Building (Timely) Bridges between Learning Analytics, Educational Data Mining and Core Learning Sciences Perspectives

Ido Roll, University of British Columbia, 6224 Agricultural Road, Vancouver, BC V6T 1Z1, ido@phas.ubc.ca
Vincent Aleven, Kenneth R. Koedinger, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213
{aleven, koedinger}@cs.cmu.edu
Matthew Berland, University of Texas at San Antonio, 1 UTSA Circle San Antonio, TX 78249, matthew@berland.org
Taylor Martin, Tom Benton, Carmen Petrick, University of Texas at Austin, 1 University Station, Austin, Texas, 78712
{taylormartin, tbenton, cpetrick}@mail.utexas.edu
Arnon Hershkovitz, Michael Wixon, Ryan Baker, Janice Gobert, Michael Sao Pedro
100 Institute Rd, Worcester Polytechnic Institute, Worcester, MA
{arnonh, mwixon, rsbaker, jgobert, mikesp}@wpi.edu
Bruce Sherin, Northwestern University, 2120 Campus Drive, Evanston, IL, bsherin@northwestern.edu
Paulo Blikstein, Marcelo Worsley, 520 Galvez Mall, Stanford, CA, 94305
{paulob, marcelo.worsley}@stanford.edu,
Paulo Blikstein, Organizer and Chair
Roy Pea, Stanford University, Discussant

Abstract: Despite the exponential growth of the research on Learning Analytics (LA) and Educational Data Mining (EDM) over the last few years, the work has been still distant from the core Learning Sciences methods, theoretical constructs, and literature. At the same time, over the last 15 years, Learning Sciences as a field has been quite innovative, eclectic, and effective in incorporating new methodological stances, such as micro-genetic methods, micro-ethnographies, and design-based research. It seems that the time has come to build sound connections between these traditions. The goal of this symposium is to bring together researchers coming from different academic perspectives, to explore and examine common LA/EDM methodological and theoretical threads with wide applicability within the Learning Sciences. The papers presented explore text mining in clinical interviews, moment-by-moment learning curves and traces, data mining of programming logs, and cognitive tutors, representing the main perspectives and methodological approaches in the field.

Overall focus of the symposium
Despite the exponential growth of the research on Learning Analytics (LA) and Educational Data Mining (EDM) (Baker & Yacef, 2009) over the next few years, the work has been still distant from the core Learning Sciences methods, theoretical constructs, and literature. At the same time, over the last 20 years, the Learning Sciences as a field has been quite innovative, eclectic, and effective in incorporating new methodological stances, such as micro-genetic methods (Siegler & Crowley, 1991), micro-ethnographies (Nemirovsky, 2011), and design-based research (Cobb, Confrey, diSessa, Lehrer, & Schauble, 2003; Confrey, 2005; Edelson, 2002).

However, in the last years, as computational technologies became truly ubiquitous, the possibilities of data collection opened up incredible new opportunities, such as the capture of students’ interaction with computer software, websites, as well as gesture, speech, position, mood, etc. There is clear interest in these techniques in the Learning Sciences for various reasons—for one, those data seem like a somewhat natural complement to micro-genetic methods, since they allow for an even greater sampling rate, and to design-based research, since they are accepting of unstructured learning activities and rapid-prototyping. Second, many learning scientists themselves have a computer science background, so they are familiar with computational modeling tools.

However, by its very nature, data-mining and machine learning tend to be agnostic of mechanism—in many fields of application (outside of education), gargantuan datasets and fast computers provide researchers with answers oftentimes much more advanced than the theoretical or explanatory models they have—what matters are correct predictions. This approach, however, is problematic in education, where understanding the causal chain is crucial for prescribing sound novel policies or educational designs.

Thus, we are at a crossroad in the history of both fields, one in search of new methods, and another searching for sound theoretical models to explain its findings. The question posed for the LS community is—
how to make a productive connection and avoid ‘big data’ from drifting towards a-theoretical, data-oriented work? The goal of this symposium is answer that questions by bringing together researchers coming from different academic perspectives, to explore and examine common LA/EDM methodological and theoretical threads with wide applicability within the Learning Sciences. The papers presented explore text mining in clinical interviews, data mining of programming logs, cognitive tutors, moment-by-moment learning traces, coming from scholars from different traditions but with a strong presence in the LS community.

