students' performance has concentrated on methodology and the exploration of algorithms, overlooking the educational context, theories, and phenomena (Baker & Yacef 2009; Romero & Ventura, 2010). Many times, computational model results are at least difficult, if not impossible, for teachers to use and explain (Ferguson, 2012). To gain a deeper understanding of the factors influencing students' learning and to build an interpretable student performance prediction model, researchers must contextualize those data factors in the construct of educational theories and semantics. However, this is a difficult problem to resolve due to the number of factors (variables) affecting students' performance. A large set of selected variables can dramatically diminish both statistical and data mining prediction power (Vanneschi & Poli, 2012, Deegalla & Bostrom, 2006). Data dimensionality can be reduced using feature selection, but in educational situations in which human judgment is key (Siemens & Baker, 2012), it is more suitable to accomplish dimensionality reduction by constructing variables according to human theories (Fancsali, 2011). The automatic processing of data generated by these environments is a conceptually "blunt instrument" due to feature selection algorithms, statistical models and data mining grounded in mathematical theories rather than theories of human behavior. In practice, approaches to variable selection and construction are usually based on ad-hoc guesswork or significantly detailed experience in the educational field (Nasiri & Minaei, 2012; Tair & El-Haless, 2012; Cetintas et al, 2009). A principled, theory-based method for synthesizing factors from raw data will connect the input to computational prediction models more coherently than previous approaches.

Our Framework for Exploring more Understandable Prediction

This paper attempts to demonstrate how this integration of prediction model utility and understandability is a promising direction for research focused on automating analytics around humans working in computational systems. We selected the prediction model (Genetic Programming) that represents what we see in our results as the most optimal tradeoff between model understandability and the predication accuracy. To explore this aim, we synthesize prior work in learning analytics, EDM and activity theory to approach student performance prediction model construction. We draw on a theory proposed by Hrastinski (2009), which emphasizes participation in online learning as a central factor affecting performance. We then contextualize participation-related data factors on a semantic background using an operationalization of activity theory. Integrating activity theory directly into our decision-making about how to operationalize participation indicators allows for a systematic construction of variables, and reduces data dimensionality in a CSCL environment to only six aspects. Activity theory is not the only theory that could prove valuable; we start with because it takes people, tasks and technology into account.

We then use our activity theory derived participation indicators as inputs to a Genetic Programming (GP) model to develop our student performance prediction model. The GP model can build a prediction model without assuming any a-priori structure of functions. Additionally, GP has the potential to develop a more useful model because it relies on theoretically grounded factorization of data. Moreover, the proposed GP model is more easily understood by users when compared with traditional statistical and data mining algorithms, providing teachers actionable information to offer individualized suggestions to students in any performance state (at-risk, just survive, average or good etc.) as well as increasing students' awareness, provided that prediction results are also presented to them. As a final product, this model defines tangible relationships between student performance and its related variables. Therefore, in terms of practical application, the resulting prediction model may be easily implemented in a real life context.

This study provides a practical and interpretable student performance prediction model that enables teachers to discern differences in performance among students in a classroom full of small group geometry learners who are working in groups of three to five in a synchronous CSCL environment, Virtual Math Teams with Geogebra (VMTwG). The paper is organized as follows: Section 2 discusses related work and background information. Section 3 introduces the theoretical framework behind this study. Section 4 shows the context of the study and data format. Section 5 describes methodology. Section 6 presents experimental results and analysis. Section 7 discusses results. Section 8 summarizes this study, pointing out limitations and future research directions.

2. Literature Review

The development of student performance prediction models is one of the oldest and most popular practices in education (Romero & Ventura, 2010). There are many examples of the application of computational techniques to predict student performance. Several exemplary works using these techniques are described here to provide our research background.

Barber & Sharkey (2012) predicted student success in a course using a logistic regression technique that incorporated data generated from learning management, student information, and financial information systems. Roberge, Rojas & Baker (2012) relied on log data from a cognitive tutor software and qualitative coded observation data, performing linear regression to predict student learning outcomes. Similarly, Myller et al. (2002) employed linear regression to predict students' exam results (pass or fail), building variables out of 103 elements. Kotsiantis & Pintelas (2005) went a step further, combining several regression techniques such as linear regression, locally weighed linear regression, etc. and monitoring twenty variables in order to predict student grades in a distance learning program. Their studies on performance prediction are based more on relationships than on model development. The mathematical format of these models and the analytical expertise required to interpret them make retrieval of actionable intelligence difficult for many users. On the other hand, various data mining techniques have been applied to student prediction modeling. Calvo-Flores et al. (2006) predicted passing or failing grades in a course using neural network models based on log data generated from Moodle. A comparison of various data mining methods (e.g. support vector machines, k-nearest neighbors) has been performed to predict student success or failure in an intelligent tutor course (Hämäläinen, 2006), a Moodle hosted course (Romero et al., 2008), and web-based instructional systems (Ibrahim & Rusli, 2007).

From an application and learning analytics perspective, neural network and support vector machines models are "black-box" models, which are difficult to implement, and difficult for teachers to understand in order to provide individualized feedback to students. Several studies have explored "white-box" data mining models for prediction, which are shown to be more easily understood and interpreted by non-programmers or statisticians because these methods expose the reasoning process underlying the predictions (Romero et al., 2013; Freitas, Wieser & Apweiler, 2010). That is, "white box" methods provide explanations for classification results. Within the category of white box representation, there are a wide range of schemes proposed in the literature and numerous variations of those schemes. Some of these model presentation formats are evaluated as easier to interpret than others. For example, Bayesian Networks have been used to predict students' success using log data from an intelligent tutor system (Pardos et al., 2007) and to predict whether a question in an intelligent tutoring system will be answered correctly (Pardos et al., 2008).

Though they are labeled as white box models, Bayesian Networks – including the Naïve Bayes classifier, tend to be difficult to understand for the end users- mostly k-12 teachers and students. Unlike the "if-then" rule model presentation format, Bayesian models can be represented by a network structure where every attribute (measures or independent variables) directly depends on a class attribute (performance level in our situation). Bayesian Networks are able to represent understandable model/knowledge due to the network's graphical structure (Korb & Nicholson, 2003). Nevertheless, the interpretation of a Bayes classifier is still difficult, requiring users' familiarity with the concept of conditional probability and also the computational procedure (Freitas, Wieser & Apweiler, 2010). Considering that the main users of our environment, VMTwG, are k-12 teachers and students, Bayesian models would require substantial training before proving to be useful.

There is a reasonable agreement that representations such as "if-then" rules are more understandable than others (Freitas, Wieser & Apweiler, 2010). After conducting an extensive literature review in machine learning and data mining literature as well as empirical tests for users, Huysmans et al. (2011) also conclude that "if-then" rules are "without any doubt the most common and useful type for model representation." The number of rules and conditions within the rules also act as benchmarks to measure the understandability of the discovered models (Freitas, Wieser & Apweiler, 2010; Huysmans et al, 2011; Fernandez et al., 2010). Both tree-based and rule-based algorithms can generate models that are represented as "if-then" rule sets. In practice, Wolff et al. (2013) have used a decision-tree method to identify at-risk students in a distance-education program; Nebot et al. (2006) predicted students' success using fuzzy association rules in a web-based educational system. On the other hand, these studies produced mixed results with respect to prediction model performance, with prediction precision ranging from roughly 40% to 90%. Lack of data is a plausible explanation for these results. Educational data sets are usually quite small, often resulting in 50 to 100 instances of student data, but decision-tree based models and rule-based models typically require thousands of rows of data to properly train the algorithms (Hämäläinen & Vinni, 2006). Excluding MOOCs, most classes nowadays are in the scale of tens to several hundreds of students. By comparison, GP is found to be especially powerful in prediction performance in smaller datasets (Ni, Wang, Zheng, & Sivakumar, 2012; Afzal et al., 2010) due to its higher diversity both in terms of the functional form as well as the variables defining the models (Zhang & Bhattacharyya, 2004). The proposed GP model is expected to outperform these traditional "white-boxes" in prediction rate and understandability for teachers.

In fact, several studies have investigated the use of GP to develop student performance prediction models. Zafra & Ventura (2009) built a model to forecast whether a student would fail or pass a course in Moodle system. The variables selected measure performance, such as assignments finished, forums used, number of quizzes passed, and time spent on the assignment and quiz. Márquez-Vera et al. (2013) constructed a prediction model using the GP method to identify at-risk students in traditional school settings. In order to reduce the 77 variables collected, a feature selection technique was applied to reduce the data to 15 attributes. Without a semantic background to complement the resulting models, teachers have difficulty in providing concrete feedback to individual students. We operationalize activity theory to contextualize those variables and expect this semantic information behind the data can contribute to the comprehensibility of the model. Furthermore, the documented studies generally lacked a systematic method of selecting and constructing variables from a large number of factors and employed a mathematical method or ad-hoc guesswork instead. From the perspectives of EDM, learning analytics and application, it is difficult for these previous modeling techniques to satisfy all the desired traits: optimized prediction rate, model interpretation and contextualization as well as easy implementation. In this study, we investigate whether we can build a practical and understandable student performance prediction model for a CSCL environment by connecting learning analytics, EDM, and theory.

3. Theoretical Framework

3.1 Online learning as online participation

Research on technology-mediated learning is increasingly influenced by interaction and practice focused lenses pioneered by Vygotsky (1978) and Wenger (1998). Knowledge is a construct that is not only recognized in individual minds but also "in the discourse among individuals, the social relationships that bind them, the physical artifacts that they use and produce, and the theories, models and methods they use to produce them" (Jonassen & Land, 2000). Most recently, Hrastinski (2009) argued both empirically and theoretically that "online participation underlies online learning in a more powerful way than any other variable currently aware of." Hrastinski (2009) proposed that online learner participation (1) is a complex process of communicating with others, (2) is supported by physical and psychological tools, (3) is not necessarily synchronous, (4) is supported by all types of engaged activities. CSCL research, in particular, usually occurs in a setting that focuses on learning that results from the collaboration of three or more individuals. This small-group focused learning is often supported by virtual, physical and psychological objects, tools and methods. Therefore, in order to predict students' performance in the technologically mediated Virtual Math Teams with Geogebra (VMTwG) environment, systematically measuring student participation in the environment through a theoretically informed model can bridge computationally focused approaches like EDM and emergent approaches represented by learning analytics.

