Tabular views of Bayesian Networks

Russell G. Almond*
Florida State University
mailto:ralmond@fsu.edu

Abstract

At the Bayesian modelling application workshop ten years ago, Almond suggested representing Bayesian networks with two matrixes. As the matrixes could be edited with standard office software, this proved to be useful in communicating between network designers and subject matter experts. The current paper describes an extension of that idea, now describing the network using four tables, as well as development of an open source software package which implements the round trip between between the tabular and graphical views of the network. These open source tools are currently being used to support the scoring engine development for the educational game *Physics Playground*.

Key words: Bayesian Networks, Elicitation, Q-Matrix, Assessment Design, Knowledge Representation

1 Knowledge Engineering for Bayesian Networks

Applications of Bayesian networks require a team of people. For example the Biomass project (see Almond, Mislevy, Steinberg, Williamson, & Yan, 2006) required experts in Biology, Biology Pedagogy, Item Writing, Cognitive Science, Psychometrics, Bayesian Networks, Graphic Design and Computer Application Development. Sometimes one person can take more than one role, but even so, they seldom have enough time to do all of the work.

The graphical visualisation of the Bayesian network provides a nice central focus for the project. In educational applications, building the structure of the Bayesian network forces the design team to wrestle with the critical issue of *what is being measured*. It then provides focus for the item writers who then construct tasks which will provide evidence about the targeted competencies.

However, when building the conditional probability tables which make up the Bayesian network, the graphical representation is not as helpful. It does answer the question of which variables need to be considered in defining a conditional probability distribution for a child variable, but not how their influence should be combined. The conditional probability table itself is a bit too technical for most experts to grapple with.

Ten years ago, Almond (2007) (see also Almond, 2010) proposed a different representation using two matrixes: Q—which expresses the relationship between the core competency variables and the observable outcomes from tasks,— and Ω—which expresses the relationships among the core competency variables. In particular, the augmented Q-matrix was based on the spreadsheet used by the members of the ACED design team (Shute, Hansen, & Almond, 2008) to manage the work for the project. The *Physics Playground* project (Kim, Almond, & Shute, 2016; Almond, Kim, Shute, & Ventura, 2013) extended the ideas, and began to build a library of software in R (R Core Team, 2017; Almond, 2017b) to build conditional probability tables from the tabular representations.

This paper describes the effort to extend these tools to a system that will translate back and forth between tabular and normal network representations of Bayesian networks. The domain experts can edit the tabular representations with the spreadsheet program in their desktop office suite, and the modelling team can quickly pull the results into the Bayes net software to visualise and explore the implications of the experts' numbers. The software will be released as part of the Peanut (a corruption of Pnet, or parameterized net-

*Paper submitted to 15th Bayesian Modelling Application Workshop 2017, Sydney, Australia.

work) package.

2 Background

Peanut is being built to support the modelling efforts in the revision of the educational game *Physics Playground* (Shute & Ventura, 2013). In this game students manipulate objects in a two dimensional world with the goal of getting a ball to a goal (balloon). They must use their knowledge of physics to manipulate the trajectory of the ball, and each game level is designed to emphasise some part of Newtonian mechanics, at a middle school level. The goal is to be able to make inferences about a student's physics ability from observations made during game play. This can be used to guide the choice of learning supports to encourage student learning.

2.1 Hub and spoke model organisation

Almond and Mislevy (1999) suggested splitting the the model into a central student model describing the inferred state of knowledge about a student's competencies, and a collection of evidence models which describe the relationships between the competency variables and performance on a single task. The hub (competency model) is always a complete Bayesian network, and the spoke (evidence model) is a fragment referencing variables in the hub model. Call the variable references in the spoke model *stub variables.* A spoke model only needs some information about the number and names of the states of the spoke variables. The probability distribution for the stub variable is given in the hub model.

Typically, the observable outcome variables are in the spoke variables and the targets of interest are in the hub. To propagate evidence from the spoke to the hub, select the appropriate spoke for the task is selected and *adjoin* it to the hub. Adjoining consists of replacing the stub variables with the corresponding original variables in the hub. In the resulting *motif*—hub and spoke combination—the evidence can be propagated as usual. One the evidence is recorded, the spoke can be *detached* and discarded.