Presentations

Hershkovitz et al. examine “disengaged behavior” and how to identify it using an automated detector. They will also relate that data with data from the Patterns of Adaptive Learning Scale and try to find out if different behavioral and epistemological issues relate to disengaged behavior. Overall, they show that students characterized by mastery or performance goal orientation have (on average) double the probability of carelessness as compared to students characterized by low scores for these goal orientations. Berland et al. present a study with students in which they use visual programming to create virtual robots. Patterns of collaboration affected students’ success at programming – students who shared complex code created better programs immediately afterwards. The study suggests that sharing is beneficial in the short term, but over the entire implementation, the average sharing measure was not correlated with code quality. Blikstein et al. paper follows a group of undergraduate students learning computer programming, as snapshots of their code are stored in a central server. Students fill a questionnaire about programming style and previous experience. The researchers use machine learning techniques to find patterns in how less and experienced learners progressed towards expertise. Roll et al. demonstrate the potential of using students’ moment-by-moment traces to offer domain-level support in scientific inquiry tasks, suggesting that adapting the task can be a productive mean for giving support in a constructivist environments. Log and assessment data from an evaluation of the Invention Lab in six grade nine classes (N = 92) was used to validate the modeling approach of the lab. Roll suggests that students’ trace data can be used to offer adaptive, individualized support in a manner that does not reduce critical elements of inquiry learning. Sherin’s paper examines techniques from statistical natural language processing (SNLP) and their use to analyze data produced by clinical interviews with middle-school students about scientific topics. In particular, he introduces Latent Dirichlet Allocation as an alternative to the more popular Latest Semantic Analysis, and uses it to segment transcripts of student’s conceptions in open-ended interviews about the seasons, verifying the algorithm against human coders.

Using Dynamic Time Warping and Cluster Analysis to Analyze the Learning of Computer Programming

Paulo Blikstein, Marcelo Worsley, Stanford University

Educational data mining and Learning Analytics (EDM; Amershi & Conati, 2009; Baker, Corbett, Koedinger, & Wagner, 2004) has grown rapidly over the last years, profiting from the vast availability of logfile data from e-learning systems, instrumented computer applications (Blikstein, 2009, 2011, Berland, 2008), computer vision systems, and web logs. The majority of the work focuses on standardized tasks delivered by cognitive tutors or clickstreams of online courses (Baker & Yacef, 2009). However, there is increasing interest in promoting types of learning activities and outcomes that are non-standardized, engaging students in constructionist (Papert, 1980) open-ended tasks such as designing robots or programming computers. In this work, we present a machine-learning-based framework to predict the performance and level of expertise of students based on their coding style. We will show results from a study in which we capture and analyze logs of undergraduate students (n=150) learning to program in an introductory Java course. In addition to logfiles, we also had students fill a questionnaire about their programming style, motivation, and previous experience.

As students learn to write program, they develop their own distinctive style of coding (top-down vs. bottom-up, “planners” vs. “tinkerers,” Turkle & Papert, 1991). The capture of these styles and their evolution during a computer science course could point to cognitive changes as well, and point to optimal points of intervention. Therefore, we analyze code fragments written by students for various assignments, based on fundamental concepts such as recursion, functional programming, object-oriented programming, error-handling, looping, etc. The code fragments committed to a global repository, giving us access to the each incremental time-stamped step in the process. First, we calculate differences between each successive commits (or “diff” statistics) by a student, in terms of the number of lines and characters added/deleted/reordered/modified, control-flow blocks added/deleted, and comments added. This gives us an approximate idea of the increments made by the student in each successive step and the time interval between steps. The individual code fragments are separately compiled and executed to capture their runtime errors and compiler errors. This results in several time-series of various diff statistics at each commit step, along-with the compiler or run-time errors. Using Dynamic Time Warping to calculate the distance between the time-series, we hierarchically cluster the time-series data of students, and use silhouette values to decide the cutoff for the number of clusters. We then analyze
the clusters for certain underlying “coding trends” which all students in that cluster share. The overall idea is to first identify the features influencing the coding style of a student, and then use them as features to group students into clusters. Finally, we cross those results with self-reported questionnaires on style, motivation, and experience. Preliminary results show distinctive changes in the style and learning pattern for self-declared “top-down” and “bottom-up” programmers, as well as connections between the evolution of coding styles and expertise.

AMOEBA: Mining how students learn to program together
Matthew Berland, University of Texas at San Antonio
Taylor Martin, Tom Benton, Carmen Petrick, University of Texas at Austin

This work details findings from AMOEBA, a new system to describe from a large dataset how students learn to program collaboratively. Work such as Hancock (2003) and Guzdial (2003) have shown key processes in how students learn to program through collaboration, but human observation is limited in the scale at which process data can be collected, the time frame over which it can be considered. Based on a single 90-minute session with students ranging from beginning to more experienced programmers, we were able examine student programming at a fine grain to produce a meaningful description of how students collectively learned to program, where similarities arose, and how students’ programs differed.