3.2 Activity theory

Activity theory is a social, psychological and multidisciplinary theory that seeks to be naturalistic, offering a holistic framework that describes activities in practice while linking together individual and social behavior (Barab, 2002; Leont'ev, 1974; Engeström, 1987; Nardi, 1996). A model of the structure of an activity system was formulated by Engeström (1999), and includes the interacting components of S*ubject, Object, Tools, Division of Labor, Community, Rules*, and *Outcome* (see Fig. 1).

Learning is "the joint activity of a student, physical/symbolic tool(s), and another person(s) performing together as a working social system to achieve some outcome under constraints such as rules" (Basharina, 2007). Activity theory focuses on how students' participation transforms objects and how components in a system mediate this transformation (Barab et al., 2002). Learning is reframed as social participation rather than as merely the product of practice (Barab et al., 2002; Engeström, 1999). In the CSCL context, the process of participation in this transformation is seen as learning. It is the sum of the system components and the tensions among them that compose this learning, thus influencing students' performance.

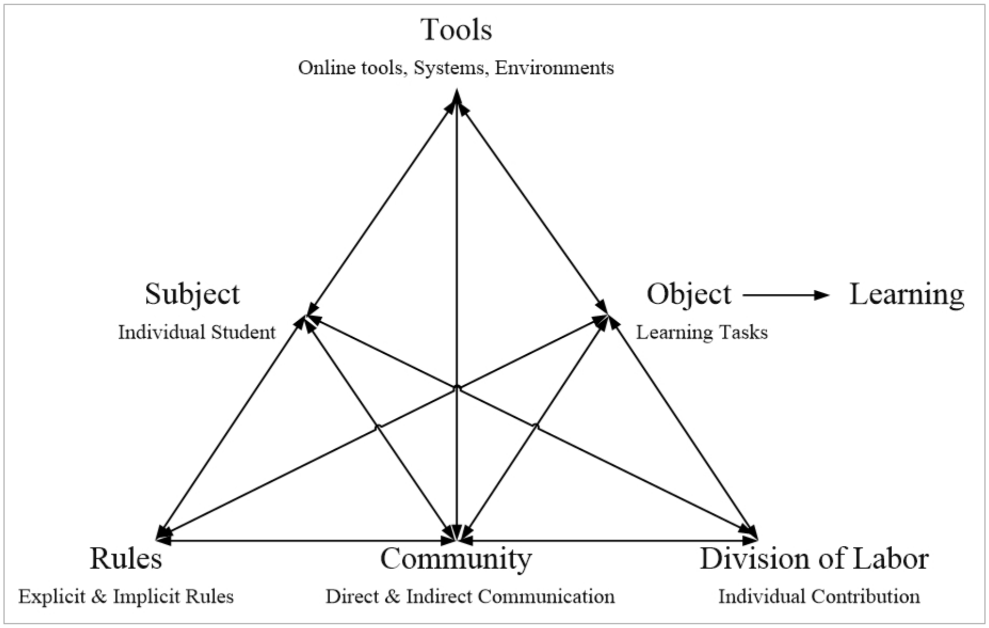


Fig. 1. Activity theory.

Activity theory enables us to address complex interactions and collaboration in the technology-mediated social environment, and to see into individual student participation in it (see Fig. 1). The activity theory system can be thought of as being built for each student, allowing us to highlight the learning of an individual student in collaborative group work in the CSCL setting (Table 1).

Table 1
Description of Activity Theory Operationalization in CSCL Context.

Measure-metric	Definition
Object	Completing learning tasks such as solving a problem or producing an artifact.
Subject	Activities involving individual students.
Tools	Computers, online tools, systems, and environments that mediate the learning activity
Community	Direct and indirect communication that enables an individual subject to maintain a sense of community with other students, teachers, and support staff
Rules	Implicit and explicit rules and guidelines that constrain the activity. For example, teachers can set specific rules for a learning task (explicit) and an individual student can only use the functions residing in the supporting tools (implicit)
Division of Labor	Concrete contributions each individual makes to the overall object

This paper contributes a more advanced, approach to understanding the activity theory-based constructs outlined in Table 1. Online learning is conceived as online participation, and activity theory provides a systematic way to frame participation and interaction. Activity theory not only enables us to holistically describe students' participation in CSCL learning, it also embeds the data in a semantic context which is the basis for building an understandable model and also for teachers to individualize interventions.

4. Research Context

4.1 VMT with Geogebra (VMTwG)

In this study, we operationalize activity theory in order to make sense of electronic trace data from a math discourse with 122 students which took place in 2013-14. Our analysis focused on four modules of a course designed to be taught with Virtual Math Teams with Geogebra (VMTwG) software (Fig. 2). The four modules that were analyzed included teams of three to five members. The four modules included: “Constructing Dynamic-Geometry Objects,” “Exploring Triangles,” “Creating Construction Tools,” and “Constructing Triangles,” The full curriculum currently includes a total of 21 topics, and is available on the project website (http://vmt.mathforum.org). Based on the learning outcomes of the students judged by human evaluators – whether the student completed the requirement in each module and its subtasks and how many tasks the student completes or understands the solutions generated by the group – cluster analysis is applied to these learning outcomes to generate granular categories for student performance. As a result, the performance distribution of 122 students is presented as: Excellent (10), Good (17), Average (39), Sufficient (38), At-Risk (18).

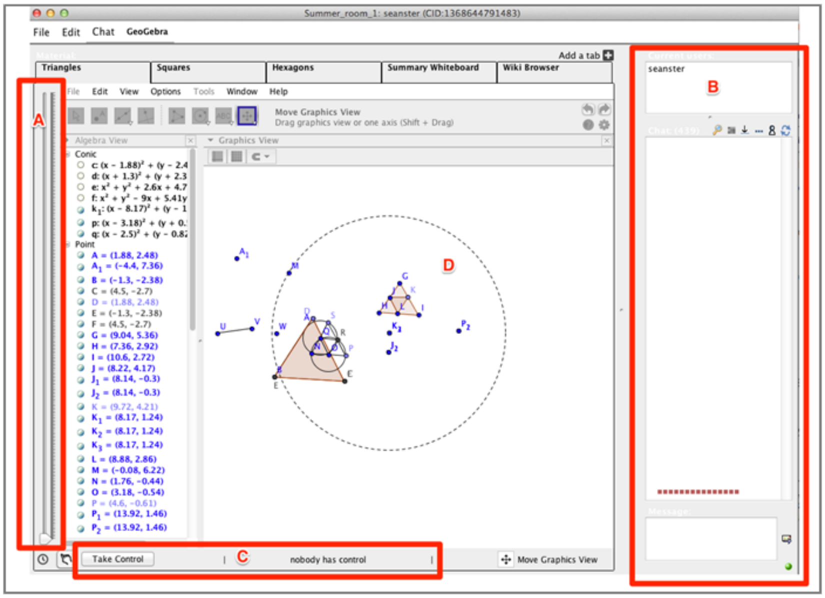
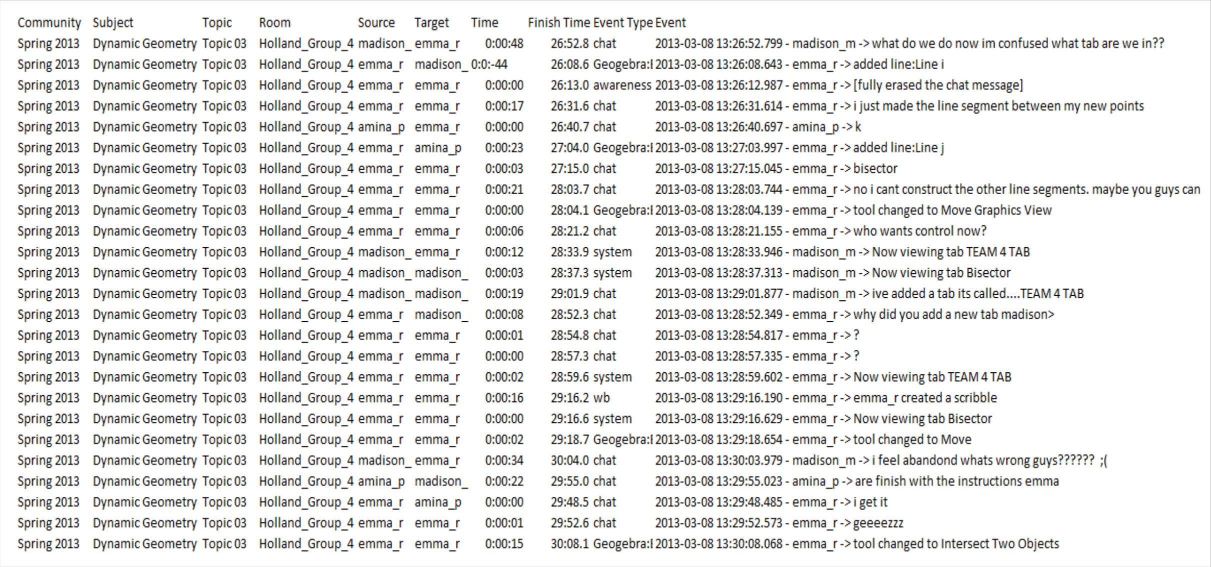


Fig. 2. VMTwG of an analytical tool for collaborative math discourse.

Fig. 2 provides a guide for understanding the cognitive learning discourse in VMT. There are four sections in Fig. 2. Section A, the VMT replayer bar, reveals the time dimension. Each action within VMTwG is logged with a timestamp. Section B is the chat window, where text is entered in chat. Sections C and D are related to Geogebra actions. C is the “Take Control” button, which gives an individual user control of the tools. Section D is the GeoGebra window itself. Here, students work to create an equilateral triangle within an equilateral triangle using multiple approaches. All the learning and assignments of this course took place in groups in the CSCL environment of VMTwG.