2.2 Matrix representations of graphical structure

A directed graph can be represented by an adjacency matrix, A; where $a_{i,j} = 1$ if and only if there is a directed edge from Node i to Node j. Undirected graphs can be represented with symmetric matrixes. Whittaker (1990) notes that if the analyst takes an inverse covariance matrix among the variables and drives the low values to zero and the high values to one, the result will be an description of an undirected graphical models which has approximately the same conditional independence constraints as the joint distribution. Almond (2007) notes that this can easily be turned into a directed graphical model by picking an orientation for the edges.

This paper calls the adjacency matrix for the variables in the hub model Ω. Almond (2007) notes that if the hub variables are all latent, frequently the information about the relationship between the latent variables come from factor analyses which naturally estimate the covariance matrix $\Sigma = \Omega^{-1}$.

While the Ω-matrix is useful for describing the graphical structure of the hub, the spokes usually take a different form. In educational testing, a very common case is that there is one observable outcome variable (often whether the student got the item right or wrong, or possibly a partial credit scale for the item). In that case, all of the parent variables for the observables are all stub variables, or references to variables in the hub.

These relationships are represented with a matrix Q. The rows of this matrix represent observables (spoke variables) and the columns represent competencies (hub variables). A value of $q_{j,k}$ means that Competency k is relevant for producing Observation j, with a zero indicating statistical independence (given the set of relevant proficiencies). This representation is common to not only Bayesian networks but also a number of other cognitively diagnostic models (Rupp, Templin, & Hensen, 2010).

As a means of representing the hub model, the Ω-matrix offers little advantage over a graphical visualisation. Its only real advantage comes later, when adding parameters (Section 3.3) or if the original source of information is from a factor analysis or other study which produces a correlation matrix.

The Q-matrix, in contrast, provides a compact view of what is happening across a large collection of spoke models. The original version of *Physics Playground* had 77 game levels, and approximately 75 new levels are under development for the new version. Summing the columns of the Q-matrix (even under the crude assumption of only one observable per level) provides a quick estimate of how much evidence is available for each of the competencies. In *Physics Playground* this can be used to target development activity towards designing game levels for which less evidence is available.

2.3 Parameterized conditional probability tables

There are a number of difficulties with directly assessing the conditional probability tables (CPTs) from experts: the number of values in the table grows exponentially with the number of parents, the representation is unfamiliar to many experts, it can be difficult to encode knowledge about how influences combine to create the CPT. In many situations the variables are ordered categorical variables, and CPTs which encode monotonicity (as skill increases so does the probability of getting the item right) are required.

Lou DiBello (Almond et al., 2001) suggested mapping each state of each parent variable onto a standard normal scale. Then the probability of the child variable can be expressed through a logistic regression-like model. Consider a binary node X_j which has parents $X_1, \ldots, X_K$. Let θ_k be a real value corresponding to the state of X_k. Then

$$\mathrm{P}(X_j = \mathrm{TRUE}|X_1, \ldots, X_K) = \mathrm{logit}^{-1}\left(\frac{1}{\sqrt{K}}\sum_{k} = 1^K \alpha_{jk}\theta_k - \beta_j\right) . \quad (1)$$

The parameters α_{jk}, following the language of item response theory (IRT; Hambleton, Swaminathan, & Rogers, 1991), are called *discriminations* and they can be thought of as slopes in a regression or loadings in a factor analysis. The negative intercept parameter β_j is called a *difficulty*. It corresponds to a point on the standard normal scale where a person would have a 50-50 shot of getting the item correct.

Although a function is needed to summarise across the parents in Equation 1, the choice of the sum (or average) is somewhat arbitrary. DiBello proposed a variety of combination rules to summarise across the parents:

Compensatory (Equation 1). In a compensatory CPT having more of one parent skill compensates for the lack of another. For example, suppose a physics problem could be solved by applying either Newton's laws or the principle of conservation of energy. People who know both would be better off than people who know only one because they could use one method to solve the problem and another to verify the solution.

