2013

Debugging the Evidence Chain

Russell Almond, Yoon Jeon Kim, Valerie J. Shute and Matthew Ventura

Debugging the Evidence Chain

Russell G. Almond* Florida State University
Yoon Jeon Kim Florida State University
Valerie J. Shute Florida State University
Matthew Ventura Florida State University

Abstract

In Education (as in many other fields) it is common to create complex systems to assess the state of latent properties of individuals — the knowledge, skills, and abilities of the students. Such systems usually consist of several processes including (1) a context determination process which identifies (or creates) *tasks*—contexts in which evidence can be gathered,—(2) an evidence capture process which records the work product produced by the student interacting with the task, (3) an evidence identification process which captures observable outcome variables believed to have evidentiary value, and (4) an evidence accumulation system which integrates evidence across multiple tasks (contexts), which often can be implemented using a Bayesian network. In such systems, flaws may be present in the conceptualization, identification of requirements or implementation of any one of the processes. In later stages of development, bugs are usually associated with a particular task. Tasks which have exceptionally high or unexpectedly low information associated with their observable variables may be problematic and merit further investigation. This paper identifies individuals with unexpectedly high or low scores and uses weight-of-evidence balance sheets to identify problematic tasks for follow-up. We illustrate these techniques with work on the game *Newton's Playground*: an educational game designed to assess a student's understanding of qualitative physics.

Key words: Bayesian Networks, Model Construction, Mutual Information, Weight of Information, Debugging

1 Introduction

The primary goal of educational assessment is to draw inferences about the unobservable pattern of student knowledge, skills and abilities from a pattern of observed behaviors in recognized contexts. The reasoning chain of an assessment system has several links: (1) It must recognize that the student has entered a context where evidence can be gathered (often, this is done by providing the student with a problem that provides the assessment context). We call such a context a *task*, as frequently it is the task of solving the problem which provides the required evidence. (2) The relevant parts of the student's performance on that task, the student's *work product*, must be captured. (3) The work product is then distilled into a series of *observable outcome* variables. (4) These observable outcome variables are used to update beliefs about the latent proficiency variables which are the targets of interest.

Bayesian networks are well suited for the fourth link in the evidentiary chain. Often the network can be designed to have a favorable topology, where observable variables from different contexts are conditionally independent given the latent proficiency variables. In such cases, the Bayesian network can be partitioned into a student proficiency model—containing only the latent proficiency variables—and a series of evidence models (one for each task)—capturing the relationships between the proficiency and evidence models for a particular task (Almond & Mislevy, 1999).

When the assessment system does not perform as expected, there is still a model with hundreds of variables that must be debugged. Furthermore, the problem may not lie just in the Bayesian network, the last link of the evidentiary chain, but anywhere along that chain. By using various information metrics, the prob-

*Paper submitted to Big Data Meets Complex Models, Application Workshop at Uncertainty in Artificial Intelligence Conference 2013, Seattle, WA.

lem can be traced to the parts of the evidentiary chain associated with a particular tasks. In particular, if the anomalous behavior can be associated with a particular individual attempting a particular task, this can focus troubleshooting effort to places where it is likely to provide the most value.

This paper explores the use of information metrics in troubleshooting the assessment system embedded in the game *Newton's Playground*(*NP*; Section 2). Section 3 describes a generic four process architecture for an assessment system. In *NP* tasks correspond to game levels; Section 4 describes some information metrics used to identify problematic game levels. Section 5 describes some of the problem identified so far, and our future development and model refinement plans.

2 Newton's Playground

Shute, Ventura, Bauer, and Zapata-Rivera (2009) explores the idea that if an assessment system can be embedded in an activity that students find pleasurable (e.g., a digital game), and that the activity requires them exercise a skill that educators care about (e.g., knowledge of Newton's laws of motion), then by observing performance in that activity, educators can make unobtrusive assessment of the students ability which can be used to guide future instruction. *Newtons Playground* (Shute & Ventura, 2013) is a two-dimensional physics game, inspired by the commercial game *Crayon Physics Deluxe*. It is also designed to be an assessment of three different aspects of proficiency: qualitative physics (Ploetzner & VanLehn, 1997), persistence, and creativity. This paper focuses on assessment of qualitative physics proficiency.

2.1 Gameplay

NP is divided into a series of levels, where each level consists of a qualitative physics problem to solve. In each game level, the player is presented with a drawing containing both fixed and movable objects. The goal of the level is to move the ball to a balloon (the target), by drawing additional objects on the screen. Most objects (both drawn and preexisting) are subject to the laws of gravity (with the exception of some fixed background objects) and Newton's laws of motion. (The open source Box 2D (Catto, 2011) physics engine provides the physics simulation.)

