Tabular views of Bayesian Networks

Russell Almond

Educational Psychology and Learning Systems
College of Education
Florida State University

Bayesian Modeling Application Workshop 2017

“I can name that Bayes net in 2 matrixes”

10 years ago (at BMAW 2007), I proposed a way of representing a Bayes net using two matrixes:

- A correlation matrix (Ω) which gives the correlation among the central variables that represent the state of the system.
  - In educational applications, this is proficiency or competency model.
- A Q-matrix which shows which observable variables provide evidence about which of the central variables.
  - *Evidence Models*
  - In educational application, which skills are required to solve which items.

Application History

2007 ACED — a tutoring system for geometric series.
- Noted design team was using Q-matrix to manage design

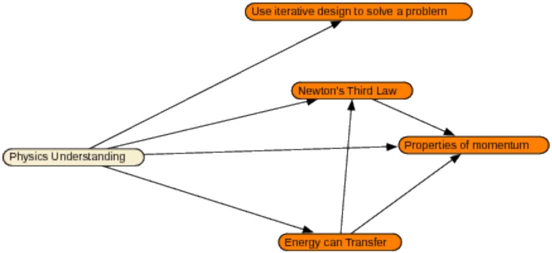
2014 Physics Playground, version 1. Physics learning game.
- One designer filled out Q-matrix.
- Custom R code to go from Q-matrix to Netica net (RNetica, CPTtools)

2017 Physics Playground, version 2. Physics more levels, adaptive algorithm.
- Round trip between Bayes nets and matrixes (Peanut, PNetica).
- Two matrixes were not quite enough!

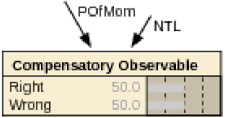
Hub and Spoke Model

- Educational Models have a special structure.
- Central *Competency Model* or *Proficiency Model* contains variables of interest.
- A collection of *Evidence Models* corresponding to *tasks* or activities that a student does.
  - These would be called *tests* in a medical or system diagnosis application.
- Evidence models link observable variables to variables in the center competency model.
  - Model fragment: competency variables are just stubs (references).
- A test or *assessment* is made up of a collection of tasks.
  - Bayes net for assessment (competency plus relevant evidence models) is called a *motif*.

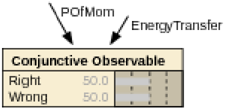
Hub and Spoke Example



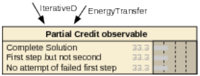
Hub: Excerpt from Physics Playground Competency Model (miniPP-CM)



Spoke 1: Compensatory Evidence Model (PPconjEM)



Spoke 3: Conjunctive Evidence Model (PPconjEM)

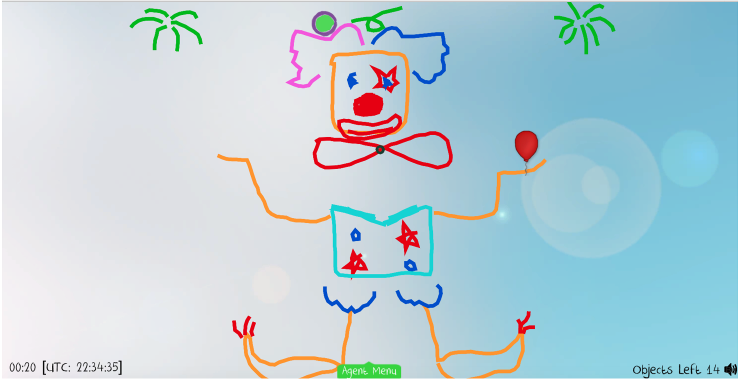


Spoke 3: Two-step Evidence Model (PPtwostepEM)

Physics Playground

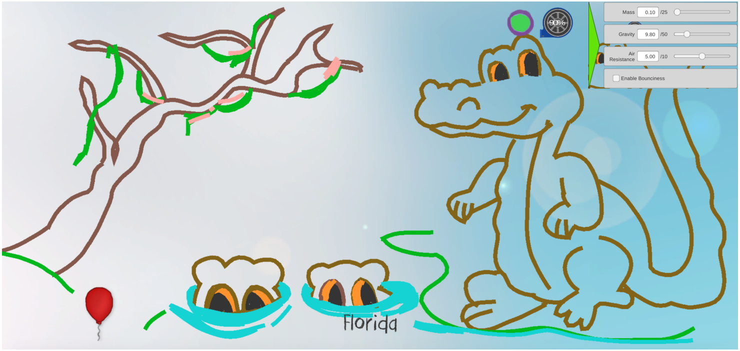
- Students explore a 2D world which follows laws of Physics.
- Can use performance in game to learn about students' knowledge of physics. (Bayes net used for scoring.)
- Version 2.0 adds feedback and instruction, new manipulation levels.
- Eventually, Bayes net will be used for sequencing of levels, triggering feedback.
- Game Demo: `http://www.empiricalgames.org/ppunity5/LE/1.2.8/demo.html`
- Level Editor: `http://www.empiricalgames.org/ppunity5/LE/1.3.1/demo.html`

Drawing Level



Goal: Get the ball to the balloon by drawing objects on the screen.