Conjunctive For a conjunctive rule the sum is replaced with a minimum and the variance stabilisation constant $1/\sqrt{K}$ is no longer needed. Thus the parent with the smallest value tends to dominate the relationship. This combination rule is useful for modelling situations in which multiple skills are all needed to solve a task. For example, a task which requires a person to read a passage and write a summary requires both the reading and writing skills and that person's performance will be limited by the weaker skill.

Disjunctive Similar to the conjunctive rule, only using a maximum instead of the minimum, so the strongest skill predominates. This combination rule is useful for modelling situations in which the skill represent alternative solution paths. For example, if a student is presented the same information in both a reading passage and a audio lecture, then the student should perform at a level dominated by the stronger of that student's reading and listening skill.

In a compensatory model, the relative size of the discrimination parameters indicates the importance of the skill in solving a problem. n principle, this idea extends to the other two rules, but the resulting models do not have as natural an interpretation. Consider the disjunctive reading/listening task. If the written passage is clear and well written, but the audio stimulus is spoken by somebody with a strong accent who speaks quickly, the listening path would have a much higher demand than the reading path. The *offset conjunctive* and *offset disjunctive* combination rules offer a different parameterization, with a single discrimination and one difficulty per parent. The offset conjunctive rule is

$$\mathrm{P}(X_j = \mathrm{TRUE}|X_1, \ldots, X_K) = \mathrm{logit}^{-1}\left(\alpha_j \min_k(\theta_k - \beta_{jk})\right) ; \quad (2)$$

the offset disjunctive rule replaces the minimum with a maximum.

Note that the choice of the inverse logistic *link function* in both Equations 1 and 2 is also arbitrary. In fact, it only works for binary child variables. For ordered categorical variables, Almond et al. (2001) suggested using graded response model. Almond et al. (2006) also describes a normal link function, which uses an extra residual variance, or *link scale* parameter.

Almond (2007) noted that all of the information needed to build one of these DiBello models for an observable variable could be noted in a single line of a table. This starts with the row of the Q-matrix which describes the parents of the variable. In addition, columns are needed for the combination rule and link function, and if necessary the link scale parameter. A collection of columns are needed for the discrimination parameters and an additional column is needed for the intercept.

2.4 Multi-step response processes

The graded response link function works by modelling separate curves for each state: $\mathrm{P}(X_j \geq s|\theta_1, \ldots, \theta_k)$, then taking the difference between the curves. In order to ensure that the curves do not cross, and the differences always result in positive probabilities, the difficulty for each curve must be in increasing order and the discrimination parameters must be the same. This may or may not fit a particular situation.

This proved to be a problem in the first version of *Physics Playground* (Almond et al., 2013). In that version, players who solved the level earned a gold trophy for an efficient solution and a silver trophy for an inefficient solution (lots of attempts which did not work). The primary outcome was a variable indicating the trophy a player received in the level, and the values it could take on were: none, silver and gold. Some levels could be solved without much physics skill by simply attempting many things. In those levels, the difference between silver and gold provided much more evidence than the difference between silver and none. In other levels, unless the player had a pretty good idea of the physics involved, they did not even know how to approach a solution. In those levels, there was much more evidence in the jump between no trophy and sliver than there was between silver and gold. The requirement that the discrimination parameter must be the same for both of those transitions was overly restrictive.

To overcome this restriction, (Almond et al., 2013) proposed an alternative link function based on the generalised partial credit model. This model assumes that the student starts in the lowest state of the child variable and then makes a series of transitions to higher states. In particular, it is useful for modelling multistep transitions; for example, it models the transition from no trophy to any trophy and any trophy to gold trophy. Thus the new curves are $\mathrm{P}(X_j \geq s+1 | X_j \geq s, \theta_1, \ldots, \theta_k)$. With this choice of link function, the step transitions are no longer required to have the same discriminations, nor even required to have the same combination function or parent variables.

