An Elicitation Tool for Conditional Probability Tables (CPT) for Physics Playground

Russell Almond, Seyfullah Tingir, Xi Lu, Chen Sun, Seyedahmad Rahimi
Florida State University
mailto: ralmond@fsu.edu, st13n@my.fsu.edu

Abstract

Building a Bayesian network is a cyclic procedure that includes identifying variables, constructing the graphical structure of the BN model using those variables, evaluating the conditional probability tables (CPTs) with the subject matter experts' inputs. This process repeats until content experts are satisfied that the Bayesian network is a valid representation for the domain. The current paper describes the steps of the procedure used to build a Bayesian network for an educational game called *Physics Playground,* emphasizing the elicitation tools. The process starts with initial consultation from physics experts to draw possible graphs for the relationship among the proficiency variables and tasks. Then, we calculate CPTs based on the Q-matrix. Next step describes how we are translating the R code into natural language so that the content experts can validate that the CPTs are sound.

Keywords: Bayesian Networks, Elicitation,

1. INTRODUCTION AND BACKGROUND

Bayesian networks are often used to represent student knowledge in educational games (De Klerk, Veldkamp, & and Eggen, 2015). As students play the game, their performance in each game level provides evidence about their knowledge, skills and abilities. Querying the Bayes net may provide accurate sets of information about students' ability that can be reported to a student or teacher (Almond et al., 2009) both in the form of assessment *for* learning or formative assessment, and assessment *of* learning or summative assessment (Shute & Rahimi, 2017). The Bayesian network can also be used to select an optimal sequence of activities (Shute, Hansen & Almond, 2008). This paper describes an effort to build the Bayesian network representing Physics competency in the game *Physics Playground*. In particular, it describes a natural language representation of the conditional probability tables used to elicit key parameters of the network and validate the network.

1.1. PHYSICS PLAYGROUND

The previous version of *Physics Playground* (*PP*; Shute & Ventura, 2013) was nonlinear: players could choose any level in the game to play or replay in any sequence. The goal of all 75 levels (or problems) in the previous version of *PP* was only to guide a green ball to hit a red balloon. Using the mouse, players drew coloured objects on the screen, which "come to life" as physical objects when the mouse button was released. These objects interacted with the game environment according to Newtonian mechanics and could move the ball towards the goal. When objects interacted within the game environment, they act as agents of force and motion—these are analogous to simple machines in formal physics: ramp, lever, pendulum, and springboard. The general proficiencies measured were force and motion, linear momentum, energy, and torque.

The current version of the *PP* still has game levels with similar mechanics mixed with new manipulation levels/tasks. In addition to the existing 75 levels, we are designing about 75 new game levels. The main difference between current and older version of the game is that the current version allows users to directly manipulate physics parameters such as gravity, mass, and air resistance and add external forces through blowers to solve the tasks (i.e., hit the balloon with the ball) Figure 1 shows an example: The player manipulates gravity and the mass of the ball using the sliders in the upper right to spin the bow tie and get the ball to drop to the center. The new levels expand physics content (i.e., Newtonian laws of motion; torque and conservation of momentum; and energy and dissipative forces).

Additionally, new kinds of feedback providing verbal and visual hints will be included in the game.

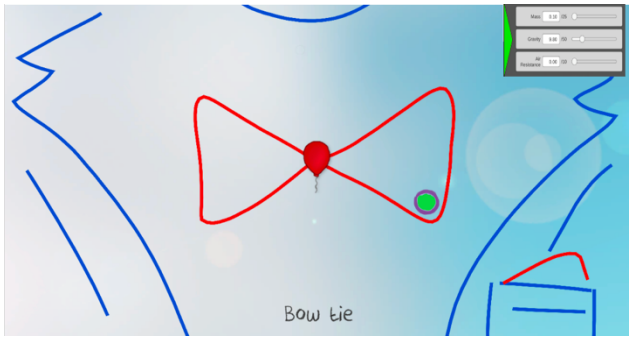

Figure 1: A new level in PP with sliders

The project has multiple teams with different responsibilities: the physics' pedagogy expert team, the level design and learning support team, the technical support and programming team, and the measurement team. The teams came together and built the competency model in an iterative process over the course of three to four months. Since each team focused on and had expertise in a different area, everyone needed to minimize the technical language when communicating with other teams. Therefore, the measurement team had to translate what they were doing into language that the rest of group could understand. This would allow the other teams—especially the physics team—confirm the important inputs.