Manipulation Level



Goal: Get the ball to the balloon by manipulating, the ball's mass, gravity, air resistance and bounciness.

Physics Playground Development Team

- Task Developers/Learning Supports
- Physics Experts
- Programmers
- Statistical Modelers
- Management
- Bayes net diagram used as common reference
  - Lots of time spent nailing down definitions of the Physics variables.
- Q-matrix used to schedule work (count up how many levels related to each competency variable).

Meta-data tables

- Goal is to have software which will round trip between Bayes net (Netica) model and two matrixes.
- Need to be able to create Bayes nets and nodes on the fly.
- Want to capture meta-data about networks and nodes in the networks.
- Based off of meta-data capabilities of Netica.

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 single compensatory observable
PPconjEM	Conjunctive Evidence Model	miniPP_CM	PPconjEM.dne	An evidence model with single conjunctive observable

Network Table Fields

Name
: Short name (must follow variable naming conventions)

Title
: Longer human readable name

Hub
: If not the hub model, the name of the hub model to which this network attaches

Pathname
: Pathname for storing the Bayes net.

Description
: Description of the name.

Note that in *Physics Playground* all of the networks except the competency model are associated with game levels.

Node Manifest LHS (Node-specific fields)

Model	NodeName	ModelHub	NodeTitle	NodeDescription	NodeLabels
miniPP_CM	EngXfer		Energy can Transfer	Energy can transfer from one object to another.	pnodes, LowLevel, Proficiencie
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				

Node Manifest RHS (State-specific fields)

Model	NodeName	Nstates	StateName	StateTitle	StateDescription	StateVa
miniPP_CM	EngXfer	3	High		Can use to solve difficult problems	0
miniPP_CM	EngXfer		Medium		Can use to solve simple but not difficult problems	0
miniPP_CM	EngXfer		Low		Can not solve simple problems.	-0
PPconjEM	ConjObs	2	Right			
PPconjEM	ConjObs		Wrong			
PPtwostepEM	TwoStepObs	3	Full	Full Credit	Complete Solution	
PPtwostepEM	TwoStepObs		Partial	Partial Credit	First step but not second	
PPtwostepEM	TwoStepObs		None	No Credit	No attempt or failed first step	

Node Manifest Fields (Node-specific)

Node-level fields (only one needed per node):

Model*	Name of model (see Net Manifest)
NodeName*	Name of the node (variable name restriction)
ModelHub	Name of hub model (redundant with Net Manifest)
NodeTitle	Longer, human readable name
NodeDescription	Description of what the node measures.
NodeLabels	Comma separated list of identifiers for node subsets. (e.g., high-level and low-level competencies.)

* Key for identifying a node.

Node Manifest Fields (state-specific)

State-level fields (one per state):

Model*	Name of model (see Net Manifest)
NodeName*	Name of the node (variable name restriction)
Nstates	Number of states.
StateName	Name of state (variable name restriction)
StateTitle	Longer, human readable name.
StateDescription	Description of state.
StateValue	Numeric value for state (used for calculating expected value, and other CPT calculations).

* Key for identifying a node.

DiBello Partial Credit Models

- Parameterization for conditional probability tables in evidence models.
- Map each scale point of each skill to a standard normal (mean 0, variance 1) scale. $\tilde{\theta}_k$
  - Can use quantiles of the normal distribution (Almond et al., 2015).
- Combine the input variables into a single scale point using a *structure function*.
- Use an Item Response Theory (IRT) model to come up with conditional probability tables for each row.
- DiBello originally suggested graded response model, but I've switched to partial credit model.
- `CPTTools` R package has code for these models.

Regression-like structure functions

Compensatory Performance is dominated by (weighted) average of the skills.

$$Z(\tilde{\boldsymbol{\theta}}) = \frac{1}{\sqrt{K}} \sum_{\text{parents}} a_k \tilde{\theta}_k - b$$

Conjunctive Performance is dominated by weakest skill.

$$Z(\tilde{\boldsymbol{\theta}}) = \min_{\text{parents}} a_k \tilde{\theta}_k - b$$

Disjunctive Performance is dominated by strongest skill.

$$Z(\tilde{\boldsymbol{\theta}}) = \max_{\text{parents}} a_k \tilde{\theta}_k - b$$

- Parameter a_k represents *discrimination* (regression slope), one per parent variable.
- Parameter b represents *difficulty* (negative intercept), one per state of child variable.
- Multiple-a pattern structure functions.