Figure 1 shows the initial configuration of a typical level called Spider's Web. Figure 2 shows one possible solution in which the player has used a springboard (attached to the ledge with two pins—small round circles) to provide energy to propel the ball up to the balloon. Deleting the weight will cause the ball to strike

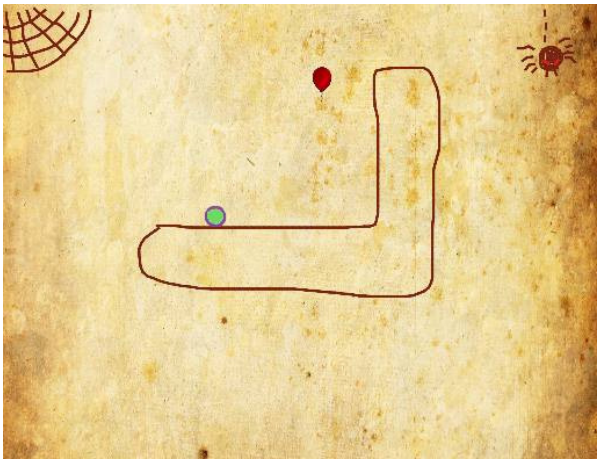

Figure 1: Starting Position for Spider Web Level

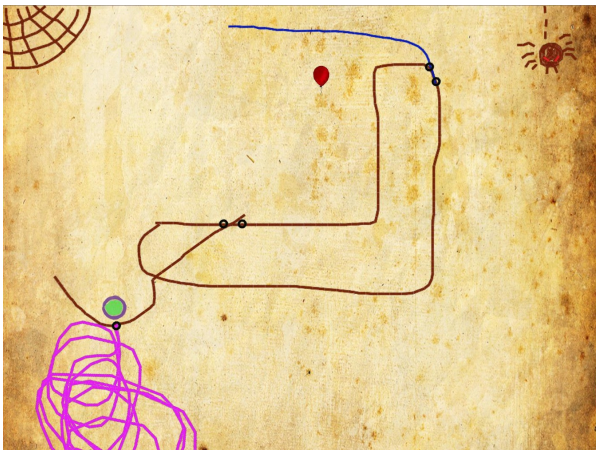

Figure 2: Spider Web Level with Springboard Solution.

ramp attached to the top of the wall which keeps the ball from flying over the target.

The focus of the current version has been on four *agents of motion* (simple machines): ramps, levers, springboards and pendulums. The game engine detects when one of those four agents was used as part of the solution. The game awards a trophy when the player solves a game level. Gold trophies are awarded if the solution is efficient (uses few drawn objects) and silver trophies are given as long as the goal is reached.

2.2 Proficiency and Evidence Models

The yellow nodes in Figure 3 show the student proficiency model for assessing a player's qualitative physics understanding. The highest level node, *Newton's Three Laws*, is the target of inference. It is divided into two components: one related to the application of those laws in linear motion, and one in angular

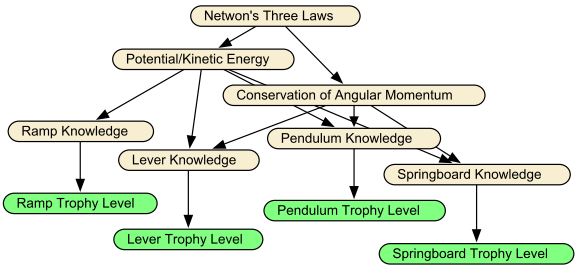

Figure 3: Physics Proficiency Model and Generic Observables

motion. The next layer has four nodes representing the four agents of motion. All of the nodes in the proficiency model had three levels: `High`, `Medium` and `Low`, and *expected a posterior* (EAP) scores could be calculated by assigning those levels a numerical value (3, 2, and 1, respectively) and taking the expectation.

The final layer of the model, shown in green represents the observable outcome variables from a generic level. These take on three possible values: `Gold`, `Silver` or `None`. The first two states are observed when the student solved using a particular agent. In that case, the observable for the correspond agent is set to the color of trophy received and the other observables are left unobserved. If the student attempts, but does not solve, the level the the observables corresponding to agents of motion the level designers thought would lead to solutions are set to `None`. The difficulty of a solution of each type, and the depth of physics understanding required, varies from level to level. So the green layer must be repeated for each game level. Version 1.0 (described in this paper) used 74 levels, so the complete Bayesian network had 303 nodes.

2.3 Field Test

In Fall 2012, a field trial was conducted using 169 8th grade students from a local middle school. The students were allowed to play the game for 4 45-minute class periods. The game engine kept complete logs of their game play. Students watched video demonstrations of how to create the four agents of motion in the game, and then were allowed to work through the game at their own pace. Game levels were grouped into playgrounds, with earlier playgrounds containing easier levels than the later playgrounds. Students were told that the player who got the most gold trophies would receive an extra reward.

One behavior which was often observed was the drawing of a large number of objects on the screen (often just under the ball to lift it higher), without a systematic plan for how to solve the level. Such "stacking" solutions had been observed in early playtests, and an object limit had been put in place to prevent it, but these "gaming" solutions were still observed during the field trial. Such solutions could lead to a silver trophy, but not to a gold trophy.

