Research Report

# Bayesian Network Models for Local Dependence Among Observable Outcome Variables

Russell G. Almond

Joris Mulder

Lisa A. Hemat

Duanli Yan

Research & Development

December 2006
RR-06-36

# Bayesian Network Models for Local Dependence Among Observable Outcome Variables

Russell G. Almond

ETS, Princeton, NJ

Joris Mulder

University of Twente, The Netherlands

Lisa A. Hemat and Duanli Yan

ETS, Princeton, NJ

December 2006

As part of its educational and social mission and in fulfilling the organization's nonprofit charter and bylaws, ETS has and continues to learn from and also to lead research that furthers educational and measurement research to advance quality and equity in education and assessment for all users of the organization's products and services.

ETS Research Reports provide preliminary and limited dissemination of ETS research prior to publication. To obtain a PDF or a print copy of a report, please visit:

http://www.ets.org/research/contact.html

Copyright © 2006 by Educational Testing Service. All rights reserved.

ETS and the ETS logo are registered trademarks of
Educational Testing Service (ETS).

## Abstract

Bayesian network models offer a large degree of flexibility for modeling dependence among observables (item outcome variables) from the same task that may be dependent. This paper explores four design patterns for modeling locally dependent observations from the same task:

- *No context*—Ignore dependence among observables.
- *Compensatory context*—Introduce a latent variable, context, to model task-specific knowledge and use a compensatory model to combine this with the relevant proficiencies.
- *Inhibitor context*—Introduce a latent variable, context, to model task-specific knowledge and use a inhibitor (threshold) model to combine this with the relevant proficiencies.
- *Compensatory cascading*—Model each observable as dependent on the previous one in sequence.

This paper explores these design patterns through experiments with simulated and real data. When the proficiency variable is categorical, a simple Mantel-Haenszel procedure can test for local dependence. Although local dependence can cause problems in the calibration, if the models based on these design patterns are successfully calibrated to data, all the design patterns appear to provide very similar inferences about the students. Based on these experiments, the simpler no context design pattern appears to be more stable than the compensatory context model, while not significantly affecting the classification accuracy of the assessment. The cascading design pattern seems to pick up on dependencies missed by the other models and should be explored with further research.

Key words: Bayesian networks, local item dependence, testlets, complex tasks, Mantel-Haenszel test

## Acknowledgments

Sections 3, 4, and 5 of this report are based on work that Joris Mulder did while a summer intern at ETS. An expanded description of those experiments is available in his final report.

We would like to thank the team doing the parallel IRT analysis, including Brad Moulder, Frank Jenkins, Bruce Kaplan, Quncai Zhou, and Youn-Hee Lim for generously sharing their results with us. Peggy Redman and Irv Katz helped by providing us with lots of the ECD documentation for the tasks used in the ICT Literacy Assessment. Alexander Matukhin did much of the programming for the StatShop software used in this analysis. Bob Mislevy and David Williamson provided helpful suggestions and cheerful encouragement throughout the project. Finally, Dan Eignor and Lou DiBello provided a number of suggestions that were quite helpful in improving the clarity of the presentation.

## 1 Introduction

Almost all latent variable models for testing assume that the observable outcome variables (the scored responses from items) are independent given the proficiency variables. Yen (1993) referred to this assumption as *local item independence* and listed a number of situations in which it breaks down. When all of the observable variables are conditionally independent given the proficiency variables, the model provides the maximum amount of information about the targeted proficiencies. Using the conditional independence model when it is not appropriate can overstate the precision of the estimate. Sireci, Thissen, and Wainer (1991) provided a number of examples in which the reliability of a test is overstated due to local dependence.

One situation that arises frequently in educational testing is a group of items that are bound by a common stimulus, such as a reading passage. Examinees who are familiar with the content domain of the passage are more likely to do well on all of the items. For example, if the passage was about dinosaurs, an examinee who had keen interest in dinosaurs as a child would already be familiar with difficult words such as *pterodactyl* or *Paleolithic*, while an examinee who lacked that background would need to decode the words and infer their definition from the context of the passage. It seems sensible that if the effect of the construct-irrelevant context could be removed from the observed outcomes, then the model would provide a better estimate of the underlying proficiencies.

The increasing use of computers in assessment is giving rise to a new source of local dependence: dependencies that arise from complex simulation tasks, such as the ones found in NetPass (Williamson, Bauer, Steinberg, Mislevy, & DeMark, 2004) or ETS's new ICT Literacy Assessment (Katz et al., 2004). Not only do these tasks rely on common stimulus material, but in many cases these multiple observables are based on common *work products* the examinee produces in the course of performing the task. For example, in a multistep problem, performance on the first step could greatly influence performance on the second or subsequent steps. Another common pattern is for a task to have two observables: one related to the correctness of the outcome and the other related to the efficiency of the path used to get there. These can be highly dependent as the best outcome may not be available unless the examinee takes a reasonably efficient path through the simulation.

Wainer and Kiely (1987) called groupings of observables tied together by common stimulus or work product *testlets* and suggested replacing all of the observables with a *super-item* (or *super-observable* in the evidence-centered design or ECD notation) representing the combined state, often replacing many small binary observables with a single partial credit observable. Following ECD notations (Mislevy, Steinberg, & Almond, 2003), we call the natural grouping of activities that share common stimulus or work product a *task* and note that a task may consist of a single item (with a corresponding observable outcome variable), multiple items, or a complex scoring rubric (either human or machine-scored) for setting multiple observed outcome variables on the basis of the work product the examinee produces for this task.

Three common design patterns are used to model local dependence. The first is to ignore it and hope that the approximation error introduced is not large. The second is to use the super-observable approach of Wainer and Kiely (1987). The third is to introduce additional latent variables to model a testlet effect (Bradlow, Wainer, & Wang, 1999). Taking Bayesian networks as our model for the assessment opens the door for new models for conditional dependence. Because the Bayesian network allows the modeler to specify arbitrary patterns of dependence among the variables, a wide variety of new approaches to representing local dependence are possible. This paper explores a few of them.