In this setup there is an inner and outer Q-matrix for the observable. The row in the outer Q-matrix corresponding to the observable indicates all of the parent variables that are used in any transition step, thus provides the topology of the graph. The inner Q-matrix has rows corresponding to states and indicates which of the parent variables are relevant for which state transitions. As the set of parameters is now different for each state transition, the augmented Q-matrix now must work at the level of the rows of this inner Q-matrix. So now there is one row in the table for every variable–state pair (except for the lowest state which does not need one).

With the addition of the partial credit link function, there are now three different link functions supported in the current Peanut implementation. All three have different requirements for supporting parameters.

Graded Response This approach models the thresholds for each state (aside from the first), and requires one discrimination parameter per parent as well as one difficulty parameter per state (except for the lowest).

Normal Link This approach assumes that the variables are discretized normal variables. It requires one discrimination parameter per parent, a single difficulty/intercept parameter, and a residual variance/link scale parameter.

Partial Credit This approach models the transitions between states. It potentially uses a different set of discrimination and difficulty parameters for each level (except the lowest), as well potentially a different combination rule for each state.

In particular, for the first two link functions, only a single row in a table can contain all of the parameters. For the partial credit link function, to fully express the model a different row is is needed for each state (except the lowest) of each variable.

3 The Four Tables

The first version of *Physics Playground* used an augmented Q-matrix to describe the model along with some custom code written in RNetica (Almond, 2017b). It also took advantage of the special structure of the game: all of the evidence models had identical structure. Therefore, evidence models for new levels could be created by copying a model skeleton and then changing the conditional probability tables for the level.

The second version of *Physics Playground* required tools able to make the round trip between the parameterized Bayesian networks (Pnets) and the augmented Q and Ω-matrixes. That way as the experts made changes in the spreadsheets, new networks could be quickly built to test the new implicit model.

It became apparent that the two matrixes were not sufficient to create the Bayesian networks. In particular, additional meta-data was needed about the networks and the nodes, and that would be awkward to store in the augmented Q and Ω matrixes. The result was a system of four tables: a network manifest (Section 3.1), a node manifest (Section 3.2), the augmented Ω-matrix (Section 3.3), and the augmented Q-matrix (Section 3.4)

3.1 The network manifest

Using the hub and spoke framework, a complete model is not just a single Bayesian network, but rather a collection of networks (hubs) and fragments (spokes). Thus, meta-data is needed for a collection of networks. Table 1 gives a sample.

The protocol assumes that the Bayesian network package has some way to attach meta-data (e.g., the description), to the network. Each node is given a name (a short identifier, possible excluding some characters like spaces), a title (a longer more human readable description), and a description. Finally, networks are associated with files.

The sample in Table 1 gives one hub (competency) model and two spokes. The spokes are marked by entries in the hub column indicating which model they are meant to be attached to. Note that there is no restriction here, the same model could potentially be a hub and a spoke in different contexts. This would allow for more complex variants on the simple hub and spoke model of Almond and Mislevy (1999).

3.2 The node manifest

There are two complications with making the node manifest. The first is that the node names are not unique. In most Bayesian network packages, each network defines its own namespace; for example, many evidence models might have a variable "isCorrect." Thus, the key for a node is (*Model*, *NodeName*). The second complication is that each node could have a different number of states each with its own meta-data. Thus, the primary key for the node manifest table is (*Model*, *NodeName*, *StateName*) and multiple rows of the table correspond to a single node. Table 2 shows an example.

Once again, the node is given both a short name and a longer title as well as a description. The `NodeLabels` column is based on a feature of Netica that has proved very handy in writing applications. The labels field contains a comma separated list of identifiers. The nodes which share a common label form a useful subset of the nodes. For example, *EngXfer* is labelled both "Proficiencies" and "LowLevel". Presumable a reporting feature could make a report based on all nodes with one of those labels.

Table 2(b) shows the remaining columns. Each row corresponds to a different state, so the state dependent features are on display. The *StateValue* column provides a real number corresponding to this state, effectively mapping the ordinal variable to an interval scale. This can be used to calculate expected values. It is also used at the theta value in Equations 1 and 2. Although the state description fields seldom have more than a cosmetic use in Bayesian network software, they are critically important in model development; hence their inclusion in the node manifest. In addition to naming the states, the states of the variable must be given an operational definition. In other words, what does it mean for a particular problem to be in a particular state?