4.2 Dataset



Community	Subject	Topic	Room	Source	Target	Time	Finish Time	Event Type	Event
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	madison_	emma_r	0:00:48	26:52.8	chat	2013-03-08 13:26:52.799 - madison_m -> what do we do now im confused what tab are we in??
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	emma_r	madison_	0:0:-44	26:08.6	Geogebra:I	2013-03-08 13:26:08.643 - emma_r -> added line:Line i
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	emma_r	emma_r	0:00:00	26:13.0	awareness	2013-03-08 13:26:12.987 - emma_r -> [fully erased the chat message]
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	emma_r	emma_r	0:00:17	26:31.6	chat	2013-03-08 13:26:31.614 - emma_r -> i just made the line segment between my new points
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	amina_p	emma_r	0:00:00	26:40.7	chat	2013-03-08 13:26:40.697 - amina_p -> k
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	emma_r	amina_p	0:00:23	27:04.0	Geogebra:I	2013-03-08 13:27:03.997 - emma_r -> added line:Line j
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	emma_r	emma_r	0:00:03	27:15.0	chat	2013-03-08 13:27:15.045 - emma_r -> bisector
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	emma_r	emma_r	0:00:21	28:03.7	chat	2013-03-08 13:28:03.744 - emma_r -> no i cant construct the other line segments. maybe you guys can
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	emma_r	emma_r	0:00:00	28:04.1	Geogebra:I	2013-03-08 13:28:04.139 - emma_r -> tool changed to Move Graphics View
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	emma_r	emma_r	0:00:06	28:21.2	chat	2013-03-08 13:28:21.155 - emma_r -> who wants control now?
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	madison_	emma_r	0:00:12	28:33.9	system	2013-03-08 13:28:33.946 - madison_m -> Now viewing tab TEAM 4 TAB
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	madison_	madison_	0:00:03	28:37.3	system	2013-03-08 13:28:37.313 - madison_m -> Now viewing tab Bisector
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	madison_	madison_	0:00:19	29:01.9	chat	2013-03-08 13:29:01.877 - madison_m -> ive added a tab its called....TEAM 4 TAB
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	emma_r	madison_	0:00:08	28:52.3	chat	2013-03-08 13:28:52.349 - emma_r -> why did you add a new tab madison>
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	emma_r	emma_r	0:00:01	28:54.8	chat	2013-03-08 13:28:54.817 - emma_r -> ?
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	emma_r	emma_r	0:00:00	28:57.3	chat	2013-03-08 13:28:57.335 - emma_r -> ?
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	emma_r	emma_r	0:00:02	28:59.6	system	2013-03-08 13:28:59.602 - emma_r -> Now viewing tab TEAM 4 TAB
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	emma_r	emma_r	0:00:16	29:16.2	wb	2013-03-08 13:29:16.190 - emma_r -> emma_r created a scribble
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	emma_r	emma_r	0:00:00	29:16.6	system	2013-03-08 13:29:16.629 - emma_r -> Now viewing tab Bisector
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	emma_r	emma_r	0:00:02	29:18.7	Geogebra:I	2013-03-08 13:29:18.654 - emma_r -> tool changed to Move
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	madison_	emma_r	0:00:34	30:04.0	chat	2013-03-08 13:30:03.979 - madison_m -> i feel abandond whats wrong guys?????? ;(
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	amina_p	madison_	0:00:22	29:55.0	chat	2013-03-08 13:29:55.023 - amina_p -> are finish with the instructions emma
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	emma_r	amina_p	0:00:00	29:48.5	chat	2013-03-08 13:29:48.485 - emma_r -> i get it
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	emma_r	emma_r	0:00:01	29:52.6	chat	2013-03-08 13:29:52.573 - emma_r -> geeeezzz
Spring 2013	Dynamic Geometry	Topic 03	Holland_Group_4	emma_r	emma_r	0:00:15	30:08.1	Geogebra:I	2013-03-08 13:30:08.068 - emma_r -> tool changed to Intersect Two Objects

Fig. 3. Sample logs from VMT.

All log data for this study centers on specific event types from VMT: *Awareness*, *Geogebra*, *System*, *Chat*, and *WhiteBoard* (*Wb*), and was collected in .txt format. The *Chat* event type logs all messages that students send to each other in the group. *Awareness* records the actions of erasing the chat messages on the chat bar. *Geogebra* logs information on how students virtually construct a geometry artifact (e.g. add a point, or update a segment etc.). The *System* event type records information related to how the VMT environment is accessed, logging when a student joins a virtual room, leaves a virtual room or views different tabs. *Wb* logs more specific actions on how tools are being used in the white board areas such as the resizing of objects, creating a textbox, etc. For every event type, we have logs of actions (adding a point, sending a chat, erasing a message, or creating a text box, etc.) that the student makes under what subjects (modules) as well as the starter (source) and receiver (target) of those actions and messages. In addition, the environment logs the information about when this action takes place (time) and in which virtual room (group) the event occurs. Fig. 3 shows a sample of original log data.

5 Methodology

5.1 Measure Construction

Since the log data is centered on event types and the facilitation of measure construction, we first process each event type into four participation dimensions: [*Individual, Group, Action Types, Module Set*], for each student. The *Individual* category is the sum of all personal actions (frequency) in which the source and the target of the action are the same (Fig. 3) in a given event type. Similarly, the *Group* category is the sum (frequency) of all actions the student makes in group projects in which the source and the target of the action are different in a given event type. The *Action Types* dimension is the number of types of actions that the student performs in a given event type. For example, if a student never erases a message in the *Awareness* event over all the modules, then the *Action Types* for *Awareness* is 0; for a *Wb* event, if a student takes the action of creating a textbox copying an object, but never uses other actions such as moving objects, resizing, etc. for the duration of the class, the *Action Types* participation dimension will equal 2. Some students may miss one or two modules. Therefore, the *Module Set* dimension records the distinct modules the student is involved in for a given event type. Rather than a single value, each module set is comprised of the modules in which events take place. In sum, the data is processed as a hierarchical structure of three levels, with individual student at the top, followed by the five event types on the middle level and the four measurements on the bottom level.

5.1.1 Subject

Subject in activity theory represents a student's individual efforts in problem solving. When mapped to our log data, individual effort can be reflected as the student actions over the five event types across all modules where he or she is both the initiator of the action (source) and the receiver of the action (target). This is spontaneous activity that is independent of the influence of other group members. An example of a *Subject* action is a participant who performs a series of 20 consecutive Geogebra actions with no input from other group members. As an action that is completed with little external influence, this is a reasonable demonstration of individual knowledge. The calculation for *Subject* returns the sum of the *Individua*l measure across all event types for a single participant.

5.1.2 Rules

According to Fig. 2, *Rules* includes implicit and explicit rules. Under the social-technical construct, the rules are the implicit rules of the VMT environment that constrain students' actions. Explicit rules are absent in this context as there are no instructors present to establish rules about collaboration or use of the tools. In this VMT context, students may only perform actions that the VMT environment offers, such as Segment, Circle, Point, and Compass. Therefore, *Rules* returns the sum of the types of actions the student uses across all the modules.

5.1.3 Tools

VMT tools facilitate the learning activity and mediate the transformation of objects. Within the VMT context, the *Tools* are *System* and *Wb* where the student's action for tool usage is registered. The *Tools* calculation returns the sum of all instances of both *Individual* and *Group* actions with respect to event types *System* (joining a room, viewing tabs, etc.) and *Wb* (resizing objects, creating a text box, etc.).

5.1.4 Community

Community includes all communications that maintain community structure. In the VMT context, students use chat to directly communicate with others, and can also erase chat messages, which can be categorized as an indirect contribution to the community and is labeled as *Awareness*. Therefore, *Community* is demonstrated as the total of *Group* and *Individual* for the *Chat* event type summed with the total of *Group* and *Individual* for the *Awareness* event type.

5.1.5 Division of Labor

Division of Labor indicates the comparative contributions of each group member to the collaborative learning modules. Though chat messages may also contribute to the development of the geometry object, concrete contributions to the geometry object construction is from the *Geogebra* dimension. Therefore, we use *Geogebra* to represent the student's *Division of Labor* aspect within the group. As a result, *Division of Labor* is calculated as the sum of *Individual* and *Group* for the *Geogebra* event type.

5.1.6 Object

The activities included in these modules are designed to achieve the object of a student's active involvement for the duration of the class. Hence, the first factor to consider is the number of distinct modules that students participate in. For example, from the *Module Set* measurement, we can obtain information such as a participant may use *Geogebra* in Modules 1,3, and 4, and use *Chat* in Modules 1, 2, and 3. This would result in a value of 4 in the *Object* dimension for the student. In order to properly quantify whether the student is active in those learning modules, the total frequency of participation and the number of event types are also incorporated. Doing this avoids inflated ratings for the student who participates in all modules but makes few actions or contributions in total.

On the other hand, the number of modules the student is involved in is the first factor considered by the *Object* dimension; overall frequency of *Individual* plus *Group* and the number of different event types used are secondary factors. Because the number of modules is in the scale of 10, while the frequency for participation is in the scale of 100, the *log* function is used to dampen the effect of the frequency measure. Though the event types are in the same scale of modules, we want to dial down the influence of the total number of distinct events types that the student used. Thus, we use a fraction to lower the effect of event type on *Object* measurement, characterized as the event type students are involved in divided by the number of event types (5). Finally, *Object* is calculated as the log of the frequency multiplied by the number of distinct modules multiplied by the number of distinct modules that the 5 event types are performed in, divided by 5.