Section 2 reviews the Bayesian network models and some of the ECD notation used in this paper. Section 3 introduces the four design patterns investigated in this paper. Section 4 describes a simulation experiment to evaluate the robustness of the design patterns to model misspecification, and Section 5 looks at some tests for identifying local dependence. Section 6 looks at an experimental use of these design patterns in real assessment data. Section 7 draws some preliminary conclusions and makes some recommendation for future practice.

## 2 Bayesian Networks and Their Parameterization

Mislevy (1994) proposed using Bayesian networks as a way of representing the relationship between observable outcomes and proficiency variables in a diagnostic setting (i.e., a model with multiple proficiency variables). Because Bayesian networks can handle arbitrary patterns of dependence among observable outcome variables, they are particularly attractive for modeling local dependence. Section 2.1 reviews the basic notation of Bayesian networks. Almond and Mislevy (1999) suggested breaking a large Bayesian network into proficiency and evidence models. Section 2.2 reviews the ECD notation for the parts of the assessment. Section 2.3 looks at the way we are parameterizing the conditional probability tables that drive the Bayesian network.

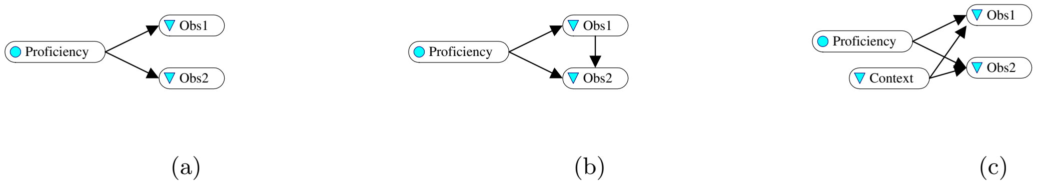


***Figure* 1 Local dependence and independence.**
***Note.* In the first graph, the proficiency variable separates Obs1 and Obs2 so they are conditionally independent. In the other two graphs, the proficiency variable does not separate Obs1 and Obs2, so the conditional independence condition necessary for local independence does not hold.**

### *2.1 A Brief Introduction to Bayesian Networks*

A Bayesian network model (see Pearl, 1988, or Almond, Mislevy, Steinberg, Williamson, & Yan, in press, for a complete description) is a compact representation of the joint probability distribution of all the (observable and latent) variables in the model. It is presented as an acyclic directed graph where the nodes represent the variables and the directed edges specify the dependency relationships between the variables. All variables in a Bayes net are discrete.

A key property of a Bayesian network model is that separation in the graph implies conditional independence of the variables (see Pearl, 1988, for the exact definition of separation). For example, in Figure 1 the proficiency variable separates Obs1 and Obs2 in the graph so the variables are conditionally independent. In Figure 1b, they are not conditionally independent because an edge directly connects Obs1 and Obs2. In Figure 1c, a path connects Obs1 and Obs2 through the context variable, so the observations still exhibit local dependence after conditioning on the proficiency variable. Note that in this latter model, conditioning on both proficiency and context variables renders the observations independent. The local dependence properties of the graphical model lead to efficient algorithms for processing them (the complexity of the computations is linear in the number of nodes, but increases exponentially as the graphs become more connected).

Associated with each variable in the Bayesian network is a conditional probability table giving its distribution. The probability of the child variable (the one pointed to by the arrow) is given conditioned on the value of its parent variables (those with arrows pointing into the child variable). If a variable has no parents, then the probability table is unconditional. The joint probability distribution over all the variables is the product of these tables. (See Almond, 1995, for a complete description of Bayesian network representations.) Section 2.3 addresses how to parameterize the tables.

### *2.2 Evidence-Centered Design Framework*

The models associated with an assessment can grow quite large, especially for an adaptive assessment in which observable variables must be added for each task that could be potentially scheduled. To make computations more modular, Almond and Mislevy (1999) suggested partitioning the complete Bayesian network model into a proficiency (or student) model that contains the variables representing the knowledge, skills, and abilities measured by the test and a collection of evidence models (one for each task) that describe how the observable outcomes from the task relate to the proficiency variables. The latter are properly Bayesian network fragments because they don't contain the proficiency variables. Rather, they contain stubs or references to the proficiency variables, which are resolved when the evidence model fragment is docked with the proficiency model to update the information about proficiency variables in light of the observed outcomes.

Note that the Almond and Mislevy (1999) procedure involves its own form of local dependence assumption. It assumes that variables from the evidence model corresponding to one task are independent of variables from the evidence model for a different task given the proficiency variables. This independence is assumed for both the observable outcome variables and any latent variables local to the evidence model (like the context variable in the models below). However, when using the Bayes net model, one has complete freedom to model local dependence between variables within a single evidence model, which can contain an arbitrary number of observable outcome variables. For example, a reading passage and a number of follow-up items can be grouped into a single task, and each item is associated with an observable in the evidence model for that task. Dependence among observables within a task can be modeled in a variety of ways, but dependence among observables from different tasks can only be modeled through the proficiency variables.

This paper follows the ECD tradition of using the terms *tasks* and *observables* (observable outcome variables) rather than *items*. A task could represent either a single discrete item, a more complex bundle of items (a testlet), a single constructed-response item that is given multiple scores (trait scoring), or a more complex pattern of observables extracted from a simulation task. In ECD notation, *observable* is the equivalent of the more familiar *item*, but it emphasizes the fact that the item may be embedded in a larger structure that may have local dependence.

A complete description of ECD can be found in Mislevy et al. (2003). The complete conceptual assessment framework contains other elements that will be used only incidentally in this paper. These include *task models*—which are used to describe families of related tasks—the *assembly model*—which describes the rules for establishing a valid form of the assessment—and the *rules of evidence*—which describe how the observable outcome variables are set on the basis of the *work products*.