2. METHOD

2.1. DEFINING THE VARIABLES

The first step was to have the experts identify the variables—competencies and sub-competencies—that we needed to turn them into variables and build a Bayesian network with. The experts created an "Evidence Statements" (ES) spreadsheet to show the competency hierarchy. Table 1 is a screenshot of the ongoing spreadsheet.

The ES spreadsheet has four columns: (1) Competency, (2) Sub-competency, (3) Explanation, and (4) Evidence. In the competency column, the experts listed five competencies (i.e., force and motion, linear momentum, energy, torque, and science and engineering practices) based on the Next Generation Science Standards for

Middle School Physics (<<Needs Ref, talk to Ginny>>). Each of the competencies contains 1-2 sub-competencies For instance, Force and motion includes Newton's three laws. Linear Momentum is divided into two sub-concepts: properties of momentum and conservation of momentum. Energy can be broken down to two things: (1) energy can transfer and (2) energy can dissipate. Torque includes properties of torque and equilibrium. The sub competency of science and engineering practices is the use of iterative design to solve a problem.

The third column, explanation, is where the experts put a succinct definition of each sub-competency. For example, the two hard and fast rules about Newton's 2nd law are: (1) mass and acceleration are inversely related; and (2) net force and acceleration are directly related. Properties of momentum, according to the experts, is directly related to mass, velocity, and is parallel to velocity. This column is especially useful for the learning support team when creating new game levels. The statements for each competency can be thought of as concrete learning objectives that we want a player to achieve in the levels of *PP*. A level designed to assess the properties of momentum sub-competency should be difficult to solve for players who lack a solid understanding of the relationships between momentum, mass, and velocity. In the evidence column, the experts briefly described what specific in-game behaviours (i.e., evidence) of a player can reveal each targeted competency. To assess whether a player understands static equilibrium in Newton's 1st law, the experts suggested that we should observe how a player applies or adjusts a force (e.g., nudge, blow, gravity, air resistance) to keep an object stationary. If we want to elicit evidence for conservation of momentum, we will need to see whether a player manipulates something to cause a collision to affect the motion of the other object.

2.2 ASSOCIATING COMPETENCIES WITH GAME LEVELS

One way that educational testing differs from other applications is that if a system lacks sufficient evidence to assess the state of a particular variable (in this case a competency variable), it is usually possible to add new tasks or items to the instrument to provide the additional measurement. In a game-based assessment, this translates to new levels for the game.

The *Q*-matrix is a tool that allows the designers to rapidly assess how much evidence is available for the various competencies in the game (Almond, 2010; Almond et al., 2015). Table 2 shows an example. The columns represent the (sub)competencies and the columns the tasks (game levels). A one in a particular cell means that the competency is useful for solving the task. Eventually, this will mean that an edge will appear in the final Bayesian network from the competency variable to one or more observable outcome variables from the task. For example, On the Upswing game level requires Newton's 1st Law as a primary necessary skill where Newton's second and third law skills are not observed.

Table 1: ES Spreadsheet

Competency	Sub-competency	Description	Evidence
Force and Motion	Newton's 1st Law	Static equilibrium (a=0 and v=0)	Player applies or adjusts a force (e.g., nudge, blow, gravity, air resistance) to keep an object stationary in at least one dimension.
Force and Motion	Newton's 2nd Law	Net force and acceleration are directly related	Player applies or adjusts a force acting on an object to cause it to accelerate at a desired rate.
Linear Momentum	Properties of momentum	Momentum is directly related to mass	Player adjusts the mass an object to affect the amount of momentum it transfers to a second object after the two collide.
Energy	Energy can transfer	Energy can transform from one type to another (e.g., GPE to KE)	Player changes parameters (e.g., mass, position, speed) to tranform more or less energy of one type to another (e.g., KE, GPE, EPE) of the same object.
Torque	Properties of torque	Force and torque are directly related.	Player adjusts the magnitude of a force to cause a corresponding change in the magnitude of a torque.
Science and Engineering Practices	Use iterative design to solve a problem	Solve a problem by making variations on previous strategies	Player makes successive adjustments of the same parameter to solve a level.