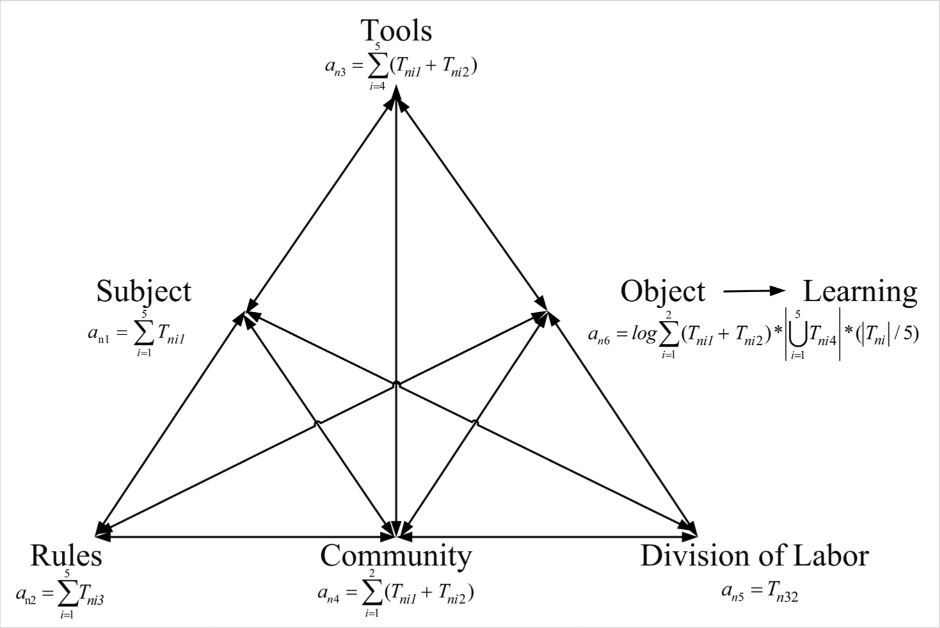

Tools
$a_{n3} = \sum_{i=4}^{5}(T_{ni1} + T_{ni2})$
Subject
$a_{n1} = \sum_{i=1}^{5} T_{ni1}$
Object
Learning
$a_{n6} = log\sum_{i=1}^{2}(T_{ni1} + T_{ni2}) * \left|\bigcup_{i=1}^{5} T_{ni4}\right| * (|T_{ni}| / 5)$
Rules
$a_{n2} = \sum_{i=1}^{5} T_{ni3}$
Community
$a_{n4} = \sum_{i=1}^{2}(T_{ni1} + T_{ni2})$
Division of Labor
$a_{n5} = T_{n32}$

Fig. 4. Activity theory quantification model for individual participation in CSCL. a_n represents the activity vector of student n. In the event type level, T_{ni} denotes the event type i of student n, meaning that there are five event types in total. Specifically, these five event types are *Awareness* (i=1), *Chat* (i=2), *Geogebra* (i=3), *System* (i=4), and *Wb* (i=5). In the measurement level, T_{nij} represents the value of measurement aspects j in event type i of student n, denoting four measurement aspects, *Individual* (j=1), *Group* (j=2), *Action Types* (j=3) and *Module Set* (j=4).

These integrated factors result in a quantified model based on activity theory that is built for individual student performance (Fig. 4) specific to the VMT environment: [*Subject, Rules, Tools, Community, Division of Labor, Object*]. In addition to providing a principled way for measure selection and construction, this theory-informed method reduces the data dimensionality to only 6 variables. These measures with the associated students' performance labels become the dataset and input for the GP algorithm to build the prediction model. In our context, there are 122 lines of data, each representing an individual student as [*Subject, Rules, Tools, Community, Division of Labor, Object, Performance Category*], where (*Subject, Rules, Tools, Community, Division of Labor, Object*) are independent variables/features and are used to predict the dependent variable/class – the *Performance Category*.

5.2 GP

GP is an evolutionary computation technique discussed in detail by Koza (1992), which can automatically generate approximate or exact solutions to a problem without telling the computer explicitly how to do so. GP can be considered as an extension of Genetic Algorithm (GA). The major distinction between GP and GA (Goldberg & Holland, 1988), lies in their representation of models. While the GA presents models as fixed length binary strings, the GP replaces models with tree-structured representations. Fig. 5 demonstrates an example of a GP function tree showing a rule that: IF X < Y THEN X > 7 AND Y > Z ELSE Y = 14. As illustrated in Fig. 5, 'branch' or inner nodes are functional with one or two arguments (such as >, =, *, Sin etc.), or Boolean arguments (such as AND, OR, NOT) or conditional operators (IF-THEN-ELSE etc.). 'Leaf' or terminal nodes represent the variables and constants (X, Y, Z, 7, 14). When it comes to solving a problem, variables and operators should be predetermined.

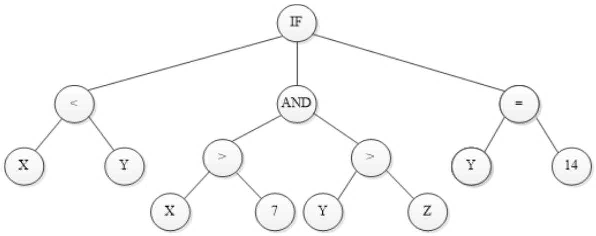


Fig. 5. Example of a GP function tree.

Generally, GP works with a population of models inspired by the Darwinian evolution process (Koza, 1992). GP starts with a certain number of models for a problem (prediction) where each model itself is a solution to the problem. Then relying on their fitness level, multiple models (parents) are stochastically selected to breed a new population of models (offspring) through genetic operations – crossover, selection and mutation. The generated offspring are then used in the next iteration of the algorithm. A GP model will stop when the number of generations reaches a pre-specified maximum, or the population reaches the predetermined fitness level. Hence, this evolution process is able to indirectly produce a better model for a given problem (Xu, Wang, & Liu, 2013). In turn, from an EDM perspective, GP is a good fit for building a student prediction model due to its optimization paradigm.

5.2.1 Genetic Operation

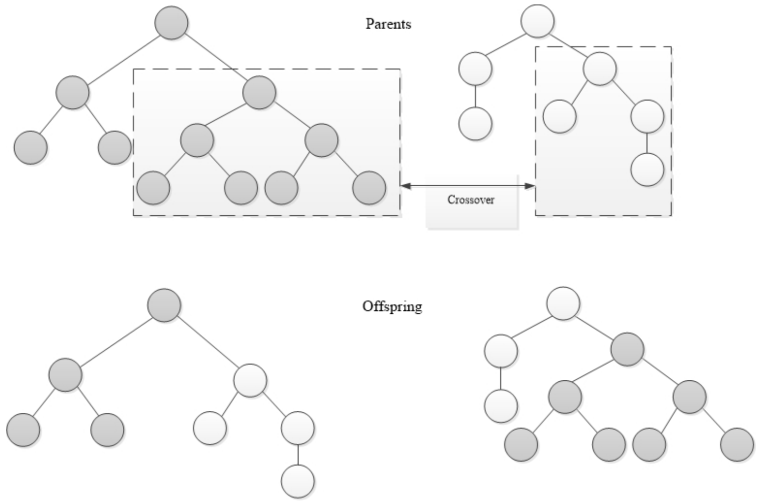


Fig. 6. GP crossover operation.

The generation of new models in GP usually results from three genetic operations: crossover, selection and mutation. GP genetic operators also include reproduction, but this operation merely selects a portion of models and places them into the next generation without any alterations. By contrast, the crossover operation creates a new model by recombining information from selected parents. Two parents interchange parts of their trees to produce two offspring relying on their fitness level as shown in Fig. 6.

The aim of mutation is to introduce new information to the population. Mutation is applied to a single model randomly selected based on their fitness level. A small portion of the tree is selected and altered according to the pre-specified terminals and functions as shown in Fig. 7. Mutation can also generate new models in the population.

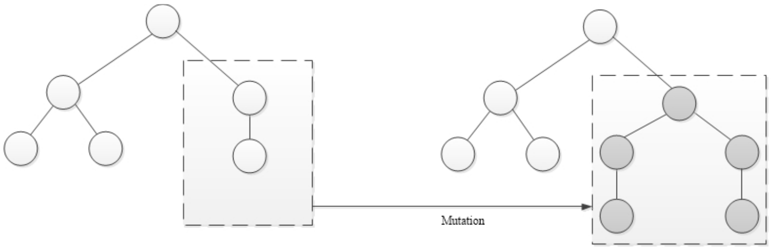


Fig. 7. GP mutation.

5.2.2 GP for Rule Discovery

In developing the GP algorithm, this study referred to the work done by Cano, Zafra & Ventura (2013) called Interpretable Classification Rule Mining (GP-ICRM), which is a variant of GP known as grammar-based GP (McKay et al., 2010). There are three main advantages in using GP-ICRM in this research context: 1) In general, grammar-based GP guarantees that every model generated is legal with respect to the specified grammar because it ensures that terminals (leaf nodes) and non-terminals (branch or functional nodes) combined can represent viable solutions and also that non-terminals are able to handle all values received as input; 2) The restricted search space defined by the grammar improves computational efficiency; 3) From a learning analytics standpoint, the resulting model is ready for teachers to interpret, allowing them to understand the cause for student at-risk or what aspect the student needs enhancement in order to improve performance .

To illustrate, even though operationalization of activity theory embeds the factors or variables in a semantic background, statistical models and many EDM models still represent the prediction model in mathematical format or as a black box. By contrast, GP-ICRM, a rule-based algorithm, is able to provide comprehensible rules, on one hand, by specifying operators in advance, where ‘>’, ‘≥’, ‘<’ ‘≤’ connect numerical attributes (activity theory-informed independent variables) and “=” and ‘≠’ connect categorical attributes (student performance) which are dependent variables. On the other hand, the interpretable rules also result from the predetermined grammar that specifies which relationship operations are allowed to appear in the antecedents of the rules and which attribute must appear in the consequents (Espejo et al., 2005). Because this study aims to predict student performance based on participation, measures informed by activity theory are the antecedents to the relationship operator and categorical data of student performance take the position of consequents after the relationship operator. The rule format adapted from (Espejo et al., 2005) is as below:

```
<Rule> :: =
IF <antecedent> THEN <consequent>
<antecedent>:: =
<condition> AND <condition>
<condition> | <condition>
<consequent>:: =
IS A <class label>
<condition>:: =
        <attribute> <rel operator> <value>
<attribute>:: =
        <Subject> <Rules> <Tools> <Community> <Division of Labor> <Object>
<rel operator>:: =
        = | ≠ | > | ≥ | < | ≤
<value>:: =
        Value in each corresponding domain
<class label>:: =
        EXCELLENT|GOOD|AVERAGE|SUFFICIENT | FAIL
```

5.2.3 Fitness function

The fitness function is an important component in GP and determines how well the model in the population can solve a problem (Xu, Wang, & Liu, 2013). In this study, a combination of two measures, sensitivity and specificity, are used. These measures can be calculated using a confusion matrix (Table 2) which allows detailed analysis of the model prediction performance. The confusion matrix provides a more reliable way to measure the real performance of a prediction model than an accuracy metric, which would yield misleading results if the dataset were unbalanced (Kohavi and Provost, 1998). For instance, if 95 students of 100 are passing the course with 5 at risk, a prediction model that marks all students as successful can still have an overall accuracy as high as 95%. While the model may have a 100% identification rate for students passing the class, it has a 0% recognition rate for struggling students, which renders it less than usable. Considering that most of the data given to student performance prediction model may be unbalanced (only a small portion of at-risk students in most courses), it is ideal to use confusion matrix to serve as the base for fitness function calculation.