In addition to playing the game, a nine-item qualitative physics pretest and a matched nine-item posttest where given to the players. The pretest and posttest were not very stable measures of qualitative physics. On six different pendulum items (three from the pretest, three from the posttest) the students performed only slightly better than the guessing probabilities. The reliabilities (Cronbach's α Kolen & Brennan, 2004/1995) of the resulting six item tests were 0.5, and 0.4 for Forms A and B respectively.[1] This is a problem as physics understanding as shown on the posttest was the criterion measure, and these numbers form an effective upper bound on the correlation expected between the Bayesian network scores and the posttest.

We trained the Bayesian network using data from the field trial, and scored the field trial students. The correlation between the EAP scores from the highest level node and the physics pretest and posttest was around 0.1, which is not significantly different from zero at this sample size. Clearly there were problems in the assessment system that needed to be identified and addressed.

3 Four Process in the Evidence Chain

Because the correlation of the within game measure of Physics is so low, there must be a bug somewhere within the assessment system. A high level architecture of the assessment system will help define possible places. Figure 4, adapted from (Almond, Steinberg, & Mislevy, 2002), provides a generic architecture onto which assessment systems can be matched. It describes an assessment system that consists of four processes: *context determination*, *evidence capture*,[2] *evidence identification*, and *evidence accumulation*. In a general system, these can be human or machine processes, and several processes may be combined into a single piece of software, but all of the steps are present. Throughout, we will assume that the goal is to make inferences about the state of certain latent variables, which we will call the targets of inference.

[1] Half the students received Form A as a pretest and half as a posttest. This counterbalancing allowed the scores on the two forms to be equated.

[2] This process is called *presentation* in Almond et al. (2002). It is renamed here because it is the role of capturing the work product of the task is more important than the

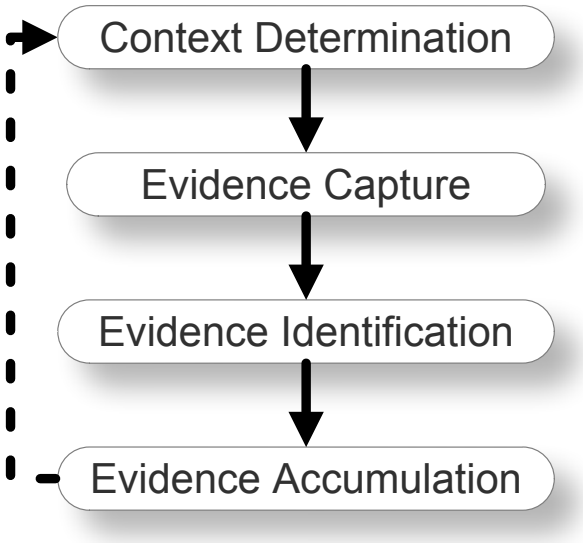

Figure 4: Four Process model of an Evidence Chain

Context determination is a process that identifies contexts in which evidence about the targets of interference can be gathered. In an educational assessment, these are often called *tasks*, as they represent problems a student must solve, or things a student must do. In a traditional assessment (like a college entrance exam), the test designers author tasks which are presented to the students forming the context for evaluating proficiency. When a student is engaged in free exploration with a simulator or game, the challenge in context determination is recognizing when current state of the simulator corresponds to a "task" that can be used to gather evidence (Mislevy, Behrens, DiCerbo, Frezzo, & West, 2012).

Other domains of application could use a mixture of engineered and natural contexts. For example, when trouble shooting a vehicle, the operators' reports of problems form natural contexts, while tests inside the garage are engineered contexts. Engineered contexts often provide stronger evidence than natural ones, because factors that might provide alternative explanations, and hence weaken the evidence for the targets of inference, can be controlled.

In *NP*, the contexts (tasks) are the game levels and they fall somewhere in between the natural and engineered range. Each of the game levels was designed by a member of the team, and each game level was designed to be solvable with a particular agent of motion (sometimes more than one). However, we had no control over which agent(s) the player would attempt to apply to the problem, and hence that part of the context was natural. Note that contexts are often described by variables (task model variables Almond, Kim, Velasquez, & Shute, 2012) that provide details about the context. In *NP*, the agents that the task designer thought provided reasonable solution paths (the `applicable agents`) and the task designer's estimate of difficulty were two such variables.

Evidence capture is a process that captures the raw data which will form the basis of the evidence. In educational assessment, we call that captured data the `work product` and note that this could come in a large variety of formats (e.g., video, audio, text, a log file of event traces). In *NP*, the evidence capture process was the game itself, and the work product consisted of a log file containing information about the player's interaction with the system (sufficient to replay the level), as well as additional information about the attempt (e.g., how long the player spent, how many objects were created and deleted, whether the player received a gold or silver trophy, etc.).

The *evidence identification* process takes the work product gathered by the evidence capture process and extracts certain key features: the *observable outcome variables*. One key difference of this process from the evidence accumulation process is that it always operates within a single context. The goal here is to reduce the complexity of the work product to a small, manageable number of variables. For example, a human rater (or natural language processing software) might rate an essay on several different traits. Those traits would be the observable outcome variables.