Table 2: Q-matrix designed by the learning support team

	Force and Motion		
	Newton's 1st Law	Newton's 2nd Law	Newton's 3rd Law
On the Upswing	1	0	0
Lead the Ball	0	0	0
Scale	1	0	0
Spider Web	0	0	0
On the Upswing	0	0	0

Eventually, the *Q*-matrix will be augmented to provide additional information used for building the conditional probability tables, but for now the focus is just the structural *Q*-matrix: the pattern of ones and zeros. In particular, counting the number of ones in a particular column gives a crude estimate of how much evidence is available for a certain competency. This allowed the design team to focus their efforts on the competencies for which there was the least evidence.

After the first draft of the ES spreadsheet was completed, one of the first tasks of the design team was to inventory the existing game levels and fill out the corresponding rows of the *Q*-matrix. The physics experts then reviewed the *Q*-matrix. In the process, both teams I identified issues related to the definitions of the competencies. This would result in a revision of the ES spreadsheet followed by updating the *Q*-matrix, providing a check on the revision of the ES spreadsheet.

The ES spreadsheet gives working definitions for the key variables needed for building the Bayesian network of *PP*. The *Q*-matrix provides a mechanism for checking the adequacy of the definitions. Both also serve as a job-aid for game level design.

2.3. BAYESIAN NETWORK CLASS STEP

The next step was for the measurement team to take the variables defined in the ES spreadsheet and produce the graphical structure for the Bayesian network. They used Netica (Norsys, 2012), but at this stage only as a drawing tool; no numbers have been added yet. They produced two candidate networks, shown in Figures 2 and 3. Both have the hierarchical structure, from the ES spreadsheet (competencies are coloured salmon and sub-competencies are coloured orange) and but the cross loadings are different.

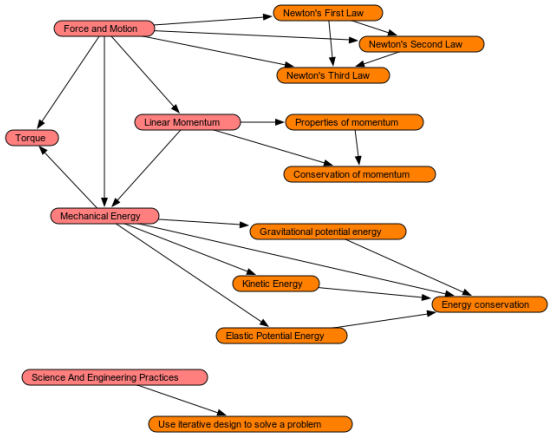

Figure 2: First candidate Bayesian net for PP

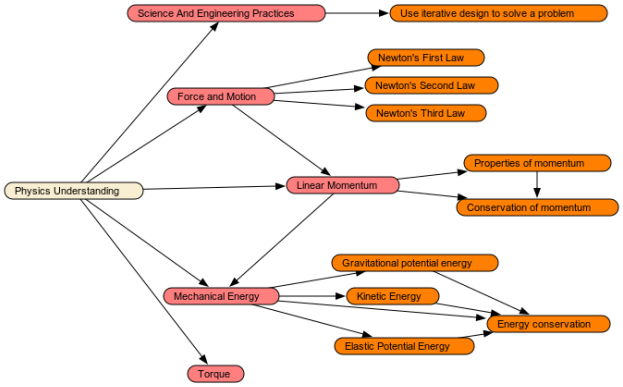

Figure 3: Second candidate Bayesian net for PP

Presenting more than one possible Bayes net to the physics experts (i.e., Figures 2 and 3) was important because were it forced the experts to think about the nature of the relationships and not just rubber stamp what the analysts did.

The differences between Figures 2 and 3 were chosen to facilitate discussions between the measurement and physics experts about key modelling assumptions. For example, in Figure 2 the science and engineering practices node is separated from the physics competencies, while in Figure 3 they are all connected through the overall Physics node. Also, Figure 3 has an overall Physics node while Figure 2 does not. Furthermore, in Figure 3, Newton's three laws are all conditionally independent given force and motion, but in Figure 2 they are dependent. Giving the experts two examples allowed the measurement team to both explain

what the decisions meant and get expert input about critical questions of conditional independence.