### *2.3 Parameterization of Bayes Net Conditional Probability Tables*

The bulk of the work in building a Bayesian network model consists of eliciting a conditional probability table (CPT) for each variable in the model. (Even if the model is to be calibrated from data, a prior distribution for each CPT is needed.) For that reason, Almond et al. (2001) introduced a set of reduced parameter distributions based on Samejima's graded response model. These models are based on a suggestion by Lou DiBello to assign each state of a parent variable an *effective* $\theta$—a value on a unit normal scale. Once the parent states are mapped to the effective $\theta$ value, models from item response theory (IRT) can be pressed into service to construct the conditional probability tables.

The DiBello effective theta method proceeds in three steps:

1. For each input variable, select a real number, $\theta_k$, to serve as an effective theta for that proficiency. This is often done based on quantiles of the normal distribution. If the parent variable has three states, this works out to approximately -1, 0, and 1 for the low, middle, and high states, respectively.

2. The inputs each represent a separate dimension. Combine the dimensions into an effective theta for the task using a function $\tilde{\theta} = g(\theta_1, \ldots, \theta_K)$, called a *structure function.* Note that there are a number of different ways to do this step to model compensatory (Equation 1), conjunctive, and inhibitor (Equation 2) relationships.

   The most common choice for the combination function is the compensatory model, which implies having more of one skill will compensate for having less of another. The combination function is basically averaging

$$g(\theta_1, \ldots, \theta_K) = \sum_{k=1}^{K} \frac{\alpha_k}{\sqrt{K}} \theta_k - \beta \ , \tag{1}$$

where $K$ is the number of input variables, $\alpha_k$ is a discrimination parameter for the observable, and $\beta$ is a difficulty parameter. In general, $\alpha_k$ and $\beta$ must either be specified by content experts of fit to pretest data through a calibration procedure. The model of Equation 1 is roughly equivalent to the compensatory model frequently used in multivariate IRT.

An alternative is the inhibitor model, in which a student has to exceed a certain threshold of ability before the main skill comes into play. A classic example is a math word problem in which a student's language skill must be sufficient to understand what is required in the problem before the math skills come into play. This situation is almost always modeled with two parent variables and has the following structure function:

$$g(\theta_1, \theta_2) = \begin{cases} \alpha_2\theta_2 - \beta & \text{for } \theta_1 \geq \theta_{1r}, \\ \alpha_2\theta_{2,0} - \beta & \text{otherwise,} \end{cases} \tag{2}$$

where $\theta_{2,0}$ is the lowest possible value for $\theta_2$. As with the compensatory model, $\alpha_2$ and $\beta$ are either specified by experts or estimated from pretest data.

3. Use the effective theta value to calculate probabilities for the dependent variable. DiBello originally proposed using Samejima's graded response model (Samejima, 1969) to determine the probabilities of observables. This model is based on a series of curves for the observable value exceeding a given threshold:

$$\mathrm{P}(X \geq x_m|\theta) = P^*_m(\theta) = \mathrm{logit}^{-1}(\theta - d_m). \tag{3}$$

The conditional probabilities are created through differences in these probabilities curves.

Constructing a conditional probability table with this method is straightforward. Each row of the table corresponds to a configuration of the parent variables, which in turn corresponds to a set of effective theta values. This can be pushed through the structure function and graded response model to get probabilities for each possible observable state.

A key feature of this class of models is the structure function (Step 2). The choice of this function dictates the type of relationship (e.g., sum for compensatory, min for conjunctive). In typical modeling situations, the experts provide not only which variables are parents of a given variable but also what is the type of the relationship. In the figures below, distributions are given an explicit icon in the graph, and the shape of the icon is based on the nature of the relationship.

## 3 Design Patterns for Tasks With Conditional Dependence

As stated above, the Bayesian network model together with the ECD framework provides considerable freedom in how to model the dependence among observable outcome variables. This section suggests four possible design patterns for capturing different kinds of dependence.

One possible design pattern for the local dependence of observables within a task is to simply ignore the dependence and assume that the observables within a task are conditionally independent given the proficiency variables. In addition to this default design pattern, this paper studies three additional possible patterns. These are described briefly below:

- *Independence/no context design pattern.* In this design pattern, we simply assume that the effects of local dependence on the inferences are minimal and fit an approximate model that assumes the observables are independent given the proficiency variables. Often the task can be designed to minimize the dependence between observables. Also, observables can be dropped or merged to make this approximation work better.

- *Compensatory context.* In this design pattern, we introduce the context variable as an additional parent variable for each observable. We model the relationship between the context variable, the targeted proficiency variable,s and the observable variable using a compensatory distribution. This model has the effect of decreasing the weight of evidence (the amount by which the probabilities change in response to evidence, see Section 3.1). Joint evidence from the observables using the compensatory context effect model is always less than from the independence model. However, the context variable and the targeted proficiency variable have a very similar effect, which can make estimating the parameters of this model difficult.

  We call the introduced variable *context* because we think of it as familiarity with the context of the task; however, the variable is never reported and is really just a mathematical convenience for capturing the local dependence between observables in a task. To see how this works,

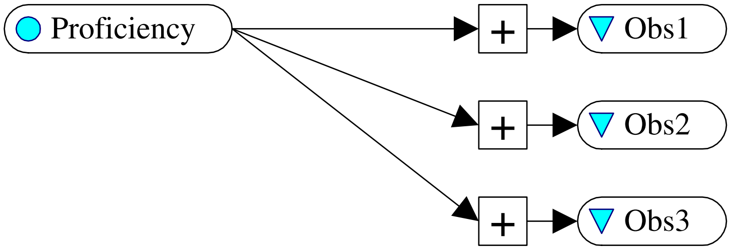


*Figure* **2 No context design pattern.**

***Note.*** **Nodes with circles represent proficiency variables common to many evidence models across all tasks. Nodes with triangles represent variables local to a particular task. The square boxes with + indicate that the relationship is modeled with a compensatory distribution.**