One design detail which is always tricky is figuring out how much processing of the work product to put into the evidence capture and how much is left for the evidence identification process. In *NP*, the evidence identification process was a collection of Perl scripts that extracted the observables from the log files. In some cases, it proved more convenient to implement the evidence identification rules in the game engine. In particular, it was important to identify if an object drawn by the player was a ramp, lever, pendulum or springboard. That was easier to do inside the game (i.e., evidence capture process) where the physics engine could be queried about the interactions of the objects. In other cases, it proved more convenient to filter the observables in the evidence accumulation process. For example, we did not want to penalize the player for failing to solve a level with a particular agent if the level was not designed to be solved with that agent. In this case, it turned out to be simpler to implement this on the Bayes net side (i.e., the evidence accumulation process), and the observable node corresponding to an agent would not be instantiated to `None` if the agent was not applicable for that level.

role of presenting the task.

The *Evidence accumulation* process is responsible for combining evidence about the targets of inference across multiple contexts. In *NP*, the evidence accumulation process consisted of a collection of Bayesian networks: a student proficiency model for each student, and a collection of evidence models for each game level. When it received a vector of observables for a particular student on a particular game level, it drew the appropriate evidence model from the library and attached it to that student's proficiency model. It then instantiated nodes in the evidence model corresponding to the observable values, and propagated the evidence into the proficiency model. The evidence model was then detached from the proficiency model which remained as a record of student proficiency. It could be queried at any time to provide a score for a student (Almond, Shute, Underwood, & Zapata-Rivera, 2009).

The dashed line in Figure 4 from the evidence accumulation process to the context determination process[3] is to indicate that in some situations the context determination might query the current beliefs about the targets of inference before selecting the next task (context). This produces a system that is adaptive (Shute, Hansen, & Almond, 2008). In *NP*, the player was free to choose the order for attempting the levels, hence this link was not used.

The four processes can be put together into a system that provides real-time inference or as a series of isolated steps. In version 1.0 of *NP*, only the evidence capture system (the game itself) was presented to the players in real-time. As the design of the other parts of the system was still undergoing refinement, it was simpler to implement them as separate post-processing steps. In a future version, these process will be integrated with the game so that players can get scores from the Bayes net as they are playing.

Developing each process requires three activities: *conceptualization*—identifying the key variables and work products and their relationships,—*requirement specifications*—writing down the rules by which values of the variables are determined,—and *implementation*—realizing those rules in code. A bug that causes the system to behave poorly can be related to a flaw in any one of those three activities, and can affect one or more of the four processes.

By the time the system was field tested, obvious bugs had been found and fixed. The remaining bugs only occur in particular particular game levels, and particular patterns of interaction with those levels. Once the levels in which bugs manifest and the patterns of usage which cause the bugs to manifest are identified, the problems can be addressed. This may entail adjust parameters for the Bayesian network fragment associated with that network, changing the level, replacing the level or making changes to the game engine, evidence identification scripts, or instructions to players.

[3]Almond et al. (2002) called this the activity selection process, to emphasize its adaptive nature.

4 Information Metrics as Debugging Tools

It is always the case that students interacting with an assessment system do so in ways that were unanticipated by the assessment designers. Information metrics provide a mechanism for flagging levels which behave in unexpected ways. In particular, we expect that a properly working game level will provide high information for the applicable agents (the ones that the designers targeted) and low information for the inapplicable agents. Extremely high information could also be an indication of overfitting the model to data.

Section 4.1 looks at the parameters of the conditional probability table as information metrics. Section 4.2 looks at the mutual information between the observable variable and its immediate parent in the model. Section 4.3 looks at tracing the score of specific individuals as they work through the game to identify problematic player/level combinations.

4.1 Parameters of the Conditional Probability Tables

Following Almond et al. (2001) and Almond (2010), we used models based on item response theory (IRT) to determine the values of the conditional probability tables. For each table, the effective ability parameter, $\tilde{\theta}$, is determined by the value of the parent variable (the values were selected based on equally spaced quantiles of a normal distribution: -0.97 for `Low`, 0 for `Medium`, and 0.97 for `High`). The model is based on estimates for two probabilities, the probability of receiving any trophy at all (using a specified agent), and the probability of receiving a gold trophy given that a trophy was received. These are expressed as logistic regressions on the effective theta value:

$$\begin{aligned}\Pr(\text{Any Trophy}|\text{Agent Ability}) \\ = \text{logit}^{-1}\, 1.7 a_S(\tilde{\theta} - b_S), \quad (1)\\ \Pr(\text{Gold Trophy}|\text{Any Trophy}, \text{Agent Ability}) \\ = \text{logit}^{-1}\, 1.7 a_G(\tilde{\theta} - b_G); \quad (2)\end{aligned}$$

where the 1.7 is a constant to match the logistic function to the normal probability curve. The two equations are combined to form the complete conditional probabilities using the generalized partial credit model (Muraki, 1992).

The silver and gold *discrimination* parameters, a_S and a_G, represent the slope of the IRT curve when $\theta = b$. They are measures of the strength of the association between the observable and the proficiency variable it measures. In high-stakes examinations, discriminations of around 1 are considered typical, and discriminations of less than 0.5 are considered low. We expect lower discriminations in game-based assessments as there may be other reasons (e.g., lack of persistence) that a player would fail to solve a game level. Still, when a game level is designed to target a player's understanding of a particular agent, very low discrimination is a sign that it is not working. High discriminations (above 2.0) are often a sign of difficulty in parameter estimation.