One issue that the measurement and physics teams needed to discuss was the type of relationship embodied in each link. Some links represented the hierarchical relationship among the concepts (Section 2.1). Others represented prerequisite knowledge: for example, usually students learn the properties of momentum before learning conservation of momentum. Still others represented concepts which are learned together. For example, Newton's first and second law are usually introduced at the same time. One other type of relationship arose that we did not expect. The physics experts pointed out that some concepts represented different ways of looking at the same problem. For example, Newton's third law (that forces are balanced) is related to the conservation of energy. This latter discovery posed problems for the design team who needed to design levels that provided evidence for one or the other of the two ways of looking at the same problem.

Another class of question represented in the Figures 2 and 3 related to the colour of the nodes. The higher level skills (directly connected to Physics in Figure 3) were coloured salmon, and the low level nodes were coloured orange. Some of the questions revolved around at which level of node the game would report scores to students. Other questions were about whether there were so many competencies in the model that they could not be assessed in a reasonable period of game play.

Figure 4 shows the result of several iterative revisions of the network. The main changes are as follows: (1) The experts liked the overall Physics node, but retained some of the cross-loadings from Figure 2. (2) Mechanical energy became simply energy, which covered only two sub-competencies—energy can transfer between types (potential and kinetic) and energy can dissipate due to forces (e.g., air resistance). (3) Two (orange) sub-competencies were added under torque: properties of torque and static equilibrium.

The network was finalized at a joint meeting of all of the project teams. As the competency model would be the focus of much of each team's work, it was important to makes sure that everyone had a chance to make input. Once the network structure was finalized, the measurement team could begin work on the conditional probability tables.

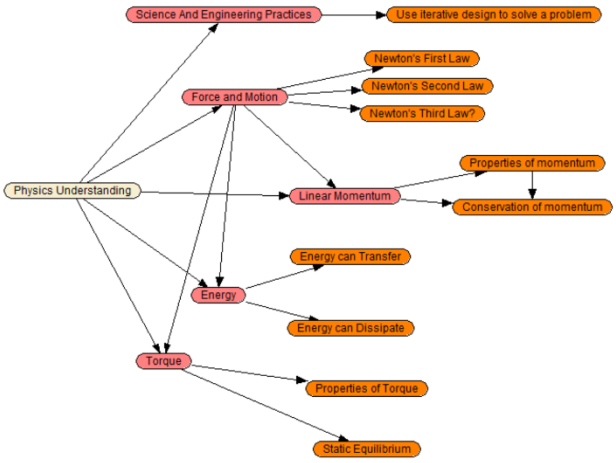

Figure 4: Final version of Bayesian net for PP

2.3 THE DIBELLO-NORMAL MODEL

For *PP*, the domain experts were physicists; so they had some prior experience with probability. They understood the concept of probability and understood what a conditional probability table was when that was explained. However, they were not used to thinking in those terms, so questions like "Suppose we have 100 students who are low in physics, how many of them would be low in force and motion, too?" were not easy for them to answer.

Almond (2010) suggested using the joint correlation matrix of the competency variables as a way of assessing conditional probability tables in the competency model. (This is especially handy if existing studies are available to estimate this correlation matrix. Unfortunately, this was not true for *PP*.) The correlation matrix could be converted to a series of regressions, which could be discretised to produce the conditional probability table.

Instead of assessing the whole correlation matrix at once, the measurement team decided to directly elicit the parameters for the regressions. These have simple to interpret parameters (slopes, an intercept and a scale parameter related to the residual variance). This method had also been successfully used to build the Bayesian network for ACED (Shute, Hansen & Almond, 2008).

The DiBello–normal model combines a regression model with a simple trick introduced by Lou DiBello for representing discrete variables with continuous ones (Almond, et al. 2015). Consider a node with K parent variables in the graphical structure. For each parent variable, assign a continuous variable θ_k and assume that its distribution in the population of interest follows a standard normal (mean 0, variance 1) distribution. Then the *effective theta* of the child variable will be:

$$\tilde{\theta} = \frac{1}{\sqrt{K}} \sum_{k=1}^{K} a_k \theta_k - b \qquad (1)$$

The factor of $1/\sqrt{K}$ is for variance stabilization. As each θ_k has a standard normal distribution, the variance of the sum is $\sum a_k^2/K$. In particular, changing the number of parent variables does not change the scale of the child. By convention from educational testing, the negative intercept (or difficulty) is used instead of the intercept.