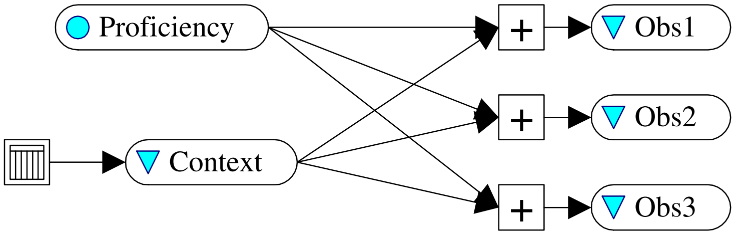


*Figure* **3 Compensatory context design pattern.**

***Note.*** **The context variable is local to the task (triangle) and is given a multinomial distribution. The relationship with the observable variables is a compensatory one.**

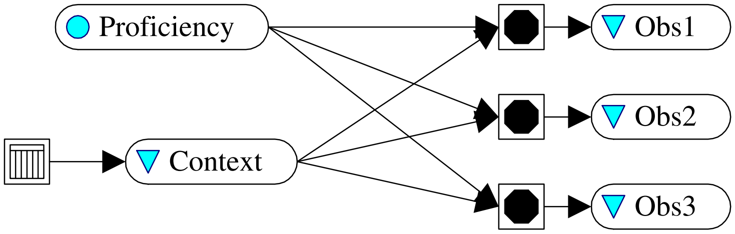


***Figure* 4 Inhibitor context design pattern.**

***Note.* The context variable is local to the task (triangle) and is given a multinomial distribution. The relationship with the observable variables is an inhibitor one (marked with a stop sign).**

consider two kinds of examinees both at the same level of proficiency, but one with a high value for the context variable and one with a low value. According to the model, the observable outcomes are independent given the proficiency level and context, but the first examinee will have a higher probability of success for each observable in the evidence model. For a given examinee, the true state of the context variable is unknown. Now, if a successful outcome is observed for the first observable, then the posterior probability of the context variable being high will increase. This will increase the predicted probability that the second (or later) observables will have a successful value.

Thus, the context variable introduces a local dependence among the observables within the evidence model (even after conditioning on proficiencies). Feller (1968) called this phenomenon *spurious contagion* and it is related to Simpson's paradox. The latent context variable is never observed and we do not report inferences about this variable. Instead, it is exploited to build a model for the conditional probabilities of the observables given the proficiency that has a pattern of dependence that matches the expert's expectations for this task.

- *Inhibitor context.* This design pattern also uses a latent context variable, but now the relationship is modeled with an inhibitor distribution. This model is particularly apt for simulation tasks with complex instructions. Unless the examinee understood the instructions, the examinee is likely to do poorly on all observable outcomes. However, once the instructions are understood, the outcomes depend on the measured proficiencies.

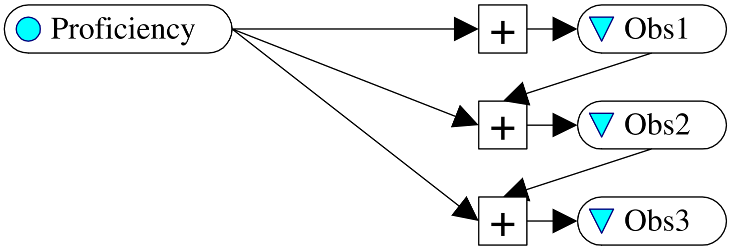


***Figure* 5 Cascading design pattern.**

***Note.* The observables are ordered and dependencies are introduced between them. In this case, the relationships are modeled with compensatory distributions.**

- *Cascading.* In this design pattern, no context variable is introduced. Instead, edges are forced between observable variables that exhibit local dependences (conditionally dependent given the proficiency variables). In the experiments below, the relationship between the pairs of observables and the proficiency variable are modeled with compensatory distribution type. Other distribution types are possible: for example, the inhibitor distribution or a more general conditional multinomial distribution (each row of the probability table is an independent multinomial distribution), possibly with zeros in the table to indicate functional patterns of dependence between observables (e.g., those introduced by scoring rules). This model is interesting for multistep tasks, where each observable corresponds to a step in sequence. Logically, performance on the second step should be influenced by performance on the first.

### *3.1 A Look at the Design Patterns in Action*

To provide a feel for the behavior of these design patterns, we built four simple Bayes nets using each design pattern. In each case, the proficiency variable had three ordered levels: above (basic), basic, and below (basic). Each case also had three observable outcome variables, each of which could take on a value of correct (1) or incorrect (0). The proficiency variable was given a uniform prior distribution to make the effect of the evidence model more visible. The evidence model parameters were chosen to be similar, but no attempt has been made (for this experiment) to calibrate the models.

In Figure 6, each bar represents the posterior distribution of the proficiency variable after observing one of the eight possible outcome vectors (R000, all incorrect, through R111, all correct).

First note that except for the cascading pattern, the posteriors are symmetric with respect to which observables are correct/incorrect; only the total number matters. This is not surprising as identical parameters were used in the conditional probability tables for each observable. However, in the cascading design pattern, the observables are not treated symmetrically. In particular, the last observable tends to provide more evidence than the first.

Good (1985) derived the weight of evidence as a measure of the amount of information a piece of evidence, $E$, provides for a hypothesis, $H$. The *weight of evidence for $H$ vs $\overline{H}$* is:

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

Note that the weight of evidence is the difference between the prior and posterior log odds for the hypothesis. Good recommended taking the logarithms to base 10 and multiplying the result by 100. He calls the resulting units centibans.

In this simple example, there are three states of the proficiency variables, and hence three basic hypothesis: proficiency is above, proficiency is basic, and proficiency is below. Figure 7 shows the weight of evidence provided by each design pattern for each of those hypotheses. In this case, only four outcome vectors (each corresponding to a different number of correct outcomes) are used. Note that the compensatory context pattern, as expected, provides substantially less evidence than the no context pattern, with one small anomaly for response pattern 0,0,1. This is exactly the effect the compensatory context pattern is designed to achieve: to avoid overcounting the evidence from the three observables.