The silver and gold *difficulty* parameters, b_S and b_G, represent the ability level required to have a 50% chance of success. They have the opposite sign of a typical intercept parameter, and they should fall on a unit normal scale: tasks with difficulties below -3 should be solved by nearly all participants and those with difficulties above 3 should be solved by almost no participants.

The complete model had four parameters, two difficulty and two discrimination parameters, for each level/agent combination. One member of our level design team provided initial values for those parameters based on the design goals, applicable agents, and early pilot testing.

Correlations between the posttest scores and the Bayes net scores using the expert parameters were low, so we developed a method for estimating the parameters from the field trial data. First, the pretest and posttest were combined (as they were so short) and then separated into subscales based on agent of motion. As the scores were short, the augmented scoring procedure of Wainer et al. (2001) was used to shrink the estimates towards the average ability. Each subscale was split into `High`, `Medium` and `Low` categories with equal numbers of students in each. This provides a proxy for the unobservable agent abilities for each student.

We used the agent ability proxies and the observed trophies to calculate a table of trophies by ability for each level. The tables were rather sparse as many students did not attempt many levels, and and typically used only one agent for each level attempted. To overcome this sparseness, the conditional probability tables generated using the expert parameters were added to the observed data, and then a set of parameter (a_S, a_G, b_S, b_G) were found that maximized the likelihood of generated the combined prior + observed table using a gradient decent algorithm.

Looking for extremely high discrimination values immediately flagged some problems with this procedure. In particular, cases where only one of two students attempted a level with a particular agent, but were successful, could result in an extremely high discrimination. Increasing the weight placed on the prior when calculating the prior+observation table reduced the occurrences of this problem.

There were still some level/agent combinations with extremely high discrimination, but we noticed that they had extremely high difficulties as well. Looking at the conditional probability tables generated by these parameter values we noticed that they were nearly flat (in other words, the three points on the logistic curve corresponding to the possible parent levels were in one of the tails of the logistic distribution). Because the conditional probability table was flat, the high discrimination does not correspond to high information, so is not likely to overweight evidence from that game level. Consequently, flagging just high discrimination produced too many false positives, and additional screening was needed.

4.2 Mutual Information

The *mutual information* of two variables X and Y is defined as:

$$MI(X,Y) = \sum_{x,y} \Pr(x,y) \log \frac{\Pr(x,y)}{\Pr(x)\Pr(y)} \ . \quad (3)$$

Calculating the mutual information for all of the level/agent combinations yielded a maximum mutual information of 0.09, with most mutual information values below 0.01. Figure 5 shows the mutual information for both applicable agent/level combinations and inapplicable ones.

Table 1 shows the conditional probability table parameters and mutual information for a few selected levels, looking at just the *Lever Trophy* observables. The particular levels were flagged because they had either high discrimination, high (in absolute value) difficulty or high mutual information. The game level "Stairs" is an example of a problem: it has an extremely high discrimination for silver trophies and an extremely high difficulty as well. Furthermore, the mutual information is toward the high end of the range. The level "Swamp People" is also a problem, it has a high gold discrimination as well as a high mutual information. Furthermore, lever was not thought to be a common way of solving the problem by the game designers.

It is important to use the mutual information as a screening criteria to eliminate false positives. The game level "Smiley" is an example of a false positive. Although the silver discrimination and difficulty are high, the mutual information is below 0.001, so

Table 1: Parameters and mutual information for selected lever observables.

	applicable	a_S	b_S	a_G	b_G	MI
Diving Board World	TRUE	0.897	5.036	0.024	1.974	0.000
Smiley	TRUE	3.368	7.255	0.002	1.479	0.000
St. Augustine	TRUE	0.897	5.036	0.024	1.974	0.000
Stairs	TRUE	11.084	10.756	0.000	0.774	0.064
Swamp People	FALSE	0.116	4.782	2.431	3.689	0.033
Ballistic Pendulum	FALSE	0.897	5.036	0.024	1.974	0.000

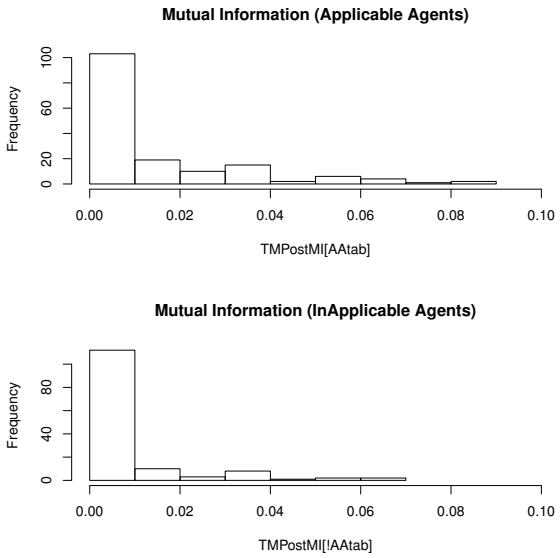

Figure 5: Histograms of Mutual Agent Distributions

the extreme parameter values are likely not causing a problem. This level could be a problem for a different reason: lever was judged to be an applicable solution agent, but the mutual information is low. The unexpectedly week evidentiary value of this level should be investigated.