Table 2
Confusion matrix.

Predict Actual	Positive	Negative
Positive	True positive A	False positive B
Negative	False negative C	True negative D

In Table 2, A is the number of correct predictions that an instance is positive; B is the number of incorrect predictions that an instance is negative; C is the number of incorrect predictions that an instance is positive; and D is the number of correct predications that an instance is negative.

Sensitivity is the proportion of actual positives which are predicted to be positive:

$$Sensitivity = \frac{\mathrm{A}}{\mathrm{A+C}}.$$

Specificity is the proportion of actual negatives which are predicted to be negative:

$$Specificity = \frac{\mathrm{D}}{\mathrm{B+D}}.$$

Then in order to maximize the accuracy and prevent problems associated with imbalanced data, the fitness function is calculated as the product of sensitivity and specificity.

$$Fitness = \frac{A * D}{(A+C)*(B+D)}.$$

5.2.4 GP workflow for Rule Discovery

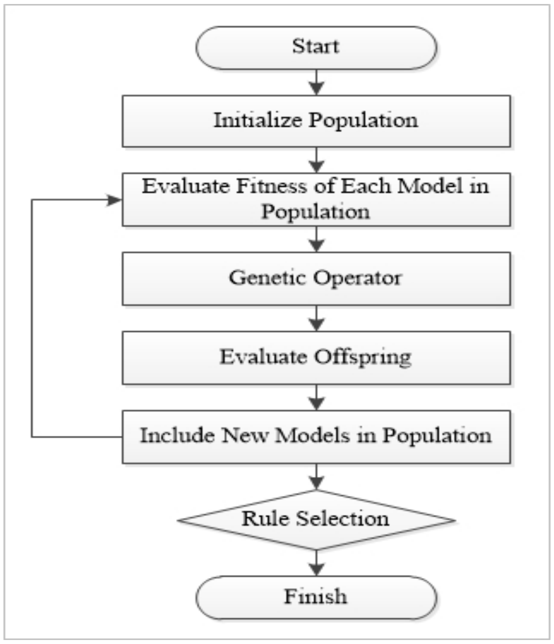


Fig. 8. GP algorithm workflow

The GP model is based on an iterative computational process used to solve problems as shown in Fig. 8, following these steps:

1) Initialization. Randomly produce a population of N models that represent potential solutions to the prediction of student performance.
2) Apply each model in the current population on the training data and evaluate the fitness of each model in the current population.
3) Choose the parent models and genetic operators probabilistically to produce offspring models until the predetermined population size has been reached.
4) Replace the N old models by new generated N models.
5) Repeat steps 2 - 4 until the predefined maximum generations reached.
6) The rule with the best fitness level is the result of the GP-ICRM algorithm – the student performance prediction model.

5.3 Experiment

Table 3
Prediction model execution, classified by ease of user understandability as reported by (Romero et al., 2013) and others as noted in our literature review.

Easier to Understand Models			More Difficult to Understand Models		
Rule-based Model		Decision-Tree	Statistical Model	Artificial Neural Network	Bayesian Network
GP-ICRM	NNge	RandomTree	Logistic Regression	Perceptron	Naïve Bayes

To strengthen our research, we executed various traditional prediction algorithms to benchmark the proposed GP model. All prediction models that were implemented are shown in Table 3. Specifically, GP-ICRM was implemented in JCLEC (Ventura et al., 2008), a Java framework for evolutionary computation. The remaining algorithms were developed in Matlab and Java. All the algorithms are evaluated using 10-fold cross validation and 10 different runs for each partition. As discussed earlier, since statistical models and Bayesian Networks may not necessarily be black box models, we grouped them together with Perceptron as models that are difficult to understand from users' perspective, and GP-ICRM, NNge and Random Tree as easy to understand due to the resulting rule-set format.

6 Result & Analysis

6.1 Activity theory based measures

In order to reduce the data dimensionality and contextualize the data for instructors, this study built measures around students' participation in a course derived from activity theory. As a result, each student can be represented by a 6 dimensional set with a semantic background as illustrated in Table 4. In fact, by looking the Table 4 alone, instructors could already obtain meaningful information. Through simply comparing students by column, the instructor is able to tell which student performs well in that dimension. For example, if Student P scored lowest in *Community*, the teacher could advise the student to communicate more with his or her team members. Activity theory equips us with a holistic way to describe students' participation performance in a CSCL environment rather than via ad-hoc guesswork. These quantified results provide semantic clues that instructors may use to investigate student performance, which is difficult to infer from feature selection algorithm results due to its mathematical nature. Merely comparing measures among students is not a valid or reliable way to predict student's performance. A prediction model, in this case built by GP, is able to fill this void.

Table 4
Sample student participation measures based on activity theory.

Name	*Subject*	*Rules*	*Tools*	*Div. of Labor*	*Community*	*Object*
A	220	11	13	246	45	3.972598
M	563	8	31	576	38	4.495296
P	277	10	80	340	26	4.238936
S	878	21	94	541	119	9.866061
W	335	18	56	310	77	8.468492
⋮	⋮	⋮	⋮	⋮	⋮	⋮

6.2 GP Prediction performance

Table 5
Student final performance prediction result.

Algorithm Result	Prediction Result	Easy to Understand Model			Difficult to Understand Model		
		Rule-based Model		Decision-Tree	Statistical Model	Artificial Neural Network	Bayesian Network
		GP-ICRM	NNge	RandomTree	Logistic Regression	Perceptron	Naïve Bayes
Fitness	Overall Prediction	80.2%	76.2% **(.000)**	72.6% **(.000)**	77.1% **(.000)**	36.6% **(.000)**	77.7% **(.000)**
	At-Risk Prediction	89.5%	82.1% **(.000)**	82.1% **(.000)**	81.1% **(.000)**	42.1% **(.000)**	94.7% (.784)
Sensitivity	Overall Prediction	80.3%	76.8% **(.000)**	72.7% **(.000)**	77.2% **(.000)**	38.9% **(.000)**	78.2% (.018)
	At-Risk Prediction	85.0%	76.2% **(.000)**	76.2% **(.000)**	78.9% **(.000)**	66.7% **(.000)**	90.0% **(.000)**
Specificity	Overall Prediction	80.3%	76.2% **(.000)**	73.0% **(.000)**	77.0% **(.000)**	36.4% **(.000)**	77.9% **(.000)**
	At-Risk Prediction	94.4%	88.9% (.028)	88.9% **(.000)**	83.3% **(.000)**	30.8% **(.000)**	100% **(.000)**

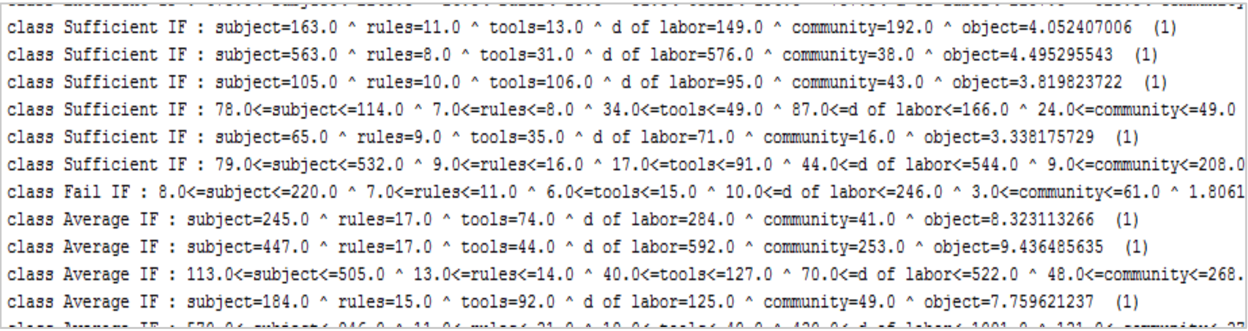
Our objective was to compare the GP model prediction performance with the benchmark models. The weighted results after executing 10-fold cross-validations and 10 different runs are shown in Table 5. This table shows the mean values of Fitness, Sensitivity and Specificity for each prediction model in both Overall Prediction (includes 5 performance categories) and specific At-Risk prediction (only students with at-risk label). Further, to increase the reliability of these values, paired t test (Lawrence & Lin, 1989) was performed between GP-ICRM with each benchmark models setting significant level at 0.05. Those p-values are shown in parentheses and with bold letters indicating a significant difference.

Fitness is an overall reflection of the prediction performance for each model. According to Table 5 of Fitness values for Overall Prediction, GP-ICRM has the best prediction result (80.2%) across all algorithms with a significant result. GP-ICRM generally outweighs an average 6% over the NNge and RandomTree white box algorithms. In terms of comparison to difficult to understand models, the GP-ICRM model outperforms every baseline model by around 3%. This has already shown the advantage of GP-ICRM because black-box or difficult to understand models usually have better prediction rate than white-box models (easy to understand models) (Romero et al., 2010; Bernardo, Hagras & Tsang, 2013). Similarly, a significant result is also obtained on Specificity side for Overall Prediction, which GP-ICRM outperforms the rest of the models. GP-ICRM has significantly better performance in Sensitivity for Overall Prediction as well except when it compares to Naïve Bayes model. Generally speaking, GP-ICRM is stable and the best choice for Overall Prediction model.

On the other hand, identifying at-risk students is the classic goal for education prediction models. Table 5 also presents the specific prediction performance of the test model and baseline models on detecting struggling students who risk failing the course. Unlike Overall Prediction performance, GP-ICRM produces mixed results in identifying at-risk students. GP-ICRM outperforms NNge, RandomTree, Logistic Regression, and Perceptron models significantly in almost all the three aspects –Fitness, Sensitivity and Specificity. However, a comparison with the Naïve Bayes model demonstrates that even though it does not produce a significant result on Fitness value, Naïve Bayes does outperform GP-ICRM on At-Risk Prediction significantly on both Sensitivity and Specificity respects. In fact, even for Overall Prediction, the difference between these two models is small. This is understandable, as Naïve Bayes is a very robust model. In empirical tests, this model has often outperformed more sophisticated models such as decision trees, general Bayesian networks, and rule-based algorithms, especially in binary (success and failure) classification tasks (Langley, Iba, & Thompson, 1992; Domingos, P., & Pazzani, 1997; Hellerstein, Jayram, & Rish, 2000). Aiming for a model that is easily understood by users often comes at the price of decreased performance, so trade-offs between model understandability and model performance need to be taken into account (Huysmans et al., 2011; Freitas, Wieser & Apweiler, 2010). Thus, in our context from EDM angle, GP-ICRM meets our intent and requirements with the ideal prediction performance. As a white-box model and with the best understandability from the user's perspective discussed in the next section, GP-ICRM has outperformed the black box and white box models in performance prediction and is comparable to Naïve Bayes in detecting at-risk students.