The inhibitor context pattern, however, looks different from the no context pattern only in the case where the person had incorrect responses to all outcomes. This is because it offers an alternative explanation for this case, that the person was working off-task the entire time.

## 4 The Effect of Calibration

Figures 6 and 7 show that the differences between the four design patterns in the weights of evidence (that is the information content of the task) are mainly in the scale and not the shape of the distribution. For example, multiplying the weights of evidence for the compensatory context design pattern in Figure 7 by a constant would yield weights of evidence almost identical to the ones from the no context design pattern. Calibrating the model—learning the model parameters from pretest data—should adjust the scale of the weights of evidence. Thus, it might be possible

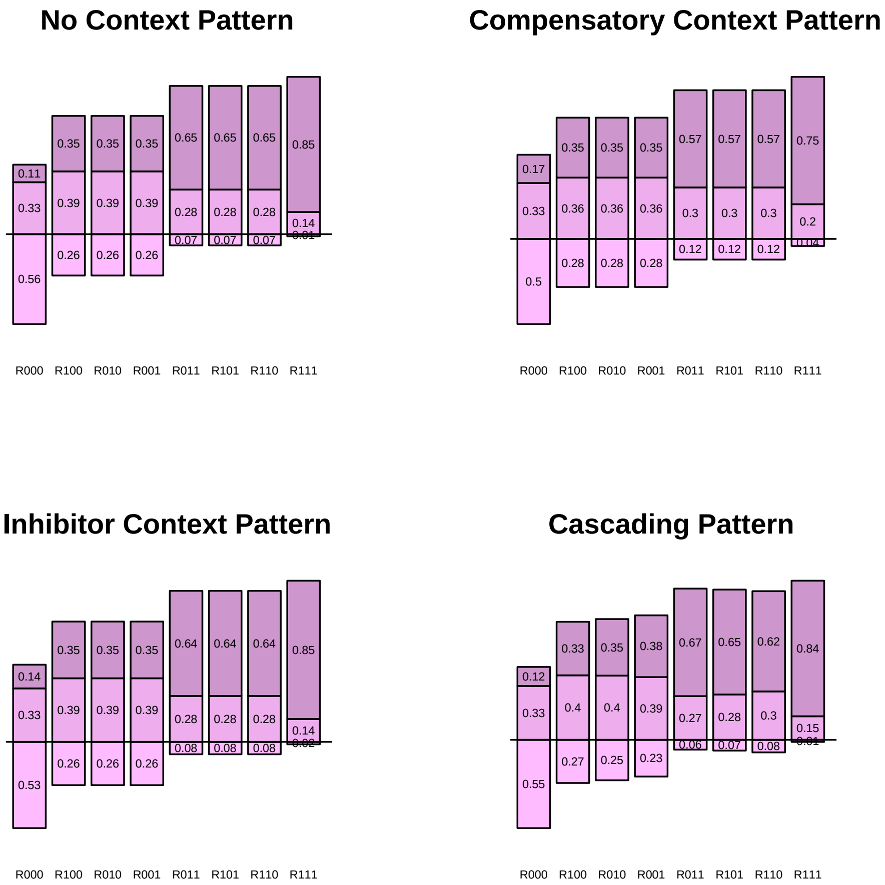


***Figure* 6 Posterior distributions by design pattern.**

***Note.* This shows the posterior distributions produced by each of eight possible outcome vectors (R000, all incorrect, through R111, all correct). Each bar shows the probability of the high, medium, and low state, respectively, for one outcome vector using the named design pattern. In each case, the prior was a uniform distribution over the three possible skill states.**

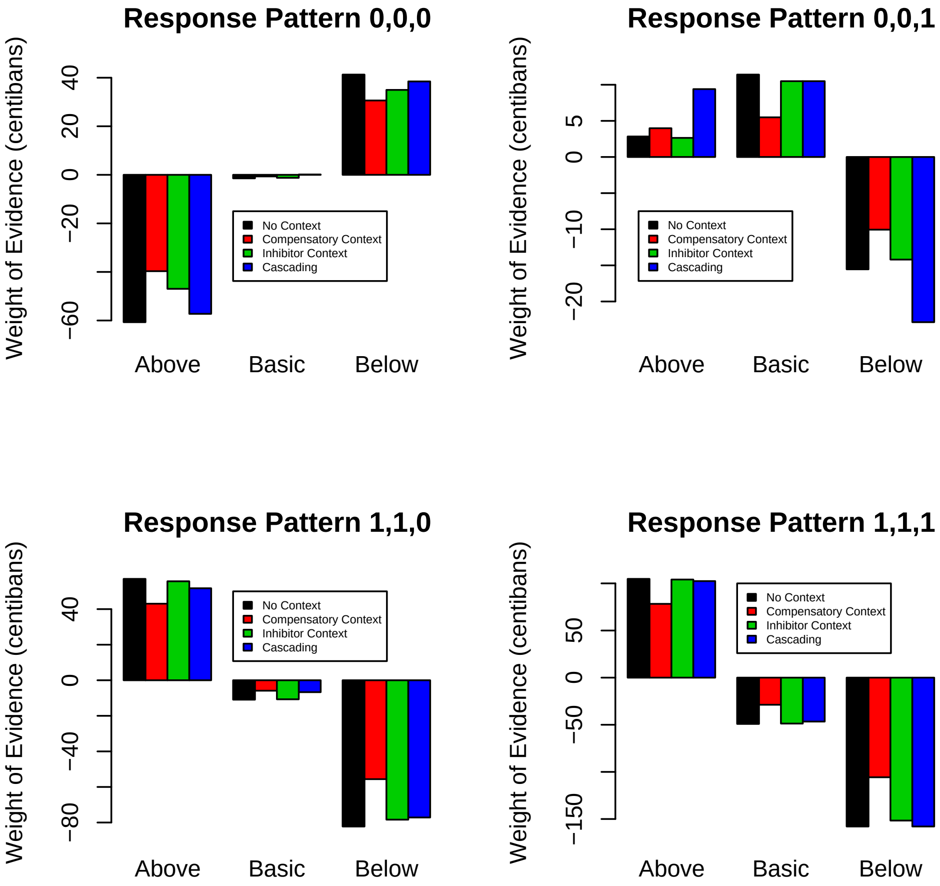


***Figure*** **7 Weight of evidence by design pattern.**

***Note.*** **This shows the weight of evidence for four selected design patterns. The grouped bars show the weight of evidence for the above (high), basic (medium), and below (low) states of the skill variable.**

to get a fairly good approximation to data from one design pattern using a model based on another design pattern if the parameters of the model were adjusted to better fit the data.