4.3 Evidence Balance Sheets

The *weight of evidence*(Good, 1985) a piece of evidence E provides for a hypothesis H versus its negation $\overline{H}$ is:

$$W(H{:}E) = \log \frac{\Pr(E|H)}{\Pr(E|\overline{H})} = \log \frac{\Pr(H|E)}{\Pr(\overline{H}|E)} - \log \frac{\Pr(H)}{\Pr(\overline{H})} \ . \tag{4}$$

If the evidence arrives in multiple pieces, E_1 and E_2 (e.g., the evidence from each game level), the *conditional weight of evidence*:

$$W(H{:}E_2|E_1) = \log \frac{\Pr(E_2|H, E_1)}{\Pr(E_2|\overline{H}, E_1)} \ . \tag{5}$$

These sum in much the way that one would expect:

$$W(H{:}E_1, E_2) = W(H{:}E_1) + W(H{:}E_2|E_1) \ . \tag{6}$$

Madigan, Mosurski, and Almond (1997) suggest a *weight of evidence balance sheet*: simple graphical display for the conditional weights of evidence. Figure 7 shows an example. The leftmost column gives the game levels in the order that they were scored, as well as the agent and trophy that was received. The central column gives the conditional probability for the target node, *Newton's Three Laws* at various points in the scoring sequence. The third column gives the weight of evidence the most recent level provides for the hypothesis that the target node is at least at the level of `Medium`.

Constructing a balance sheet requires selecting a particular student. Interesting students can be identified by looking for outliers in the regression of the posttest (or pretest) scores on the Bayesian network EAP scores (Figure 6). Certain students were identified in this plot. Student S259 got no pretest items right (although that student got about 4 posttest items right, which was a good score), and had an EAP score of 2.3 (which is in the medium category for physics understanding).

Figure 7 shows the pattern of scores for this student. Early levels tend to have higher weights of evidence than later levels. Note that somewhere towards the middle of the sequence there are two huge spike in the weight of evidence. These correspond to the levels "jar of coins" and "Jurassic park"; both had weights of evidence of over 75. Table 2 presents the same information in a tabular fashion. Here the information is screened so that only levels with high weights of evidence are shown.

To systematically investigate what causes the spikes in the weight of evidence, we reviewed replay files of the identified students. For example, S259 mostly used solutions that "game the system"(e.g., crashing the system by drawing random large objects) and rarely tried to use applicable agents. Thus when he somehow managed to use an applicable agent and earned a trophy, the weight of evidence jumped.