Equation 1 is a deterministic function of the parent variables. Most of the time, there should be some random variability around $\tilde{\theta}$. To make Equation 1 a regression add a residual variance, ϵ, which follows a normal distribution with mean 0 and standard deviation a_0.

To go from the regression model to a conditional probability table, it is necessary to discretise the thetas. First, consider a θ_k corresponding to a parent variable X_k which has m_k states. As θ_k follows a normal distribution, divide the real line up into m_k parts with equal area. The cut between State *s* and State $s+1$ is $\Phi^{-1}(\frac{s}{m_k})$, where $\Phi^{-1}(\cdot)$ is the inverse normal cumulative distribution function. Now associate each state, *s,* with the midpoint with respect to the normal distribution of that interval so that when $X_k = s$ then $\theta_k = \Phi^{-1}(\frac{s-1/2}{m_k})$.

As each row of the conditional probability table corresponds to a mapping of X_k to a state, s_k, it is straightforward to calculate an effective theta, $\tilde{\theta}$, for each row. The conditional probability distribution is then approximately a normal distribution with mean $\tilde{\theta}$ and standard deviation a_0. Discretising this normal distribution yields the values for the row of the conditional probability distribution.

To discretise the child variable, let *m* be the number of states. As before, set up the normal cut points and let $c_0 = -\infty$ and $c_m = +\infty$. Then the probability that the child variable will fall in State *s* is $\Phi\left(\frac{c_s - \tilde{\theta}}{a_0}\right) - \Phi\left(\frac{c_{s-1} - \tilde{\theta}}{a_0}\right)$. This can be used to calculate all of the values for each row of the conditional probability table.

The Peanut package (Almond, 2015, 2017a) provides functions that compute the conditional probability tables in the R language (R Core Team, 2017). Peanut also provides a mechanism for associating the parameters (the slopes or alphas, the intercept or beta, and the link scale parameter a_0) with the node in the network. Thus the conditional probability distributions for all of the nodes in the competency model could be expressed with a series of statements in R.

2.4. TRANSCRIPTION THE CODE TO A WORD DOCUMENT

The Peanut package works by associating meta-data with each node about how to build its conditional probability table. Figure 5 shows an example for the node ForceAndMotion which has a single parent, Physics (Figure 4). The first line of code simply picks binds the node to a variable. The assignments of `PnodeRules(fam)` to "Compensatory", and the assignment of `PnodeLink(fam)` to "normalLink" establish that model to be used is an additive (compensatory) regression (normal link). In other words, it selects the model parametrisation described in Section 2.3. The last three lines of code set the parameters of the distribution. `PnodeAlphas(fam)` sets the slope (discrimination) parameters a_k. Note that this is a labelled vector of numbers, as if there was more than one parent, each parent would get a separate slope. `PnodeBetas(fam)` sets the difficulty (negative intercept). The `PnodeLinkScale(fam)` function sets the residual variance, a_0.

```
fam <- PP.High$ForceAndMotion
PnodeRules(fam) <- "Compensatory"
PnodeLink(fam) <- "normalLink"
PnodeLinkScale(fam) <- sqrt(.2)
PnodeAlphas(fam)<-c(Physics=sqrt(.8))
PnodeBetas(fam) <- -.5
```

Figure 5: R code to describe the competency model

The numbers are default numbers supplied by the measurement team. The negative value for the beta value indicates that the ForceAndMotion skill is somewhat easier than the parent Physics skill; that is more students should be at a higher level of mastery of ForceAndMotion than of Physics (in general).

The link scale and alpha parameters are best understood as a unit. In particular, the multiple correlation coefficient for the regression is:

$$R^2 = \frac{\sum a_k^2/K}{\sum a_k^2/K \; + a_0^2}$$

The values of $a_1 = \sqrt{.8}$ and $a_0 = \sqrt{.2}$ were chosen so that $R^2 = .8$; in other words, changes in the parent variable explain about 80% of the variability of the child variable.

Unlike the graphical structure, where the measurement team presented two choices, for the parameters, the

measurement team presented a single choice. This will activate the anchoring heuristic (Tversky & Kahneman, 1974) influencing the Physics experts to provide numbers closer to the ones suggested by the measurement experts. In our previous experience, this is necessary because experts are used to working with observed score correlations, not the correlations between the latent variables (tetrachoric correlations). Thus the measurement team deliberately starts with a high R^2 hoping to anchor the physics experts towards the higher values.