Each model drawn from any of the four design patterns forms a joint probability distribution over the proficiency variable and the observable outcome variables defined in the evidence model (three observable variables in our example). The goal of calibration is to find a set of parameters for a model from a different design pattern that best approximates the target joint distribution. Given that there is already a computer program (StatShop; Almond, Yan, Matukhin, & Chang, 2006) that would calibrate the model to data, the easiest way to find the closest approximating distribution was to first simulate data from the target distribution and then use the existing calibration algorithm to find the best approximation.

Normally calibration is challenging because the parameters of the evidence model and the unknown proficiency variables must be simultaneously estimated. The design of the test (primarily the amount of evidence available to estimate each proficiency variable) will affect the precision with which the proficiency can be estimated. To eliminate the effect of test design, we studied the effect of calibration in the rather artificial situation in which the proficiency variable is known during calibration. This should provide a realistic upper bound on how well a model from one design pattern can approximate data from a different design pattern in ideal circumstances.

We started with four data generation evidence models, one from each design pattern. Each data generation model tapped a single proficiency variable and had three observable outcome variables. The following procedure was used to find an approximate evidence model from each of the four design patterns (including the data generation pattern) for each generating design pattern:

1. We simulated 2,000 cases from each design pattern, including simulating the values of the proficiency variables.

2. We found a best-fitting model from each of the four design patterns using Markov chain Monte Carlo (MCMC) estimation, treating the proficiency variable as observed. (MCMC was used because the context variable was still latent for models that included it.)

3. We replicated the original models to make a test of seven tasks (21 observables) and generated data from each of those tests. (Replicates with identical evidence model parameters were used in place of a more realistic scenario of generating random parameters as this way the

approximation step, Step 2, would not need to be repeated for each evidence model.) The estimated evidence models were similarly replicated to produce four scoring models for each data set.

4. We scored the test with the data generation model and all four estimated models looking at the classification accuracy (comparing to the known simulated proficiency variables) in each case.

Note that the MCMC run for the compensatory context and inhibitor context design patterns took roughly twice the CPU time as the no context and cascading design patterns. This is because the MCMC algorithm must impute values for the context variables at each cycle of the MCMC loop, roughly doubling the workload. This can be a problem in large models (the ICT model described in Section 6 took 60 hours of run time on three machines).

Table 1 shows the results. The first surprise is that, for the purpose of classification, all the models seem robust to design pattern misspecification. In particular, the simplest no context pattern suffers only a small decrease in classification accuracy over the data generation model. When the model parameters are estimated from the data, the cascading design pattern provides the worst classification accuracy. This may be due to collinearity between the observable parent and the proficiency of interest.

There are two important cautions when interpreting these results. The first is that we treated the proficiency as known when we calibrated the model. In real life, that usually is not possible. Furthermore, if the local dependency is strong enough, it can bias the estimation of the proficiency variable, which must happen simultaneously with the estimation of the parameters. A strong local dependency among observables can cause model parameters (especially the discrimination/slope parameter) to blow up. This phenomenon is known in factor analysis as a Heywood case (Robert Mislevy, private communication, July 2004).

The second caution is that the approximation is specific to a given configuration of proficiency and observable outcome variables. If the task is simplified by removing an observable variable or made more complex by adding another observable, then the approximation may no longer hold.

**Table 1**

***Classification Accuracy for Different Design Patterns***

| | Estimated design pattern | | | | |
|---|---|---|---|---|---|
| Data | Cascading | Compensatory | Inhibitor | No context | Data generation |
| Cascading | 0.7715 | 0.7930 | 0.7615 | 0.7930 | 0.8005 |
| Compensatory | 0.7350 | 0.7370 | 0.7285 | 0.7345 | 0.7355 |
| Inhibitor | 0.8125 | 0.8390 | 0.8555 | 0.8300 | 0.8585 |
| No context | 0.8160 | 0.8780 | 0.8830 | 0.8975 | 0.9010 |

*Note.* Each row in the table represents a data set generated using one of the four design patterns. Each column represents a model using a different design pattern with parameters chosen to approximate design pattern in the column that is used to score the assessment. In the last column, the data generation model is used to score the assessment. The numbers give the classification accuracy of the maximum a posteriori (MAP) estimate of proficiency.

## 5 Testing for Local Dependence

Having established a number of possible models for how local dependence could arise, we would like to know how well we can detect its presence. Chen and Thissen (1997) presented some indices for measuring local dependence in the context of IRT models that model the latent state with continuous variables. For Bayes nets, an index that accounts for the discrete nature of the proficiency is needed.

We have developed two indices both based on the same principle. First, we take a data set and score it with our best estimate for the current model (in the case of the simulation experiment below, this is the data generation model). We use this model to produce a MAP estimate for the latent proficiency variables for each subject. We can now produce a three-way contingency table containing two observable outcomes from the same task and the proficiency variable they are trying to measure.

Two tests can be performed on this three-way table. The first is to simply compare the fit of the model that assumes that the two observables are independent given the imputed values for the latent proficiency variable to the staturated model with no indepedendence assumption. The second is to look at the Mantel-Haenszel test for consistency between the strata defined by the levels of the proficiency variable. Both tests are described in Bishop, Fienberg, and Holland (1975).