To explore student patterns of programming, we built the IPRO programming environment to collect those fine-grained data. The IPRO programming language was designed to foster quick programming on the fly (“tinkering”); it utilizes a restricted library of sensors, logical operators, mathematical operators, and actions. IPRO programs are immediately executable, and no syntax errors are possible in the environment. We designed IPRO with these features so that: student programs can be easily categorized and analyzed; it is easy to start programming with no background or prior knowledge; and students can share code easily. Berland (2008) and Hancock (2003) both describe how features such as reusable code and easy testability engage novices by allowing them to learn as they make rewarding progress on their programs.

We deployed the system to 53 students across 16 teams (of 3-4 virtual robots each) on the first day of a girls’ programming summer camp. Each student made between 100-500 edits to her programs over the course of 90 minutes. The new programs produced by each edit were automatically evaluated by the system in terms of how successful that program was at scoring goals. Metrics were created to evaluate similarity of any two edits as well as uniqueness of a program at a particular time. To investigate these relationships, we built AMOEBA.

AMOEBA explores the log data by visualizing patterns of similarity across programs. AMOEBA tracks the similarity of all programs of each of the 53 students on 16 teams over the course of 90 minutes. In short, our similarity metric reports how many subcomponents of the program are found to be in common with all of the other students’ programs in existence at that moment during the class. Similarity is scored by the number of components in common times the inverse frequency of that subcomponent in the corpus as a whole (similar to tf-idf as per Salton, 1989). AMOEBA in Figure 1 (left), below, shows “connections” between students where a connection is any similarity between two programs that is unlikely to be random. Those connections are marked with the most unlikely subcomponent that the two students’ programs have in common. Note that the students are groups by team in the diagram, to highlight when similarity is more common on teams and when it is not. We also report, as per Figure 1 (right), a “running correlation” of similarity to quality.

Our results show that students routinely wrote remarkably similar code at the same time. As seen in Figure 2, the correlation between similarity and quality also decreased. That is, the benefit gained by writing similar program code to your peers (through sharing or through mutual understanding of a topic) decreased as the situation became more complex. By matching AMOEBA log data to video, we noted that most of the
Student Attributes and Carelessness in Science Microworld
Arnon Hershkovitz, Michael Wixon, Ryan Baker, Janice Gobert, Michael Sao Pedro, Worcester Polytechnic Institute

Student attributes – including student goals, attitudes, and beliefs – play a key role in their learning outcomes. These attributes might impact learning by creating different forms of disengagement. One disengaged behavior is carelessness, i.e., when a student fails to answer an item correctly despite possessing the required skills (Clements, 1982). We have operationalized carelessness using an automated detector of contextual slip, i.e., the probability that the student performed incorrectly at a specific time despite knowing the needed skill (Baker, Corbett, & Aleven, 2008). The notion of contextual slip matches previous carelessness definitions, but is easier to apply than previous operational definitions. Our detector uses a log-based machine-learned model, hence can be scaled without being overly time-consuming.

We study carelessness in demonstrating science inquiry skills (e.g., control for variable strategy) in the context of a phase change activity using microworlds. Participants are 148 eighth grade students, aged 12-14 years old, from a public middle school in Central Massachusetts. Before taking the activity, the participants took the Patterns of Adaptive Learning Scale (PALS) survey. All students’ fine-grained actions were logged and then analyzed at the “clip” level; a clip is a consecutive set of a student’s actions describing activity in its context. Carelessness detector was developed based on 73 features describing the clips (N=2114), and using REPTree, a regression tree classifier (resulted in a 6-fold cross-validation correlation of r=0.63).

Overall, mean carelessness was low with a value of 0.05 across clips (SD=0.16) and 0.06 (SD=0.05) across students (N=130). Students were then clustered based on their PALS measures, resulting in three clusters: Learning Goals, Performance Goals, and Lack of Goals. Surprisingly, it was found that mean carelessness in the Lack of Goals cluster was significantly lower from its mean in both Learning Goals and Performance Goals clusters.