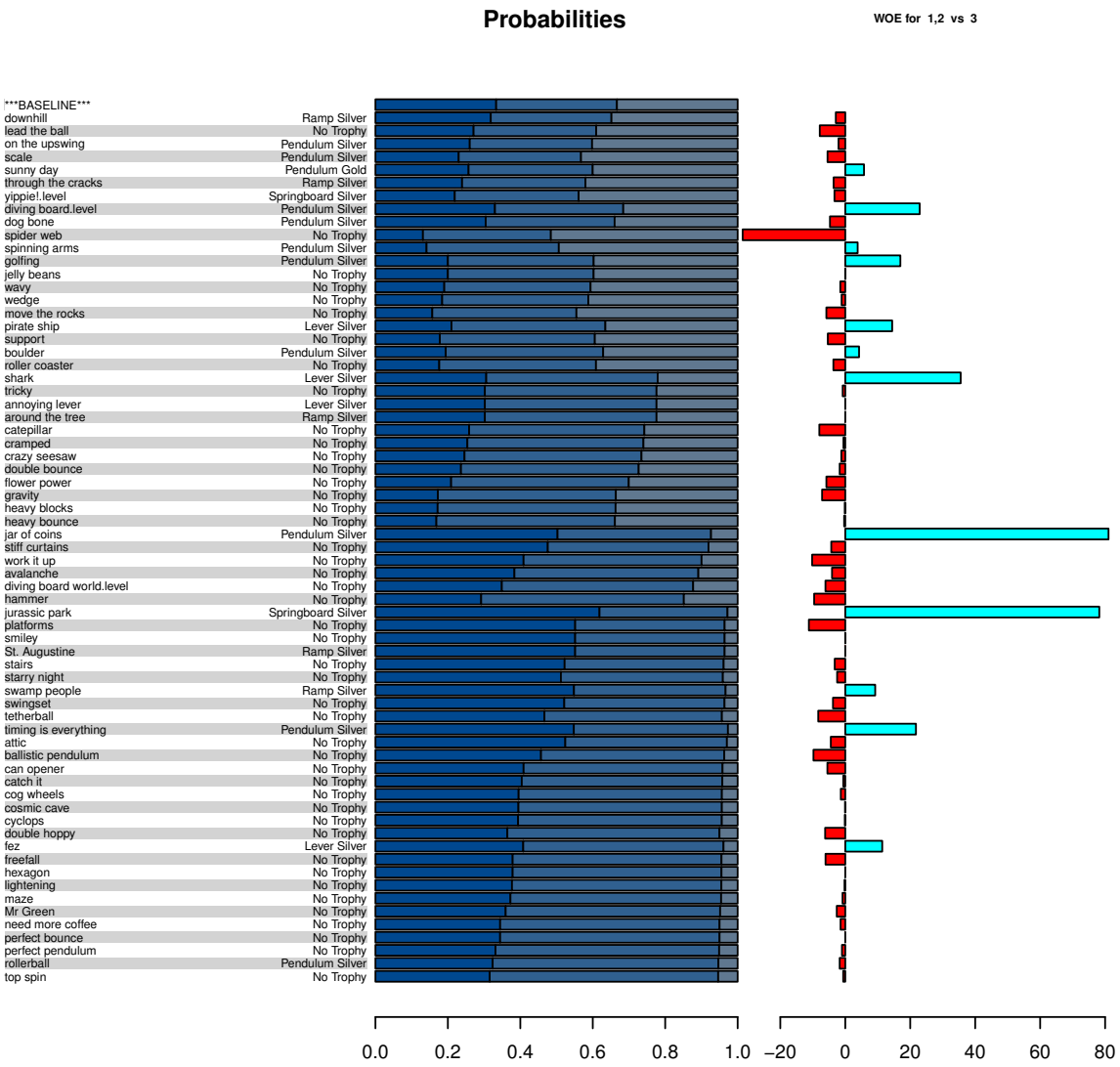

Figure 7: Weight of Evidence Balance Sheet for Student S259

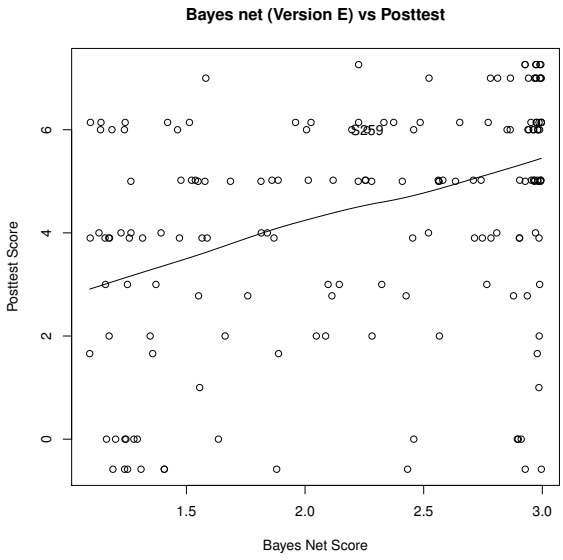

Figure 6: Scatterplot of Postttest versus Bayes net scores.

Table 2: Levels with high weights of evidence for Student S259

Level	WOE
lead the ball	-7.84
diving board	22.92
spider web	-32.59
golfing	16.97
pirate ship	14.45
shark	35.54
caterpillar	-7.99
jar of coins	80.04
work it up	-10.2
hammer	-9.58
Jurassic park	78.2
platforms	-11.21
swamp people	9.22
tether ball	-8.32
timing is everything	21.77
ballistic pendulum	-9.8
fez	11.38

For the case of "jar of coins", it is one of the levels that already has an applicable agent built in the level as an incomplete form (i.e., pendulum for this level), and all the player needs to do is to make the built-in agent work by completing it (e.g., add more mass to the pendulum bob). The review of his replay files revealed that he exploited the system again for jar of coin, but the system recognized his solution as an applicable due to the built-in agent. This finding should lead to one or more follow-up actions: (a) decrease discrimination for pendulum in the CPT of jar of coins, (b) revise the level to make it harder to "game" the system, and/or (c) replace the level with one that forces the player to directly draw the agent. We chose the third option for the next version of *Newton's Playground*.

5 Lessons Learned and Future Work

The work on constructing the assessment system for Newton's Playground is ongoing. Using these information metrics helped us identify problems in both the code and level design. For example, one case of unexpectedly low discrimination led to the discovery of a bug in the code that built the observed tables from the data (the labels of the `High` and `Low` categories were swapped and the observation table was built upside down). Unexpected high and low information also forced the designers to take a closer look at which agents students were actually using to solve the problems leading to a revision in the agent tables. Finally, viewing replays led us to identify places where the agent identification system misidentified the agent used to solve the problem. This led to improved values for the observable outcomes.

Correcting these problems lead to a definite improvement in the correlation between the Bayes net score and the pretest and posttest. With the revised networks and evidence identification code, the correlation with the pretest is 0.40 and with the posttest is 0.36, a definite improvement (and close to the limit of the accuracy available given the lack of reliability of the pretest and posttest).