Unfortunately, one key variable needed for this test, the proficiency variable, is unobservable in real data (although it can be observed in simulation experiments). Practically speaking, the proficiency variable must be estimated from data. There are two choices: (a) use the MAP estimate from the posterior distribution after observing all outcomes from a test, or (b) use the marginal distribution of the proficiency variable as weights. Thus, if the posterior distribution over the skill was .5, .3, .2 for above, basic, and below, and this student scored correct on both observables under consideration, then we would add .5 to the cell corresponding to $(above, correct, correct)$, .3 to the cell corresponding to $(basic, correct, correct)$, and so forth, summing the values over all students in the data set. The resulting table would be a table of weights rather than counts.

Because three observable outcome variables (i.e., three items) make a very short test, we replicated the basic design pattern six times to make a short assessment with seven evidence models (corresponding to seven tasks) for a total of 21 observables (these are essentially the data generation models used in Section 4). The replicated evidence models had parameters identical to the initial evidence model illustrating the design pattern. This is slightly unrealistic, but it should not have a strong effect on our purpose (to illustrate the use of the test statistic). We then generated 2,000 simulees from each assessment and scored the assessment using the data generation model. Call the observables from the first task Obs1, Obs2, and Obs3. We calculated the Mantel-Haenszel statistic for the pairs Obs1 and Obs2 and Obs2 and Obs3 using the two methods for estimating proficiency described above. Because this was simulated data, we could also compute the Mantel-Haenszel statistic using the true value of the proficiency.

Table 3 shows the results. Unsurprisingly, the no context pattern does not reject the null hypothesis of no local dependence. Surprisingly, the value of the statistic for the compensatory context pattern is low. This suggests that whatever effect the context variable is having, it does not look like a violation of the local dependence assumption to the Mantel-Haenszel test.

To understand what these tests are detecting, look at the effect of each of the design patterns on the two-by-two table for the two observable variables at a given proficiency level (one slice of the three-way table). Under the no context design pattern, this table should be symmetrically distributed around the main diagonal. The inhibitor design pattern puts a special weight on the cell corresponding to the lowest value of each of the observables. The cascading design pattern puts

**Table 2**

***Mantel-Haenszel Test Statistics for Data From Various Patterns***

| Design | Obs1 and Obs2 | | | Obs1 and Obs2 | | |
|---|---|---|---|---|---|---|
| pattern | MAP | Margin | True | MAP | Margin | True |
| No context | 6.00 | 5.19 | 5.18 | 6.69 | 7.28 | 7.55 |
| Compensatory | 1.21 | 2.73 | 3.28 | 5.15 | 3.90 | 3.66 |
| Inhibitor | 69.92 | 78.33 | 76.73 | 50.80 | 56.54 | 52.62 |
| Cascading | 16.68 | 24.55 | 24.55 | 44.87 | 50.01 | 49.63 |

*Note.* These statistics should nominally follow a $\chi^2$ distribution with 4 degrees of freedom. The critical value for the test is 9.49.

more weight on the lower triangle of the table, where the second observable value is lower than the first. The test statistics are sensitive to these kinds of deviations. The compensatory context pattern, on the other hand, concentrates more weight on the main diagonal (two observables agreeing with each other). The test statistics are not sensitive to this kind of departure from the independence model, although this kind of departure is bad from the perspective of overstating the amount of information in the test.

## 6 ICT Literacy Assessment

We performed some exploratory analyses of the 2005 ICT Literacy Assessment data (Katz et al., 2004) to see if we could model the local dependence in the simulation tasks used in that assessment. The ICT Literacy Assessment was entirely composed of short simulation tasks with multiple observables per task, with automated scoring used for every task. Consequently, it seemed an ideal test bed for these design patterns. Although eventually a different method was chosen to score the assessment, these analyses are still very instructive as to how the design patterns play out in practice.

Section 6.1 gives a brief overview of the assessment design. Section 6.2 describes a couple of tasks showing clear examples of the kinds of local dependence that arise in simulation-based assessment. Section 6.3 looks into a couple of instances in which complex scoring rules yield *functional dependence* between the observable outcomes. Two models were used in our study, one using the no context design pattern and one using the compensatory context design pattern. Section 6.4 describes experiments that try to fit those two models to the 2005 data.

### *6.1 Assessment and Form Design*

Working with a committee of experts, the ICT design team (Katz et al., 2004) identified seven proficiencies related to information communications technology. They designed a pool of 48 tasks around those proficiencies. Of those tasks, 39 were short tasks that tapped only a single proficiency and had an average of 3 observables each. Each short task was supposed to take approximately 4 minutes to complete. Of the remaining tasks, 3 were long tasks tapping four of the seven proficiencies, averaging 12 observables and taking 30 minutes to complete; and 6 were medium length tasks (15 minutes) involving two proficiencies and about 5 observables. One form of the assessment would be administered in two separately timed blocks of 1 hour each; the first consisting of only short tasks and the second containing two medium length tasks and a long task.

The 2005 administration of the ICT Literacy Assessment was designed to be a low stakes survey reporting only at the group level. Consequently, the short tasks in the pool were rearranged into forms that each concentrated on two of the seven proficiencies. The medium and long tasks were paired with the forms to attempt to get optimal information about the two proficiencies on the form (Almond, Yan, & Hemat, 2006, gives a more thorough description of the model and prior). The test was administered in the winter and spring of 2005, and just under 5,000 students were selected from the universities that had chosen to participate in the study.

Even though each form yields approximately 60 observable outcome variables, there were questions about whether that was enough information to distinguish seven distinct proficiencies. For that reason, the ICT design team chose three pairs of proficiencies to merge. The resulting model had four proficiency variables on which the assessment would report. This four-proficiency model was the basis of all of the analyses described below.