Furthermore, we studied how carelessness is being manifested over three consecutive activities. Overall, carelessness increased over all trials (as students have more practice opportunities for the same skill), however there were important differences among the clusters: In the Learning Goal cluster mean carelessness does not significantly increase from activity 1 to 2, nor from activity 2 to 3, but it is significantly higher in activity 3 compared to activity 1; in the Performance Goal cluster mean carelessness significantly increases between activities 1 to 2, but does not significantly increase between activities 2 to 3; and in the Lack of Goals cluster there is no significant differences were found in mean carelessness for either pair of activities.

This work shows that students characterized by mastery or performance goal orientation have (on average) double the probability of carelessness as compared to students characterized by low scores for these goal orientations. It is possible that students with higher amounts of mastery or performance goals succeed in learning and correspondingly become more confident, and following that they tend to carelessness despite their goal orientation. Differences in manifestation of carelessness over consecutive trials demonstrate important individual differences which might imply on timely scaffolding for those students who need to be encouraged to be more focused during learning. The methodology used in this research demonstrated the strength of using student log files and data mining methods, as the current operationalization of carelessness might be relatively easily (under limitation of, e.g., transferability) large-scaled.

Computing student science conceptions with Latent Dirichlet Allocation
Bruce Sherin, Northwestern University

Computationally-based analytic methods are becoming part of the standard toolkit employed by researchers in the learning sciences, and education more broadly. But, as a field, we have only scratched the surface of what is possible. Some techniques, such as Latent Semantic Analysis, have seen extremely wide use (Landauer, Foltz, & Laham, 1998). But the popularity of these techniques might be accidental; they were employed, with great success, by a few researchers (e.g., Graesser, Wiemer-Hastings, & Wiemer-Hastings, 2000; Magliano, Wiemer-Hastings, Millis, Munoz, & McNamara, 2002; Wade-Stein & Kintsch, 2004). This initial success then led to their wide adoption.

But we should be aware that there are many other existing methods, that are sometimes more appropriate to the task at hand. Here I will discuss one such method Latent Dirichlet Allocation (LDA), that can
be applied to solve problems that have been solved by LSA (Blei, Ng, & Jordan, 2003). Although these two methods can be applied to some similar tasks – and although they both have the word “Latent” in their names – they are actually quite different. Furthermore, in some cases, LDA is more powerful and more appropriate. In the remainder of this summary, I will first explain the research context. Then I will briefly explain LDA and how it can be employed in this context.

Over the last few decades, there have been literally thousands of research studies devoted to the study of students’ alternative conceptions in science (Duit, 2009). In many cases this research proceeds as follows; (1) We interview some students about a target domain or phenomenon. These interviews are videotaped. (2) We transcribe the videos. (3) We somehow extract a set of student conceptions from the transcripts. This generally requires repeated reading of the transcript and viewing of the video. And it usually involves some type of hand coding of the transcripts.

What I would like to do is to employ learning analytic techniques to perform the tasks in step (3); I want to give transcripts to a computer algorithm, and have it extract student conceptions. In particular, the data I will discuss is drawn from a corpus of clinical interviews in which middle students were asked to explain the Earth’s seasons. Thus, I would like my learning analytic algorithms to extract students’ seasons-related conceptions from transcripts of these interviews.

There are two additional complications. First, I adopt a knowledge-in-pieces perspective (diSessa, 1993). This means that I do not assume that students have existing models of the seasons. Instead, in many cases, they construct models of the seasons from fragments of knowledge. Thus, I actually want my analysis to identify these fragments, rather than full-blown models. Second, I expect that the fragments that a given student draws on may change as an interview unfolds. For this reason, I will not want to analyze transcripts, viewed as a whole. Instead, I must break each transcript into segments and determine the knowledge fragments associated with each segment.

LDA is well-suited to this task. Although the computational algorithms that perform LDA are quite complicated, the underlying conception is straightforward. In LDA, a corpus of text is modeled by a set of topics. In my case, these topics are the fragments. Each of these topics/fragments is associated with a probability distribution over a set of words. If a particular fragment is active, then it will generate words according to this probability distribution. Finally, each segment of a transcript is modeled as a mixture of the fragments. This means that, within the segment, a particular fragment is chosen with a probability determined by this mixture. Then a word is generated according to the probability distribution associated with the fragment that was selected. This imaginary process iterates, once for each word, generating a text.

This leads to a computational problem that must be solved. We have to discover the topics/fragments, the probability distribution of terms associated with each fragment, and the mixture of fragments associated with each segment of text. This is a difficult computational problem, but it is one that has been solved for us (Blei, et al., 2003).