First consider the two observable nodes, *ConjObs* and *TwoStepObs*. These correspond to tasks which are presumably scored right/wrong and with some kind of partial credit scoring. For the second observable, it takes on the value `Full`, `Partial` or `None` depending on how many steps the student completed. Note that this definition is not yet operational, as how one would know that the first or second step was completed has not yet been defined. The second observable omits the description. Presumably there is a key, and if the provide answer matches the key, the item is score correct. Still there may be questions. For example, if the expected answer is a fraction, is the answer required to be reduced to simplest form?

In the case of the proficiency variables, the question is even trickier. There is often not a concrete description of what the variable means, rather it is a psychological construct which is thought to be useful. Wilson (2005) develops a *construct map*, which is a diagram which shows the latent variable on a scale and identifies regions of the scale with characterisations of people in that region, or tasks that people at that level could perform. Almond, Kim, Velasquez, and Shute (2014) shows some operational examples. The node manifest provides a tabular representation of the construct map. Note that the example provided is again still not sufficiently complete. It talks about "complex" and "simple" problems, but does not describe how they differ.

Getting good operational definitions of the key variables early in the design process is important. In *Physics Playground*, high, medium and low values for a physics understanding variable (like *EngXfer*) is relative to the target population, in this case middle school students. Having taken several years of Physics in college, I tend to think about problems in Newtonian mechanics using vector notation. However, only the most mathematically advanced middle school students take algebra during middle school, and vectors are not covered extensively in the early algebra courses. Thus, my mental model for high physics understanding is something that would be unrealistic for middle school science. Discussing this issue with the experts in Physics pedagogy (and my daughters who recently graduated from middle school) helped me gain a better understanding of what to expect from our students. This,

Table 1: A Network Manifest

Name	Title	Hub	Pathname	Description
miniPP_CM	Physics Playground Excerpt		miniPP-CM.dne	A few selected nodes from Physics Playground for testing Peanut/PNetica
PPcompEM	Compensatory Evidence Model	miniPP_CM	PPcompEM.dne	An evidence model with a single compensatory observable
PPconjEM	Conjunctive Evidence Model	miniPP_CM	PPconjEM.dne	An evidence model with a single conjunctive observable

in turn, helped the design team design game levels that are better targeted for the audience.

3.3 The augmented Ω matrix

Although in principle, any type of model could be used for either the competency variables in the hub model or the observables in the spoke models, in practice, experts tend to think about them very differently. The Ω-matrix is designed to handle the variables in the hub (in which the parents could be any other variable in the model, as long as the acyclicity condition is enforced). In the Ω-matrix, each variable appears as both a row and a column. The Q-matrix (Section 3.4), in contrast, is designed to support observable in the spoke models where most of the parent variables are actually stubs referencing the competency variables in the hub.

For thinking about the relationship among competency variables, the normal link function tends to be the most useful. In particular, the other two link functions (graded response and partial credit) assume a tendency to cause from the parent to the child variables; that is, the skill of the student causes a strong or weak performance on the task. In the competency model, the direction of causality might not be known, or may be an artefact of the way the subject is taught. For example, there may be a high correlation between the nodes representing understanding of Newton's laws and the principle of transfer of energy. That correlation could be because knowledge of one helps the student learn the other (in either direction). It could also be because the two subjects are often taught in the same class together or in a specific sequence. It could be because some unmeasured variable (e.g., mathematical sophistication) drives both.

Almond (2007) originally proposed starting with the (inverse) correlation and using that to build regression models. For eliciting parameters from experts, going directly to the regression models is often more useful. This is represented by the choice of compensatory rule and normal link function and requires columns for slopes, intercepts and a link scale (residual standard deviation) parameter. Although there is some redundancy between the structure and the non-zero slopes, putting the slopes in a separate set of columns both separates structural and parametric decisions and allows additional error checking.