The plan was for the measurement team to make the first pass suggesting numbers for the conditional probability table, and then have that work reviewed by the physics experts. However, the R code will not be a friendly representation for the experts. Therefore, the measurement team transcribed the R code into natural language. The transcription of the code from Figure 5 is shown in Figure 6.

Force and Motion: its parent is physics only.
We are setting a regression of force and motion on physics understanding.

Link scale parameter only gives us R-squared, which is the percent of the explained variance by the predictors on the outcome variable. The value of R-squared is 0.8.

The shift of about half of the standard deviation (.5) up towards more people having the skill. The shift is telling us a person who is medium on the on the parent variable is going to be somewhere about half way between medium and high. Most of the weights were split between medium and high.

Figure 6: Transcription of the R code for physics experts

In addition to transcribing the code into natural language, the measurement team calculated the conditional probability table for the experts. Table 3 shows the CPTs corresponding to the parameters in Figure 5.

Table 3: CPT for force and motion with physics

	Force and Motion		
Physics	High	Medium	Low
High	0.98	0.02	0.00
Medium	0.56	0.42	0.02
Low	0.04	0.52	0.44

Table 3 shows the effect of the difficulty parameter. The shift of ½ standard deviation means that students who are low in Physics are evenly divided between medium and low on Force and Motion. Students at the medium level of Physics are split between high and low, and almost all students who are high in Physics are high in Force and Motion. (The zero in the first row is not a structural zero, but rather a small probability that rounds to zero).

Figure 7 shows another example for the node Energy which has two parents: Physics and ForceAndMotion. The code is similar to that used in Figure 5 except for the fourth line. Since Energy has two parents, `PnodeAlphas(eng)` sets the two slopes parameters at the same time; the values are tagged as to which parents they belong to. Note that, the choice of the `"Compensatory"` rule in the code actually makes a difference because the node Energy has two parents.

```
eng <- PP.High$Energy
PnodeRules(eng) <- "Compensatory"
PnodeLink(eng) <- "normalLink"
PnodeLinkScale(eng) <- sqrt(.2)
PnodeLnAlphas(eng)<-
log(c(Physics=sqrt(.7),ForceAndMotion=
sqrt(.9)))
PnodeBetas(eng) <- 0
```

Figure 7: R code to describe a one child two parents competency model

Figure 8 shows the transcription of Figure 7 into natural language. Now the general R^2 shows that the parent variables jointly explain about 80% of the variability in the child variable. The relative values of the coefficients shows that force and motion, which is often taught right before energy, is more important than general physics understanding.

Energy Can Transfer: its parents are Physics Understanding and Force/Motion

This skill has a compensatory rule, which determines a combination of the necessary skills with weights.

We are setting a regression of force/motion and energy on physics understanding. This model has two predictors naturally. Energy has both direct and indirect effect on physics understanding. Indirect effect goes through force/motion.

One skill can offset another skill in this model. Link scale parameter only gives us R-squared, which is the explained variance by the predictors. The average value of R-squared is 0.8.

0.8 comes from an average of physics=0.7 and force/motion=0.9.

There is no difficulty shift. Somebody who is medium on the average of the parent variables will be roughly medium on the child variable.

Figure 8: Transcription of the R code for physics experts

Table 4 presents one child and two parents' CPT. Node Energy does not have a difficulty shift. The table shows the general structure of a compensatory model: the distribution of each row tends to centre around the average of the parent values. When both are the same, the distribution has a mode at that value. When the parent variables differ by one value, then most of the mass is split between those values in the child. For the two rows where one parent is high and the other low, most of the mass is in Medium.

Table 4: CPT for Energy having parents of force and motion and physics

Physics	Force and Motion	Energy-High	Energy-Medium	Energy-Low
High	High	0.96	0.04	0.00
Medium	High	0.69	0.30	0.01
Low	High	0.21	0.66	0.13
High	Medium	0.62	0.36	0.01
Medium	Medium	0.17	0.66	0.17
Low	Medium	0.01	0.36	0.62
High	Low	0.13	0.66	0.21
Medium	Low	0.01	0.30	0.69
Low	Low	0.00	0.04	0.96

2.5. GETTING CONSULTATION FROM CONTENT EXPERTS

As of this writing, the conversion into natural language is complete. The document has been presented to the Physics team for their review, although given the number of things they need to review and approve they have not yet completed their review. We expect this to be completed by the time of the conference so we be able to report on our experiences with this approach.