The bottom line, as I will describe in my talk, is that the application of LDA to the seasons corpus produces an analysis that is sensible, and that aligns extremely closely with the work of human analysts; it discovers similar fragments, and it also produces a sensible account of the fragments of knowledge that are drawn on as each interview unfolds. I will conclude with a comparison to a similar analyses I performed using LSA.

Automated Task Adaptation to Support Students’ inquiry Learning

Constructivist instructional activities hold the promise of helping students acquire transferable knowledge (Duffy & Cunningham, 1996; Tobias & Duffy, 2009). However, research often shows that students fail to acquire the desired learning goals when learning from constructivist instruction (Kirschner, Sweller, & Clark, 2006; Tobias & Duffy, 2009). The challenges of learning from constructivist instruction are most apparent in exploratory learning environments and scientific inquiry activities. In these activities students are expected to reveal an underlying model that governs the behavior of given data or simulation (de Jong & van Joolingen, 1998).

One source of difficulty in scientific inquiry tasks is the relative lack of explicit support at the domain level. Since students are expected to discover the deep structure of the domain and learn by a process of exploration, inquiry environments usually withhold domain knowledge. For example, the Invention Lab (Figure 2) asks students to invent a method for calculating spread based on given data (Roll, Aleven, & Koedinger, 2010). Students are not given direct instruction on how to use standard deviation (or other alternatives), as this will short-circuit the reasoning process. At the same time, in the absence of explicit guidance, students often default to solutions that are partial at best. For example, many students simply settle for range (that is, max minus min), to measure spread (Roll, Aleven, & Koedinger, 2011).
A common solution to the lack-of-support problem is often given in the form of process scaffolding or the use of cognitive tools. For example, students are often prompted to raise hypotheses and test them. However, it seems that process support alone, without complementary domain-level support, is insufficient for novice students (Mulder, Lazonder, & de Jong, 2009).

Two main challenges limit the ability to give domain-level support during scientific inquiry tasks. First, as suggested above, explicit support is likely to short-circuit the desired sense-making processes. Second, given the high agency that students have in scientific inquiry tasks, students often explore the domain in different directions. Thus, support cannot be “one size fits all”, and canned responses cannot address many of the situations in which students are in need for support.

In this talk we demonstrate the potential of using students’ moment-by-moment traces to offer domain-level support in scientific inquiry tasks. First, we suggest that adapting the task can be a productive mean for giving support in a constructivist environment. By adapting the task to students’ demonstrated difficulties one can create tasks that are within students’ ZPD, thus addressing the need for explicit support in overly-complex situations. Second, adapting the task does not reduce the autonomy and agency of the learner, thus adhering to the constructivist principles of inquiry learning. Last, task adaptation does not short-circuit the reasoning behavior. Putting it more figuratively, while explicit support helps learners reach the goalpost, task adaptation brings the goalpost closer to the learner. Second, we suggest that the combination of constrained based modeling (Mitrovic, Koedinger, & Martin, 2003) and symbolic modeling (Anderson et al., 2004) is best suited for adapting the task to individual students.

Specifically, the Invention Lab analyzes students’ invented models in real time. Since no two models are identical, the Invention Lab looks for pre-defined desired features in these models. For example, does the model uses all the given data? (While range uses only the extreme data points, using all data points is essential to capture the true distribution of the data).

By identifying the shortcomings of students’ invented models, the Invention Lab creates new sets of data for students to analyze. The new data targets these knowledge gaps one by one. For example, students who “invent” range may be asked to compare two data sets with distinct distributions and a common range (e.g., \{1,4,5,5,8\} vs \{1,2,3,6,7,8\}). This iterative process of adapting available data to students’ invented methods is assumed to achieve two goals. First, it breaks down the problem and allows students to tackle the deep concepts of the domain one-by-one. Second, the iterative invention using adaptive data sets encourages students to integrate the different features of the data into a unified schema.

In addition, the invention lab uses its knowledge of students’ moment-by-moment actions to offer feedback on students general inquiry behaviors.

Log and assessment data from an evaluation of the Invention Lab in six grade nine classes (n = 92) was used to evaluate the modeling approach of the lab. Overall, this work suggests that students’ trace data can be used to offer adaptive, individualized support in a manner that does not reduce critical elements of inquiry learning.

References


Hancock, C. M. (2003). Real-time programming and the big ideas of computational literacy. MIT.


Acknowledgments
For “Automated Task Adaptation to Support Students’ inquiry Learning”: This work was supported by the Pittsburgh Science of Learning Center, which is supported by the National Science Foundation (#SBE-0836012), and by the University of British Columbia through the Carl Wieman Science Education Initiative.