To illustrate the Ω matrix, Figure 1 shows an excerpt from the *Physics Playground* competency model and Table 3 shows the corresponding Ω-matrix. The left hand columns (Table 3(a)) show the adjacency matrix corresponding to the graphical structure in Figure 1. They also show the choice of link function and combination rule, which are always normal and compensatory (the regression model).

Table 3(b) shows the remaining columns. The columns are given a new name by prepending "A" to the variable name. All of the "A" columns show the regression weights (or factor loadings), except for the values on the diagonal. As this value is always not applicable, it is instead used for the link scale (residual standard deviation) parameter. The column marked "B" is the difficulty (negative intercept). The last column gives a prior weight to be used when learning the models (Almond, 2015).

This representation trades compactness (one line per variable) for expressiveness. In particular, it really only supports the graded response and normal link function models. The partial credit models potentially need a lot more parameters.

3.4 The augmented Q matrix

Three different evidence models will serve to illustrate the various ways in which the augmented Q-matrix works. Each one has a single observable variable, but it has different relationships with the parent variables. The first, `PPcompEM`, has a single dichotomously scored observable, *CompObs*, and uses the compensatory combination rule—student success is driven by the average of the parent abilities. The second, `PPconjEM`, has a single dichotomously scored observable, *ConjObs*, and uses the offset conjunctive combination rule, both skills must meet the thresh-

Table 2: Node Manifest

(a) Left hand (not-state specific) columns.

Model	NodeName	ModelHub	NodeTitle	NodeDescription	NodeLabels
miniPP_CM	EngXfer		Energy can Transfer	Energy can transfer from one object to another.	pnodes, LowLevel, Proficiencies
miniPP_CM	EngXfer				
miniPP_CM	EngXfer				
PPconjEM	ConjObs	miniPP_CM	Conjunctive Observable	A binary response whose probability of success is related to average of parent variables.	onodes, Observables, pnodes
PPconjEM	ConjObs				
PPtwostepEM	TwoStepObs	miniPP_CM	Partial Credit observable	A partial credit response where each step requires different inputs.	onodes, Observables, pnodes
PPtwostepEM	TwoStepObs				
PPtwostepEM	TwoStepObs				

(b) Right hand (state specific) columns).

Model	NodeName	Nstates	StateName	StateTitle	StateDescription	StateValue
miniPP_CM	EngXfer	3	High		Can use to solve complex problems	0.97
miniPP_CM	EngXfer		Medium		Can use to solve simple but not difficult problems	0.00
miniPP_CM	EngXfer		Low		Can not solve simple problems.	-0.97
PPconjEM	ConjObs	2	Right			
PPconjEM	ConjObs		Wrong			
PPtwostepEM	TwoStepObs	3	Full	Complete Solution		
PPtwostepEM	TwoStepObs		Partial	First step but not second		
PPtwostepEM	TwoStepObs		None	No attempt or failed first step		

old for there to be a high chance of success. The last model, `PPtwostepEM`, has an observable, *TwoStepObs*, with three possible values: None (no credit), Partial (credit) and Full (credit). The idea is that it represents a task with two steps: solving the problem and selecting an explanation. Partial credit is awarded for completing the first, and full credit for completing both. The two steps use different combinations of skills.

Table 4 shows a sample Q-matrix. Each variable is identified with two columns, the model and the node and takes one row fewer than the number of states. Note that the two rows for `TwoStepObs` are not the same: the transition from no to partial credit requires both energy transfer and iterative design, but the transition to full credit only involves the energy transfer. The list of parents for the *TwoStepObs* variable is the set of variable which have a one in any of the rows.

All three models use the partial credit link. The partial credit and graded response links are identical when the variable is binary, so this only really matters for the third variable.