Offset Conjunctive and Disjunctive Models

Conjunctive Performance is dominated by weakest skill. (All skills needed to solve problem.)

$$Z(\tilde{\boldsymbol{\theta}}) = a \min_{\text{parents}} (\tilde{\theta}_k - b_k)$$

Disjunctive Performance is dominated by strongest skill. (Multiple solution paths, assume student takes best for them.)

$$Z(\tilde{\boldsymbol{\theta}}) = a \max_{\text{parents}} (\tilde{\theta}_k - b_k)$$

- Parameter a represents *discrimination* (regression slope), one per child variable.
- Parameter b_k represents *difficulty* (negative intercept), one per parent variable.
- Multiple-b pattern structure functions.

Partial Credit Model

- Observable variable takes on states $0, \ldots, S$
- $\Pr(X \geq s | X \geq s-1, \tilde{\boldsymbol{\theta}}) = \text{logit}^{-1} Z_s(\tilde{\boldsymbol{\theta}})$
- Define $Z_0() = 0$.
- $Z()$ does not need to have the same parameters for each level of s
- $Z()$ does not need to take the same functional form for each s
- $Z()$ can use a subset of the parent variables for each level of s
- Therefore, need to define structure function and parameters at the level of the state (except for the lowest state).
- Could switch between multiple-a and multiple-b patterns row-by-row.

Q-matrix part 1: 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		

Q-matrix part 1 (meta-data)

General description of the conditional probability table.

Model*	Name of model
NodeName*	Name of the node
Nstates	Number of states
State	Name of state. No row is needed for lowest state.
Link	Function used to compute conditional probability tables from $Z_s(\tilde{\boldsymbol{\theta}})$. Current choices are `partialCredit`, `gradedResponse` and `normalLink`.
LinkScale	Used for the normal link function (residual variance).

* Key for identifying a node.

Q-matrix part 2: Structure

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

Q-matrix part 2 (Structure)

This part describes the structure of the graph.

Model* Name of model

NodeName* Name of the node

Q-matrix One column per proficiency model variable. 1 if the variable is a parent, 0 if not.

Rules Functional form for $Z_s(\tilde{\boldsymbol{\theta}})$.

- For the `TwoStepObs` the parents are different for each transition: parents in graph are union of rows corresponding to child.
- Mix of multiple-a and multiple-b rules, even within one variable.

Q-matrix part 3: 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	

Q-matrix part 3 (Parameters)

This part provides the parameter values

Model*	Name of model
NodeName*	Name of the node
A-matrix	One column per proficiency model variable. Gives the value of a_k for multiple-a rules and the value of b_k for multiple-b rules.
B	Gives the value of b for multiple-a models and a for multiple-b models.
Weight	Prior weight for parameter learning algorithm

- Redundancy between Q-matrix (part 2) and A-matrix (part 3).
- Overloaded meaning for A matrix and B column.

Describing the Competency Model

- 2-matrix paper described starting from correlation matrix Ω and deriving proficiency model.
- Physics Playground approach: start with model structure, then add parameters.
- Use discretized regression (normal link function) models.
- Normal link-functions: these need link-scale function (residual variances).

Ω-matrix LHS: 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

Ω-matrix RHS: Parameters

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 Notes

- Nodes appear as both nodes and columns.
- Graphical Structure and A-matrix giving slopes are redundant.
- A-matrix has slopes, and residual standard deviations on the diagonal.
- B is a difficulty (negative intercept) not an intercept.
- Columns are provided for link function and combination rules, but other link functions require more parameters.

Peanut Software

- Available from `https://pluto.coe.fsu.edu/RNetica`.
- Peanut and CPTtools (builds CPTs) are Bayes net engine independent. Reference implementation (PNetica) uses (R)Netica as Bayes engine.
- Parameter and meta-data from Q and Ω matrix are attached as properties to parameterized net (Pnet) and parameterized node (Pnode) objects.
- Code written to go back and forth between 4 tables and parameterized nets and nodes, but not yet fully tested or documented.

Limitations

- Overloading of multiple-a and multiple-b modes is poor design, need to fix.
- Currently forcing the Ω matrix to use the normal link.
- Currently no way to represent dependencies among observables within an evidence model (extra columns in the Q-matrix).
- Currently only tested with Netica as Bayes net engine.
- May need to reparameterize the Ω-matrix.

Thanks!

- ACED: National Science Foundation Grant No. 0313202. Val Shute, PI.
- *Physics Playground v. 1*: Bill & Melinda Gates Foundation, grant *Games as Learning/Assessment: Stealth Assessment* (#0PP1035331, Val Shute, PI)
- *Physics Playground v. 2*: National Science Foundation grant *DIP: Game-based Assessment and Support of STEM-related Competencies* (#1628937, Val Shute, PI).

- Game Demo: `http://www.empiricalgames.org/ppunity5/LE/1.2.8/demo.html`
- Level Editor: `http://www.empiricalgames.org/ppunity5/LE/1.3.1/demo.html`
- Peanut and RNetica `https://pluto.coe.fsu.edu/RNetica`