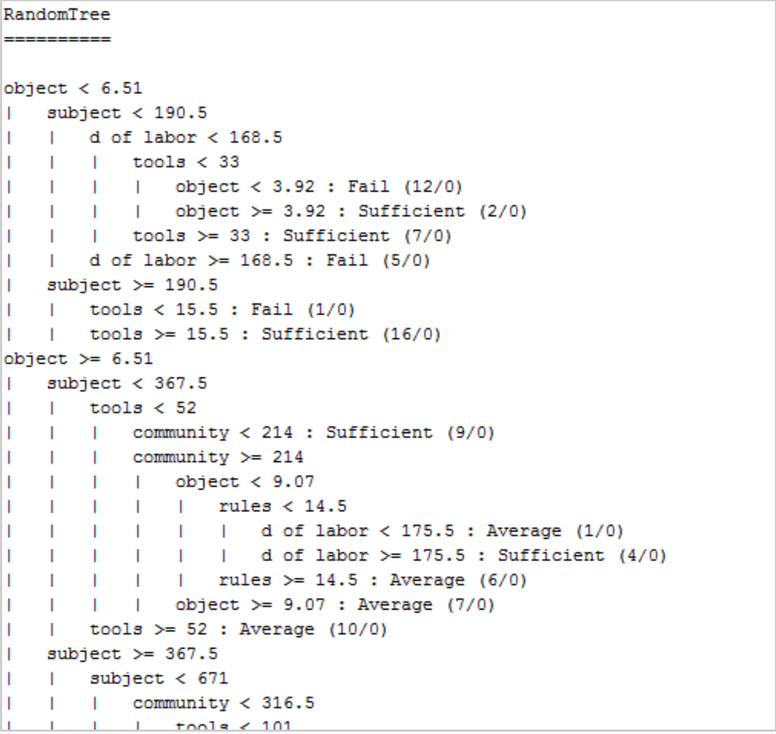
6.3 Sample model

When approaching prediction model through the lens of learning analytics, we consider understandability of different models, which forms a base for teachers to offer concrete and individualized suggestions to students. Granted, understandability is a subjective concept which depends on many factors outside the model, such as the user's experience and his/her prior knowledge. Some representation formats, especially the "if-then" rules set, are generally considered to be more easily interpretable than others (Freitas, Wieser & Apweiler, 2010; Huysmans et al., 2011; Romero et al, 2013). However, simply using "if-then" rules for model representation does not guarantee that the discovered knowledge is understandable. If the number of discovered rules and/or rules conditions is very large, the discovered knowledge can hardly be called comprehensible (Freitas, 2002). In fact, it is routine to use the number of rules and conditions in each rule to measure the understandability of the rule-based models (Freitas, Wieser & Apweiler, 2010; Cano, Zafra & Ventura, 2010; Freitas, 2002). Based on the previous discussion, we implement four standards to measure the understandability of the models generated: 1) whether it is a white box model 2)whether the model can be presented as if-then rules 3) number of rules contained in the model 4) number of conditions in the rules. Fig. 9, Fig. 10 and Fig. 11 show part of the easy to understand models produced by NNge algorithm, RandomTree algorithm and GP-ICRM algorithm respectively.



```
class Sufficient IF : subject=163.0 ^ rules=11.0 ^ tools=13.0 ^ d of labor=149.0 ^ community=192.0 ^ object=4.052407006  (1)
class Sufficient IF : subject=563.0 ^ rules=8.0 ^ tools=31.0 ^ d of labor=576.0 ^ community=38.0 ^ object=4.495295543  (1)
class Sufficient IF : subject=105.0 ^ rules=10.0 ^ tools=106.0 ^ d of labor=95.0 ^ community=43.0 ^ object=3.819823722  (1)
class Sufficient IF : 78.0<=subject<=114.0 ^ 7.0<=rules<=8.0 ^ 34.0<=tools<=49.0 ^ 87.0<=d of labor<=166.0 ^ 24.0<=community<=49.0
class Sufficient IF : subject=65.0 ^ rules=9.0 ^ tools=35.0 ^ d of labor=71.0 ^ community=16.0 ^ object=3.338175729  (1)
class Sufficient IF : 79.0<=subject<=532.0 ^ 9.0<=rules<=16.0 ^ 17.0<=tools<=91.0 ^ 44.0<=d of labor<=544.0 ^ 9.0<=community<=208.0
class Fail IF : 8.0<=subject<=220.0 ^ 7.0<=rules<=11.0 ^ 6.0<=tools<=15.0 ^ 10.0<=d of labor<=246.0 ^ 3.0<=community<=61.0 ^ 1.8061
class Average IF : subject=245.0 ^ rules=17.0 ^ tools=74.0 ^ d of labor=284.0 ^ community=41.0 ^ object=8.323113266  (1)
class Average IF : subject=447.0 ^ rules=17.0 ^ tools=44.0 ^ d of labor=592.0 ^ community=253.0 ^ object=9.436485635  (1)
class Average IF : 113.0<=subject<=505.0 ^ 13.0<=rules<=14.0 ^ 40.0<=tools<=127.0 ^ 70.0<=d of labor<=522.0 ^ 48.0<=community<=268.
class Average IF : subject=184.0 ^ rules=15.0 ^ tools=92.0 ^ d of labor=125.0 ^ community=49.0 ^ object=7.759621237  (1)
```

Fig. 9. NNge model



```
RandomTree
==========

object < 6.51
|   subject < 190.5
|   |   d of labor < 168.5
|   |   |   tools < 33
|   |   |   |   object < 3.92 : Fail (12/0)
|   |   |   |   object >= 3.92 : Sufficient (2/0)
|   |   |   tools >= 33 : Sufficient (7/0)
|   |   d of labor >= 168.5 : Fail (5/0)
|   subject >= 190.5
|   |   tools < 15.5 : Fail (1/0)
|   |   tools >= 15.5 : Sufficient (16/0)
object >= 6.51
|   subject < 367.5
|   |   tools < 52
|   |   |   community < 214 : Sufficient (9/0)
|   |   |   community >= 214
|   |   |   |   object < 9.07
|   |   |   |   |   rules < 14.5
|   |   |   |   |   |   d of labor < 175.5 : Average (1/0)
|   |   |   |   |   |   d of labor >= 175.5 : Sufficient (4/0)
|   |   |   |   |   rules >= 14.5 : Average (6/0)
|   |   |   |   object >= 9.07 : Average (7/0)
|   |   tools >= 52 : Average (10/0)
|   subject >= 367.5
|   |   subject < 671
|   |   |   community < 316.5
|   |   |   |   tools < 101
```

Fig. 10. RandomTree model

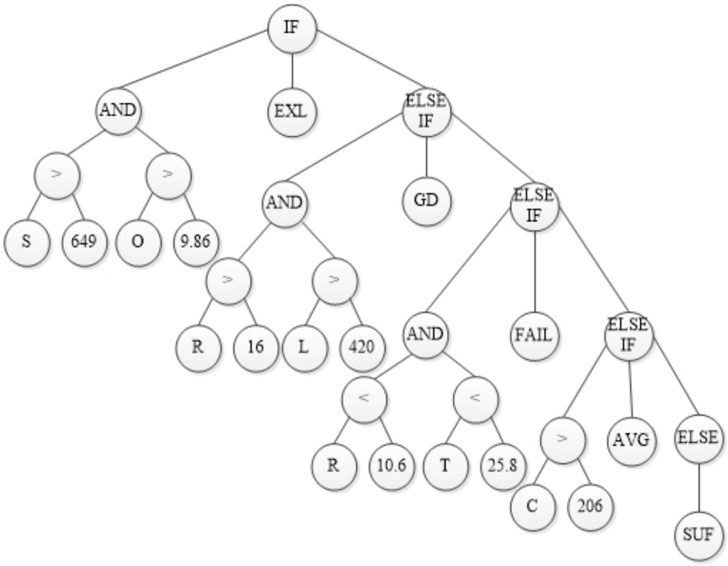


Fig.11. GP-ICRM model

Compared to these three models, NNge (Fig. 9) has the lowest understandability because each discovered rule is very long and includes multiple of conditions. In addition, these rules have OR operators, requiring more time and effort to assign the student to the proper category. NNge produced 27 rules in total. Although RandomTree (Fig. 10) generates a model in tree structure, it can be transferred into IF-THEN-ELSE structure as below:

```
IF (Object  < 6.51) THEN
{           IF (Subject < 190.5) THEN
{           IF (Division of Labor  < 168.5) THEN
            {           IF (Tools < 33) THEN
                        {           IF (Object < 3.92) THEN {Result = FAIL}
                                    IF (Object >= 3.92) THEN {Result = SUFFICIENT}
                        }
            }
}
ELSE IF (Subject >= 190.5) THEN
{           IF (Tools < 15.5) THEN {Result = FAIL}
            IF (Tools >= 15.5) THEN {Result = SUFFICIENT}
}
}
ELSE IF(Object  >=  6.51) THEN …
```

RandomTree organizes the conditions into a hierarchical structure (Luke, 2000). Therefore, it requires less efforts to determine a student's performance because classification always begins at the root of the tree and ends when it arrives at a leaf. Also, the RandomTree structure does not include the OR operator. The size of the tree is 47, indicating that there are 47 rules used to predict students' performance but with less conditions in each rule. GP-ICRM has a format similar to the RandomTree algorithm (Fig. 11) with IF-THEN-ELSE rules:

```
IF (Subject >= 649 AND Object >= 9.86) THEN (Result = EXCELLENT)
ELSE IF (Rules >= 16 AND Division of Labor >= 420) THEN (Result = GOOD)
ELSE IF (Rules <= 10.6 AND Tools <= 25.8) THEN (Result = FAIL)
ELSE IF (Community > = 206) THEN (Result = AVERAGE)
ELSE Result = SUFFICIENT
```

However, GP-ICRM is much simpler with five rules in total, each rule having one or two conditions. Therefore, in addition to the higher prediction performance, GP-ICRM also has an advantage in understandability and interpretability compared with other white-box models. For example, in conjunction with Table 4, the teacher could infer that Student A is struggling in the course and at-risk of failure because he or she is falling short in the *Tools* dimension (13 < 16). The teacher could then encourage the student to explore different functions in the VMT environment and discuss any difficulties the student may have with using the various tools. Also, Student W gets an AVERAGE performance label according to the rule because W does not satisfy the first three rules but does satisfy the fourth rule in *Community* (310 > 206.1). In order to be moved to GOOD standing, he or she needs to work more on the *Division of Labor* because he satisfied all the conditions on GOOD performance except *Division of Labor* (310 < 420.335). Since *Division of Labor* is mainly influenced by the Geogebra event type, the teacher could advise the students to put more effort into this dimension, which concerns concretely constructing the geometry objects. Based on the GP generated comprehensible model, teachers are able to identify the reasons of students' failure or the general performance level of students, enabling them to provide more individualized advice and feedback to more students for performance improvement. On the other hand, previous discussions are all from teacher's perspective. In fact, students can be easily granted access to those prediction results which may have more powerful influence for students' awareness and reflection for learning.