The last part of the table (Table 4(c)) shows the discrimination parameters (starting with "A") and the difficulty parameter "B". It also shows the combination rule used. Note that for the third variable a different rule is used for each transition. One complication is that the compensatory and offset conjunctive rules have different parameterizations; the compensatory rule has one discrimination parameter per parent and a single difficulty, while the offset conjunctive rule has one difficulty per parent and a single dis-

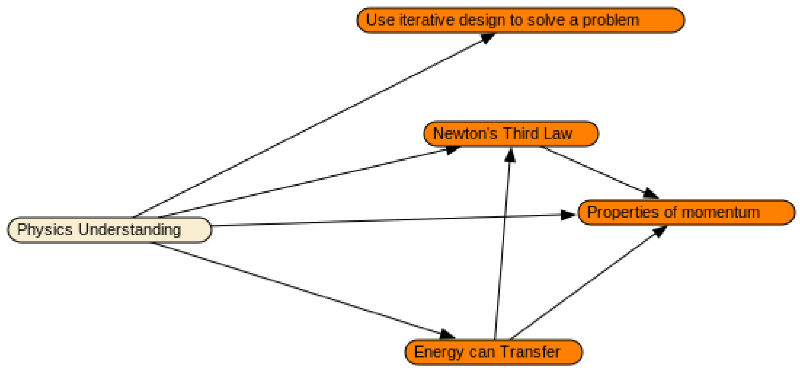

Figure 1: Excerpt from Physics Playground Competency Model

Table 3: Ω-Matrix

(a) Left side columns showing structure.

Node	Physics	EngXfer	IterDes	NTL	POfMom	Link	Rules
Physics	1	0	0	0	0	normalLink	Compensatory
EngXfer	1	1	0	0	0	normalLink	Compensatory
IterDes	1	0	1	0	0	normalLink	Compensatory
NTL	1	1	0	1	0	normalLink	Compensatory
POfMom	1	1	0	1	1	normalLink	Compensatory

(b) Right side columns showing loadings.

Node	A.Physics	A.EngXfer	A.IterDes	A.NTL	A.POfMom	Intercept	Weight
Physics	1.00					0.00	
EngXfer	0.89	0.45				0.00	
IterDes	0.89		0.45			-0.20	
NTL	0.95	0.84		0.45		0.30	
POfMom	0.77	0.95		0.95	0.45	0.00	

Table 4: Q-matrix

(a) Q-matrix, left side columns showing meta-data.

Model	Node	NStates	State	Link	LinkScale
PPcompEM	CompObs	2	Right	partialCredit	0
PPconjEM	ConjObs	2	Right	partialCredit	0
PPtwostepEM	TwoStepObs	3	Full	partialCredit	0
PPtwostepEM	TwoStepObs		Partial		

(b) Q-matrix, middle columns showing Q-matrix.

Model	Node	POfMom	NTL	EngXfer	IterDes	Rules
PPcompEM	CompObs	1	1	0	0	Compensatory
PPconjEM	ConjObs	1	0	1	0	OffsetConjunctive
PPtwostepEM	TwoStepObs	0	0	1	0	Compensatory
PPtwostepEM	TwoStepObs	0	0	1	1	OffsetConjunctive

(c) Q-matrix, right side columns showing parameters.

Model	Node	A.POfMom	A.NTL	A.EngXfer	A.IterDes	B	Weight
PPcompEM	CompObs	1.10	0.90			0.30	
PPconjEM	ConjObs	0.50		-0.50		1.00	
PPtwostepEM	TwoStepObs			1.00		0.50	
PPtwostepEM	TwoStepObs			1.00	0.10	1.00	

crimination. To fit these into the table structure the roles of “A” and “B” are swapped for the offset conjunctive (and disjunctive) rules. It is unclear whether or not this overloading of the meaning of the columns will be too complex for users in practice.

Once again, the prior weight column is used by learning algorithms where the expert numbers would be taken as a prior distribution. In fact, all of the columns of the table correspond to properties of the Pnode (parameterized node) object in Peanut (Almond, 2017a, 2015).

4 Testing, Limitations and Future Work

Currently, the software for round-tripping between the tabular and graphical views is under development. It is mostly complete and testing is expected to be complete by the time of the conference. This includes both ordinary functional testing and using the tables operationally as part of the *Physics Playground* development efforts. This give us experience with how both the physics experts (actually, experts in physics pedagogy) and the other member of the design team (mostly graduate students) react to these representations. Changes are likely, and code will be posted to the RNetica web site.