As before, we expect it will be an iterative process, with the experts making changes and the measurement team updating the numbers, regenerating the tables and the experts checking again. Furthermore, after the numbers are entered into the Bayesian network (again, done by the same R code), the network itself can be made available to check that more complicated queries have answers that match expectations.

3. FUTURE WORK

The next obvious step is an evaluation of how well this representation works. Working through the process of assigning conditional probability tables to the competency model with the physics experts should allow us to test the wording of the natural language descriptions as well as get feedback of which representations the physicists find useful and which are not. We hope to be able to discuss at least preliminary findings at the conference.

Another obvious next step is to automate the natural language generation. The text in Figure 6 is mostly template text, so writing functions in Peanut to generate it should be straightforward. Code to go in the other direction isn't necessary, as probably only a few numbers will change at each iteration; however, a web tool that made the natural language a form and then updated the numbers immediately might make the user experience even better.

Another issue which the proposed field trial should address is whether or not the parameters described in Section 2.3 are the right ones for elicitation. One simple question is, is it better to use the intercept or discrimination (negative slope). While the negation is simple to do in one's head, like negatively worded questions it adds cognitive load to the elicitation process.

A more complicated question is whether it is better to use R^2 instead of the residual standard deviation, a_0. The square of the multiple correlation is more familiar from elementary statistics and more directly interpretable. Suppose that in place of $a_0, a_1, \ldots, a_k$ the parameters were given as $R^2, \widetilde{a_1}, \ldots, \widetilde{a_k}$. Then $a_0 = \sqrt{(1 - R^2)}$ and

$$a_k = \widetilde{a_k} \sqrt{\frac{R^2}{\sum \widetilde{a_k}^2 / K}}.$$

The reparametrized model has another advantage. The neutral value for $\widetilde{a_k}$ is now one, with more important parents having higher values and less important parents having lower values.

Although the use of the graphical structure of the model in elicitation has been well known for a long time (Howard, 1989), translating the conditional probability tables into natural language adds a tool which could be useful in other projects. In particular, when combined with the regression model for conditional probability tables (Almond, 2010; Almond et al., 2015), it helps translate conditional probability tables into a language that might be more familiar to experts.

References

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*(3), 450-460.

Almond, R. G. (2010). "I can name that Bayesian network in two matrixes!". *International Journal of Approximate Reasoning*, *51*(2), 167-178.

Almond, R. G., Mislevy, R. J., Steinberg, L. S., Yan, D., & Williamson, D. M. (2015*). Bayesian Networks in Educational Assessment*. Springer.

Almond, R. (2015, July). An IRT-based parameterization for conditional probability tables. In J. M. Agosta & R. N. Carvalho (Eds.), *Proceedings of the twelfth UAI Bayesian modeling applications workshop (BMAW 2015)* (Vol. 1565, p. 14-23). Amsterdam, The Netherlands. Retrieved from http://ceur-ws.org/Vol-1565/bmaw2015paper4.pdf . (Additional material available at http://pluto.coe.fsu.edu/RNetica/)

Almond, R. G. (2017a, July). Peanut: An object-oriented framekwork for parameterized Bayesian networks (.3-1 ed.) [Computer software manual]. Retrieved from http://pluto.coe.fsu.edu/RNetica/Peanut.html (Open source software package)

De Klerk, S., Eggen, T. J. H. M., & Veldkamp, B. P. (2015). Psychometric analysis of the performance data of simulation-based assessment: A systematic review and a Bayesian network example. *Computers & Education*, *85*, 23-34.

Howard, R. (1989). Knowledge Maps. *Management Science,* **35,** 903–902.

NGSS Lead States (2013). *Next Generation Science Standards: For States, By State*s. Washington, DC: The National Academies Press.

Shute, V. J., & Rahimi, S. (2017). Review of computer based assessment for learning in elementary and secondary education. *Journal of Computer Assisted Learning*, *33*(1), 1-19.

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

Tversky, A. & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *Science*, **185,** 1124–1131.

Acknowledgements

The development of *Physics Playground* is supported by the National Science Foundation project Game-based Assessment and Support of STEM-related Competencies (#1628937, Val Shute, PI).