From an engineering/application perspective, it takes much less energy to write the rules into an application for VMT than it does when using black-box models. This rule-based application also requires less computing power. Statistical models can show tangible relationships between independent and dependent variables, putting it in a readable format, but it cannot contextualize the model for teacher to interpret. Hence, similar to other black-box models, it is not suitable here to build a performance prediction model. Compared with other white-box models, GP-ICRM generated simpler rules in number and format.

7. Discussion

Building a practical and interpretable student performance prediction model is a shared goal for learning analytics and EDM. It is a difficult task not only because factors involved can be overwhelming but also because lack of semantic background for teachers to interpret the model developed. Most previously developed models identified at-risk students, but were unable to predict student performance in a more granular level. As an exploratory study, this study first narrowed down the factors scope into students' participation based on the theory introduced by Hrastinski (2009). Then in order to systematically describe participation as well as contextualize the data, activity theory was employed to work as a semantic base for those variables. While activity theory helps accomplish the data contextualization and holistically describe participation, it also automatically reduces data dimensionality to only 6 variables. Next, this paper applied GP to build the prediction performance model which generated a more accurate and understandable rule format compared with other modeling techniques. GP-ICRM model presents a comprehensible format for teachers to identify reasons that students are struggling or to account for the performance level that a student has at a particular time, enabling the teacher to provide more concrete and individualized suggestions to each student. Students may also be able to increase their awareness and reflective learning if they are able to access this prediction information. Finally, the generated model is easily implemented in a real learning environment.

7.1. Theoretical implications

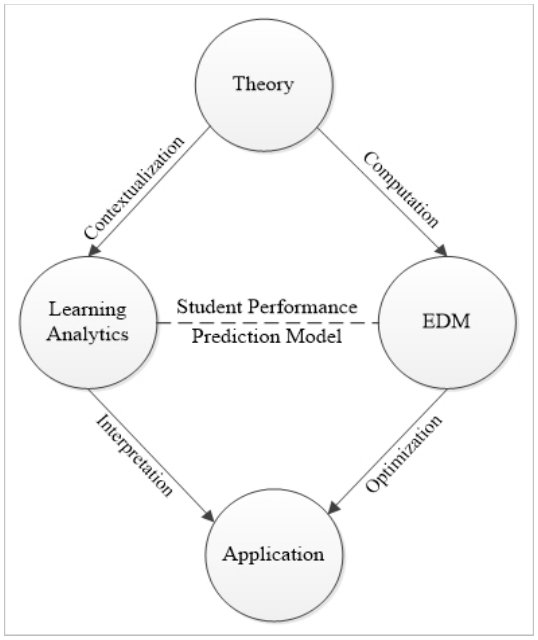


Fig. 12. Theoretical framework for student performance prediction model.

In this study we have applied and integrated four different approaches to student performance prediction modeling: learning analytics, EDM, applied practice, and HCI theory. Previous research has focused on single elements of this approach (Gunnarsson & Alterman, 2012; Shum & Ferguson, 2012; Romero & Ventura, 2010; Zafra & Ventura, 2009). What has been lacking are links between theory and computation, optimization and interpretation so to develop an easy application prediction model in a real life learning environment. In order to develop practical and interpretable student performance prediction applications, we proposed the theoretical framework (Fig. 12) for the research community in this field to refer to.

Useful student performance prediction models employ theory to guide the research and development process. One practical way to accomplish this is to refer to the research context whether it is a face to face program or online course or hybrid one, an independent study course, or collaborative learning course or one of mixed type. In our study, learning as participation is the theory that led us to focus solely on factors related to participation. Activity theory was chosen to contextualize our data and facilitate measure construction. Then learning analytics requires that the model be understandable and interpretable for teachers, enabling them to individualize the education process (actionable intelligence). EDM requires a model with optimized prediction results. In our case, GP outperforms all baseline models in overall performance prediction, but the Naive Bayes model achieved the best results in terms of detecting at-risk students. Considering these pros and cons, GP-ICRM is still the preferable choice. Compared with the rule format and computing power requirement with the white-box models as well as black-box models, GP is ideal for the development of a student performance prediction model due to its easy implementation.

7.2. Practical implications

Numerous factors influence learning. This study shows the potential for predictions of student's final performance to be inducted from student participation. Besides the traditional feature selection algorithm and ad-hoc guesswork, researchers can refer to theory for factor selection and construction. In a CSCL context, activity theory is a useful approach. Application of theory is very powerful for analytics because the components of a theory (in this case, activity theory) are understandable by users interpreting the resulting analytics (Halverson, 2002). Being able to map data along with the names in activity theory system affords an additional rhetorical advantage, providing teachers with a common language to draw comparisons across their experiences working with a particular tool. A human subject theory based factor construction and selection method serves as the background for teachers to interpret the prediction model. On the other hand, the available dataset for education is quite small and limited (except for MOOCs). Even if some modeling algorithms (such as Artificial Neural Networks) have a good prediction rate in other disciplines, those algorithms do not necessarily perform equally well when addressing educational problems in each case. This potential gap is due to limitations in the availability of training data. GP may serve as a starting point when considering prediction model construction in education due to its power of working with small datasets.

It is important to represent these analyses as comparative, not as absolute measures of performance. Distinguishing between patterns of student performance and participation will make it easier for teachers to key in on groups of students that are being very successful and students that are struggling. For these reasons, we prefer the term "indicator" over "measure" when describing the results of our work to teachers. We will iterate and evaluate many designs in classroom situations in the near future. Design based research iterations will be our path forward, and these will create increasingly useful learning analytic indicators.

7.3. Implications for Design

Representing summary performance information in technology mediated classrooms is a difficult challenge. Years of work by members of the author team on VMTwG with the Math Forum at Drexel focuses on bridging the gulf between how teachers experience and interpret summary information presented to them. Early results showing basic information, like number of actions or time in a VMTwG room removed some of the "black box" of what students were most active; but the challenge of representing performance information in a useful way remains. This article contributes a promising, new approach that blends automated data processing with specific approaches that users are more likely to be able to interpret.

Future designers of learning analytic systems focused on providing teacher overviews should consider the fundamental, logical and algorithmic approaches demonstrated here. Analytics will be more understandable if the data is structured for interpretation using theory; ideally a theory that is understandable for teaching a particular subject in a particular context. Here, we use activity theory as a lens. Through this theoretically organized data, we build a model for performance prediction using a GP algorithm. GP exposes the logic to end users in a visual way that makes it more understandable. We are initiating extensive user studies as a core of our future work.

8. Conclusion

This paper described a methodology which connected perspectives from learning analytics, EDM, theory and application to solve the problem of predicting students' performance in a CSCL learning environment with actual small datasets. We operationalized activity theory to holistically quantify student participation in the environment. We then coded an advanced GP technique to construct the prediction model. Results show that the GP based model is interpretable and has an optimized prediction rate as compared to the traditional modeling algorithms. In terms of the uniqueness of the educational field, where human judgment is a key factor, we developed a theoretical framework to guide future performance prediction model research and application. We also outlined practical recommendations that can leverage the best prediction model among the available algorithms. Theoretically, we argue that to build a student performance prediction model, learning analytics and EDM have to work under the guidance of educational theories to create an applicable model. Practically, measure and algorithm selection and construction are the keys to a successful student performance prediction model.

The proposed method has two limitations. In our measure construction, we did not consider the quality of ultimate artifacts or objects that may be generated at the end of a course. In addition, communication and language is also a powerful way of learning (Fromkin & Hyams, 2009). Lesser consideration of the qualitative aspects of collaborative work in measure construction is one of the limitations of our proposed method. Secondly, researchers less familiar with the VMT environment and without experience analyzing interactions in the environment may have a difficult time replicating our results in a different context. There are a few directions for future work: first, this study considers the quantitative aspect of the log data, future work could incorporate qualitative aspect into the activity theory system. For example, using natural language processing to process the chat logs of students, and then adding more factors to the *Community* dimension to see whether it can improve the prediction rate; second, researchers can carry out more experiments in other learning environments using this methodology and test its transferability; third, model comprehensibility is a subjective concept. Future studies can test the effectiveness of model understandability of different model representations and valuate how teachers interpret and use the indicators

References

Afzal, W., Torkar, R., Feldt, R., & Gorschek, T. (2010). Genetic programming for cross-release fault count predictions in large and complex software projects. *Evolutionary Computation and Optimization Algorithms in Software Engineering*, 94-126.

Agudo-Peregrina, Á. F., Iglesias-Pradas, S., Conde-González, M. Á., & Hernández-García, Á. (2013). Can we predict success from log data in VLEs? Classification of interactions for learning analytics and their relation with performance in VLE-supported F2F and online learning. *Computers in Human Behavior*.

Baker, R. S. J. D., & Yacef, K. (2009). The state of educational data mining in 2009: A review and future visions. *Journal of Educational Data Mining*, *1*(1), 3-17.

Barab, S. A., Barnett, M., Yamagata-Lynch, L., Squire, K., & Keating, T. (2002). Using activity theory to understand the systemic tensions characterizing a technology-rich introductory astronomy course. *Mind, Culture, and Activity*, *9*(2), 76-107.