A possible limitation is that the Ω-matrix only uses one row per variable. This format is more compact, and hence easier to use, but it also limits the complexity of the distribution descriptions. Another limitation is that the Q-matrix does not have any way to indicate connections between two observables in an evidence (spoke) model. So far, these are not needed for *Physics Playground*, but they might be in the future, or in future projects. In these cases, the tabular formats will need to be revisited.

Peanut is meant to be agnostic to the Bayesian network package used. Although I have tried to make sure the operations are fairly standard, all testing has been done using Netica. In particular, I make use of Netica’s ability to add meta-data to a node or network object. I suspect that many Bayesian network packages support this, but I am not sure. The real proof of its generalisability will come when an implementation is created using a different Bayes net package.

I hope that the tools used for the *Physics Playground* project will be useful to others developing Bayesian networks. They are available at `https://pluto.coe.fsu.edu/RNetica/` and will be updated as development and testing continues. I hope to be able to report more on our own use of them at the workshop.

References

Almond, R. G. (2007). ‘I can name that Bayesian network in two matrixes’. In K. B. Laskey, S. M. Mahoney, & J. A. Goldsmith (Eds.), *Proceedings of the 5th UAI bayesian modelling applications workshop* (Vol. 268, p. (online)).

CEUR Workshop Proceedings. Retrieved from http://ceur-ws.org/Vol-268/paper1.pdf (Held in conjunction with the 23rd Conference on Uncertainty in Artificial Intelligence (UAI-2007))

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

Almond, R. G. (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/bmaw2015_paper4.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)

Almond, R. G. (2017b, May). RNetica: Binding the Netica API in R (-4.6 ed.) [Computer software manual]. Retrieved from http://pluto.coe.fsu.edu/RNetica/ (Open source software package)

Almond, R. G., DiBello, L., Jenkins, F., Mislevy, R. J., Senturk, D., Steinberg, L., & Yan, D. (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., Shute, V. J., & Ventura, M. (2013). Debugging the evidence chain. In R. G. Almond & O. Mengshoel (Eds.), *Proceedings of the 2013 UAI application workshops: Big data meet complex models and models for spatial, temporal and network data (UAI2013AW)* (pp. 1–10). Retrieved from http://ceur-ws.org/Vol-1024/paper-01.pdf

Almond, R. G., Kim, Y. J., Velasquez, G., & Shute, V. J. (2014). How task features impact evidence from assessments embedded in simulations and games. *Measurement: Interdisciplinary Research and Perspectives*, *12*(1–2), 1–33. (with Discussion) doi: 10.1080/15366367.2014.910060

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

Almond, R. G., Mislevy, R. J., Steinberg, L. S., Williamson, D. M., & Yan, D. (2006). *Bayesian networks in educational assessment.* (Manuscript in preparation to be published by Springer)

Hambleton, R. K., Swaminathan, H., & Rogers, H. J. (1991). *Fundamentals of item response theory.* Sage.

Kim, Y. J., Almond, R. G., & Shute, V. J. (2016). Applying evidence-centered design for development of game-based assessments in Physics Playground. *International Journal of Testing*, *16*(2), 142-163. (Special issue on cogntive diagnostic modeling) doi: 15305058.2015.1108322

R Core Team. (2017). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from https://www.R-project.org/

Rupp, A. A., Templin, J., & Hensen, R. J. (2010). *Diagnostic measurement: Theory, methods, and applications.* Guilford Press.

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.

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

Whittaker, J. (1990). *Graphical models in applied multivariate statistics.* Wiley.

Wilson, M. (2005). *Constructing measures: An item response modeling approach.* Psychology Press.

Acknowledgements

ACED development and data collection was sponsored by National Science Foundation Grant No. 0313202. The initial round of *Physics Playground* development was supported by the Bill & Melinda Gates Foundation grant *Games as Learning/Assessment: Stealth Assessment* (#0PP1035331, Val Shute, PI) and the National Science foundation grant *DIP: Game-based Assessment and Support of STEM-related Competencies* (#1628937, Val Shute, PI). The tool development is the work of the author and bug reports should be sent to him (not Val).