We have also identified some conceptual errors that we are still working to address. In particular, a large number of the students (e.g., S259) engaged in off-track "gaming" behaviors, often earning silver trophies in the process. It is clear that the Bayesian network is lacking nodes related to that kind of behavior. Also, we need a better system for detecting that kind of behavior. These are being implemented in Version 2.0 of *Newton's Playground*.

References

Almond, R. G. (2010). 'I can name that Bayesian network in two matrixes'. *International Journal of Approximate Reasoning*, *51*, 167–178. Retrieved from `http://dx.doi.org/10.1016/j.ijar.2009.04.005`

Almond, R. G., DiBello, L., Jenkins, F., Mislevy, R. J., Senturk, D., Steinberg, L. S., et al. (2001). Models for conditional probability tables in educational assessment. In T. Jaakkola & T. Richardson (Eds.), *Artificial intelligence and statistics*

2001 (p. 137-143). Morgan Kaufmann.

Almond, R. G., Kim, Y. J., Velasquez, G., & Shute, V. J. (2012, July). *How task features impact evidence from assessments embedded in simulations and games.* Lincoln, NE. (Paper presented at the International Meeting of the Psychometric Society (IMPS))

Almond, R. G., & Mislevy, R. J. (1999). Graphical models and computerized adaptive testing. *Applied Psychological Measurement*, *23*, 223-238.

Almond, R. G., Shute, V. J., Underwood, J. S., & Zapata-Rivera, J.-D. (2009). Bayesian networks: A teacher's view. *International Journal of Approximate Reasoning*, *50*, 450–460.

Almond, R. G., Steinberg, L. S., & Mislevy, R. J. (2002). Enhancing the design and delivery of assessment systems: A four-process architecture. *Journal of Technology , Learning, and Assessment*, *1*, (online). Retrieved from `http://www.jtla.org/`

Catto, E. (2011). Box2D v2.2.0 user manual [Computer software manual]. Retrieved from `http://box2d.org/` (Downloaded July 25, 2012 from)

Good, I. J. (1985). Weight of evidence: A brief survey. In J. Bernardo, M. DeGroot, D. Lindley, & A. Smith (Eds.), *Bayesian statistics 2* (p. 249-269). North Holland.

Kolen, M. J., & Brennan, R. L. (2004/1995). *Test equating, scaling, and linking: Methods and practices* (2nd ed.). Springer-Verlag.

Madigan, D., Mosurski, K., & Almond, R. G. (1997). Graphical explanation in belief networks. *Journal of Computational Graphics and Statistics*, *6*(2), 160-181. Retrieved from `http://www.amstat.org/publications/jcgs/index.cfm?fuseaction=madiganjun`

Mislevy, R. J., Behrens, J. T., DiCerbo, K. E., Frezzo, D. C., & West, P. (2012). Three things game designers need to know about assessment. In D. Ifenthaler, D. Eseryel, & X. Ge (Eds.), *Assessment in game-based learning: Foundations, innovations, and perspectives* (pp. 59–81). Springer.

Mislevy, R. J., Steinberg, L. S., & Almond, R. G. (2003). On the structure of educational assessment (with discussion). *Measurement: Interdisciplinary Research and Perspective*, *1*(1), 3-62.

Muraki, E. (1992). A generalized partial credit model: Application of an em algorithm. *Applied Psychological Measurement*, *16*, 159–176.

Ploetzner, R., & VanLehn, K. (1997). The acquisition of informal physics knowledge during formal physics training. *Cognition and Instruction*, *15*(2), 169–205.

Shute, V. J., Hansen, E. G., & Almond, R. G. (2008). You can't fatten a hog by weighing it - or can you? Evaluating an assessment for learning system called ACED. *International Journal of Artificial Intelligence in Education*, *18*(4), 289–316. Retrieved from `http://www.ijaied.org/iaied/ijaied/abstract/Vol_18/Shute08.html`

Shute, V. J., & Ventura, M. (2013). *Stealth assessment in digital games.* MIT series.

Shute, V. J., Ventura, M., Bauer, M. I., & Zapata-Rivera, D. (2009). Melding the power of serious games and embedded assessment to monitor and foster learning: Flow and grow. In U. Ritterfeld, M. J. Cody, & P. Vorderer (Eds.), *Serious games: Mechanisms and effects* (pp. 295–321). Routledge, Taylor and Francis.

Wainer, H., Veva, J. L., Camacho, F., Reeve III, B. B., Rosa, K., Nelson, L., et al. (2001). Augmented scores — "borrowing strength" to compute scores based on a small number of items. In D. Thissen & H. Wainer (Eds.), *Test scoring* (pp. 343–388). Lawrence Erlbaum Associates.

Acknowledgments

Many aspects of the *Newton's Playground* examples are based on work of the *Newton's Playground* team, Val Shute, P.I. In addition to the authors, the team includes Matthew Small, Don Franceschetti, Lubin Wang, and Weinan Zhao. Pete Stafford assisted with the data analysis. Work on *Newton's Playground* and this paper was supported by the Bill & Melinda Gates Foundation U.S. Programs Grant Number #0PP1035331, *Games as Learning/Assessment: Stealth Assessment.* Any opinions expressed are solely those of the authors.