Barber, R., & Sharkey, M. (2012). Course correction: using analytics to predict course success. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge* (pp. 259-262). ACM.

Basharina, O. K. (2007). An activity theory perspective on student-reported contradictions in international telecollaboration. *Language Learning & Technology*, *11*(2), 82-103.

Bernardo, D., Hagras, H., & Tsang, E. (2013). A genetic type-2 fuzzy logic based system for the generation of summarised linguistic predictive models for financial applications. *Soft Computing*, *17*(12), 2185-2201.

Shum, S. B., & Ferguson, R. (2012). Social Learning Analytics. *Journal of educational technology & society*, *15*(3).

Calvo-Flores, M. D., Galindo, E. G., Jiménez, M. P., & Piñeiro, O. P. (2006). Predicting students' marks from Moodle logs using neural network models. *Current Developments in Technology-Assisted Education*, *1*, 586-590.

Campbell, J. P., DeBlois, P. B., & Oblinger, D. G. (2007). Academic analytics: A new tool for a new era. Educause Review, 42(4), 40.

Cano, A., Zafra, A., & Ventura, S. (2013). An interpretable classification rule mining algorithm. *Information Sciences*, *240*, 1-20.

Cetintas, S., Si, L., Xin, Y. P., & Hord, C. (2009). Predicting Correctness of Problem Solving from Low-level Log Data in Intelligent Tutoring Systems. In *EDM* (pp. 230-239).

Deegalla S., Bostrom H. (2006) Reducing high-dimensional data by principal component analysis vs. random projection for nearest neighbor classification. In: *International conference on machine learning and applications*, pp 245–250

Dennen, V. P. (2008). Looking for evidence of learning: Assessment and analysis methods for online discourse. *Computers in Human Behavior*, *24*(2), 205-219.

Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, *40*(2), 139-157.

Engeström, Y. (1987). Learning by expanding. An activity-theoretical approach to developmental research.

Fancsali, S. E. (2011). Variable construction for predictive and causal modeling of online education data. In *Proceedings of the 1st International Conference on Learning Analytics and Knowledge* (pp. 54-63). ACM.

Espejo, P. G., Romero, C., Ventura, S., & Herrera, F.(2005). Induction of classification rules with grammar-based genetic programming. In Proceedings of the 2nd Int. Conf. Mach. Intell. (ACIDCA ICMI) (pp. 596-601), Tozeur, Tunisha.

Espejo, P. G., Ventura, S., & Herrera, F. (2010). A survey on the application of genetic programming to classification. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, *40*(2), 121-144.

Essa, A., & Ayad, H. (2012). Student success system: risk analytics and data visualization using ensembles of predictive models. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge* (pp. 158-161). ACM.

Fromkin, V., Rodman, R., & Hyams, N. M. (2009). *An introduction to language*. Cengage Learning.

Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine learning*, *3*(2), 95-99.

Gress, C. L., Fior, M., Hadwin, A. F., & Winne, P. H. (2010). Measurement and assessment in computer-supported collaborative learning. *Computers in Human Behavior*, *26*(5), 806-814.

Gunnarsson, B. L., & Alterman, R. (2012). Predicting failure: a case study in co-blogging. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge* (pp. 263-266). ACM.

Hämäläinen, W., & Vinni, M. (2006). Comparison of machine learning methods for intelligent tutoring systems. In *Intelligent Tutoring Systems* (pp. 525-534). Springer Berlin Heidelberg.

Harrell, F. E. (2001). *Regression modeling strategies: with applications to linear models, logistic regression, and survival analysis*. Springer.

Halverson, C. A. (2002). Activity theory and distributed cognition: Or what does CSCW need to DO with theories?. *Computer Supported Cooperative Work (CSCW)*, *11*(1-2), 243-267.

Hrastinski, S. (2009). A theory of online learning as online participation. *Computers & Education*, *52*(1), 78-82.

Ibrahim, Z., & Rusli, D. (2007). Predicting students‟ Academic Performance: Comparing Artificial Neural Network, Decision Tree and Linear Regression. In *Proceedings of the 21 Annual SAS Malaysia Forum, Kuala Lumpur, Malaysia* (pp. 1-6).

Jonassen, D. H., & Land, S. M. (2000). Preface. In D. H. Jonassen & S. M. Land (Eds.), Theoretical foundations of learning environments (pp. 3–9). New Jersey: Lawrence Erlbaum.

Kohavi, R., & Provost, F. (1998). Glossary of terms. *Machine Learning*, *30*(2-3), 271-274.

Kotsiantis, S. B., & Pintelas, P. E. (2005). Predicting students marks in hellenic open university. In *Advanced Learning Technologies, 2005. ICALT 2005. Fifth IEEE International Conference on* (pp. 664-668). IEEE.

Koza, J. R. (1992). *Genetic Programming: vol. 1, On the programming of computers by means of natural selection* (Vol. 1). MIT press.

Leont'ev, A. N. (1974). The problem of activity in psychology. *Journal of Russian and East European Psychology*, *13*(2), 4-33.

Luke, S. (2000). Two fast tree-creation algorithms for genetic programming.*Evolutionary Computation, IEEE Transactions on*, *4*(3), 274-283.

Márquez-Vera, C., Cano, A., Romero, C., & Ventura, S. (2013). Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data. *Applied Intelligence*, 1-16.

Mckay, R. I., Hoai, N. X., Whigham, P. A., Shan, Y., & O'Neill, M. (2010). Grammar-based genetic programming: a survey. *Genetic Programming and Evolvable Machines*, *11*(3-4), 365-396.

Mintrop, H., & Sunderman, G. L. (2009). Predictable failure of federal sanctions-driven accountability for school improvement—and why we may retain it anyway. *Educational Researcher*, *38*(5), 353-364.

Myller, N., Suhonen, J., & Sutinen, E. (2002). Using data mining for improving web-based course design. In *Computers in Education, 2002. Proceedings. International Conference on* (pp. 959-963). IEEE.

Nardi, B. A. (Ed.). (1996). *Context and consciousness: Activity theory and human computer interaction*. The MIT Press.

Nasiri, M., & Minaei, B. (2012). Predicting GPA and academic dismissal in LMS using educational data mining: A case mining. In *E-Learning and E-Teaching (ICELET), 2012 Third International Conference on* (pp. 53-58). IEEE.

Nebot, A., Castro, F., Vellido, A., & Mugica, F. (2006). Identification of fuzzy models to predict students performance in an e-learning environment. In *The Fifth IASTED International Conference on Web-Based Education, WBE* (pp. 74-79).

Ni, Q., Wang, L., Zheng, B., & Sivakumar, M. (2012). Evolutionary algorithm for water storage forecasting response to climate change with small data sets: The Wolonghu Wetland, China. *Environmental Engineering Science*,*29*(8), 814-820.

Pardos, Z. A., Beck, J. E., Ruiz, C., & Heffernan, N. T. (2008). The composition effect: Conjunctive or compensatory? An analysis of multi-skill math questions in ITS.

Pardos, Z. A., Heffernan, N. T., Anderson, B., & Heffernan, C. L. (2007). The effect of model granularity on student performance prediction using Bayesian networks. In *User Modeling 2007* (pp. 435-439). Springer Berlin Heidelberg.

Roberge, D., Rojas, A., & Baker, R. (2012). Does the length of time off-task matter?. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge* (pp. 234-237). ACM.

Romero, C., López, M. I., Luna, J. M., & Ventura, S. (2013). Predicting students' final performance from participation in on-line discussion forums.*Computers & Education*, *68*, 458-472.

Romero, C., & Ventura, S. (2010). Educational data mining: a review of the state of the art. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, *40*(6), 601-618.

Romero, C., Ventura, S., Espejo, P. G., & Hervás, C. (2008). Data Mining Algorithms to Classify Students. In *EDM* (pp. 8-17).

Siemens, G., & d Baker, R. S. (2012). Learning analytics and educational data mining: towards communication and collaboration. In*Proceedings of the 2nd international conference on learning analytics and knowledge* (pp. 252-254). ACM.

Strijbos, J. W., & Fischer, F. (2007). Methodological challenges for collaborative learning research. *Learning and Instruction*, *17*(4), 389-393.

Tair, M. M. A., & El-Halees, A. M. (2012). Mining Educational Data to Improve Students' Performance: A Case Study. *International Journal of Information*, *2*(2).

Thomas, E. H., & Galambos, N. (2004). What satisfies students? Mining student-opinion data with regression and decision tree analysis. *Research in Higher Education*, *45*(3), 251-269.

Vanneschi, L., & Poli, R. (2012). Genetic Programming—Introduction, Applications, Theory and Open Issues. In *Handbook of Natural Computing* (pp. 709-739). Springer Berlin Heidelberg.

Ventura, S., Romero, C., Zafra, A., Delgado, J. A., & Hervás, C. (2008). JCLEC: a Java framework for evolutionary computation. *Soft Computing*,*12*(4), 381-392.

Vygotskiĭ, L. L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard university press.

Wenger, E. (1998). *Communities of practice: Learning, meaning, and identity*. Cambridge university press.

Wolff, A., Zdrahal, Z., Nikolov, A., & Pantucek, M. (2013). Improving retention: predicting at-risk students by analysing clicking behaviour in a virtual learning environment. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge* (pp. 145-149). ACM.

Xing, W.L. Wadholm, B. & Goggins, S. (2014). Learning Analytics in CSCL with a Focus on Assessment: An Exploratory Study of Activity Theory-Informed Cluster Analysis. In *Proceedings of the Fourth International Conference on Learning Analytics and Knowledge*. ACM.

Xing, W.L. Wadholm, B. & Goggins, S. (2014). Assessment Analytics in CSCL:Activity Theory based Method. In *Proceedings of the International Conference on Learning Sciences'14,*Boulder, Colorado,USA. .

Xu, C., Wang, W., & Liu, P. (2013). A Genetic Programming Model for Real-Time Crash Prediction on Freeways. *Intelligent Transportation System, IEEE Transactions on*, 14(2), 574 - 586

Zafra, A., & Ventura, S. (2009). Predicting student grades in learning management systems with multiple instance genetic programming. *EDM*, *9*, 309-319.

Zhang, Y., & Bhattacharyya, S. (2004). Genetic programming in classifying large-scale data: an ensemble method. *Information Sciences*, *163*(1), 85-101.