There is a certain amount of indeterminacy between the scale and location of the distribution of the proficiency variables, and the average difficulty and discrimination parameters. In order to remove that indeterminacy, *scale anchors* (Almond, Mislevy, & Yan, 2005) were used to anchor each of the four scales. These consisted of a set of observables (in this case, all of the observables from the simple tasks) for which the sum of the difficulties was constrained to be 0, and the product of the discriminations was constrained to be 1.

In addition to our Bayes net analysis, another group (Jenkins & Qian, 2005, April) performed an IRT scaling on the same data set. We had access to the results of both their item analysis and the scaling. In several cases, the IRT team decided to drop certain items or to collapse certain categories on the basis of both item analysis and difficulties achieving convergence in the IRT scaling (in particular, very large discrimination or difficulty parameters). For the most part, we adopted changes to the model made by the IRT team and made the corresponding changes in the Bayes net model.

The final model had five proficiency variables (including an unreported ICT commonality variable used to model dependence among the reported proficiency variables) and 234 observables (including 2 that were functionally dependent on other observables) distributed among 48 tasks. Compare this to the original target of 186 observables. Furthermore, the number of observables per task were distributed in a nonuniform fashion, so that the ECD model description was necessary to keep track of what went where. In total there were 782 parameters, 37 of which were proficiency model parameters (conditional probability tables), and the remaining 745 were evidence model parameters (primarily difficulty, discrimination, and difficulty increment parameters for each observable). The complete model description is available in XML and HMTL formats in Almond and Hemat (2005).

### *6.2 Examples of Local Dependence*

A brief description of a couple of example tasks will help describe how local dependence arises in simulation-based assessments. We looked closely at the rules of evidence (scoring rules) for two tasks in detail. These show how the fact that the observables arise from the same work product leads naturally to the dependence.

The first was a short task in which a dialogue was simulated through a series of multiple-choice questions. At the end of the dialogue, the examinee was supposed to make a recommendation about what to do next. The rules of evidence called for three observable variables: (a) the quality of the final recommendation, (b) the quality of the path through the dialogue, and (c) the quality of the first step on the path. Obviously, there is a great deal of dependence between Observable 2 and Observable 3 in this model. Furthermore, because the examinee would not reach the best final recommendation unless he or she took a near optimal path through the dialogue, there was also a great deal of local dependence between Observable 1 and Observable 3. For this task, Observable 3 was dropped from both the IRT and Bayes net models.

The second was a long task that called for the examinee to make a presentation for two different audiences. Three of the observables were (a) the quality of the first presentation, (b) the quality of the second presentation, and (c) whether the presentations were appropriately adapted for the two audiences. This is also an interesting pattern because although any pair of observables are not highly dependent, all three are. The third observable was particularly important as one of the experts had identified this as a critical part of the communicate proficiency. Although dropping any one of the three would solve the technical problem, the three variables do not have the same value to the content experts. As always, the decision of which variables to drop must involve both statistical and content experts.

To assess the extent of the local dependence in these data, we performed the Mantel–Haenszel test described above for each pair of observables from the same task. To get the proficiency estimates, we scored the data with an uncalibrated model. (These may not be ideal estimators of the proficiency variables, but they should be at least correlated with the right variables.) The Mantel–Haenszel statistic was significantly large for 80% pairs (369 out of 459 pairs; significance was taken based on the 0.05% point of the $\chi^2$ distribution). This indicates that departures from the local dependence assumption are widespread, although it does not say to what extent they will interfere with estimation of proficiency.

### *6.3 Functional Dependence*

In a few cases, the scoring rules implied that certain combinations of observable values were logically impossible. For example, the requirements for getting a high score on Observable 2 might include all of the requirements for getting a high score on Observable 1. We called these cases *functional dependence* and made a special effort to model them. (There were four cases of functional dependence in the 2005 ICT Literacy Assessment.)

Cases of functional dependence were modeled using the cascading model in which the conditional probability table for the second observable had a hyper-Dirichlet distribution (an independent multinomial distribution for each row of the table with a corresponding collection of Dirichlet laws for the parameters). The functional dependence still causes problems as the logical impossibilities implies zero cells in the tables and zero hyperparameters in the Dirichlet distribution. Although the hyperparamaters of the Dirichlet law must be positive, StatShop software (Almond et al., 2006) was used to fit the model relied on the convention that a zero prior probability meant that a zero probability should be assigned to that cell in the multinomial distribution. Thus, zeros in the prior hyperparameter table could be used to mark combinations of observables that are logically impossible.

Some caution is necessary in the definition of logically impossible. The member of the design team who had the task of flagging observable pairs with functional dependence originally flagged several pairs because no reasonable person would score a high on the second observable without scoring a high on the first. Sure enough, in the pilot test data (sample size of about 100) two individuals had the impossible score pattern, resulting in zero likelihoods and serious problems for the scoring algorithm.

The hyper-Dirichlet models proved to be difficult to fit. Almond et al. (2006) noted an identifiability issue associated with hyper-Dirichlet models in which the posterior distribution had multiple modes, with rows of the table swapped (the average table from Chain 3 for the MCMC simulation looked like the average table from Chain 2 with the second and third rows reversed). This is a problem related to label identifiability in classical latent class models. The original model included four pairs of observables with functional dependence (and hence modeled with hyper-Dirichlet distributions). This problem occurred in two of the four cases. For the purposes of this analysis, these two observables were dropped and the others remained. Note that the IRT team dropped all four observables. The row-swapping problem stems from the lack of any constraint in the hyper-Dirichlet model so that increasing values of the parent variable should produce probabilities for the outcome variables at least as high as the previous step. The constrained hyper-Dirichlet model proposed by Almond et al. (2006) is a possible solution for this problem.

### *6.4 Model Calibration Studies*

StatShop (Almond et al., 2006) uses Markov chain Monte Carlo to estimate model parameters from pretest data. The standard practice for using StatShop has evolved as follows:

1. Start three chains, one from the midpoint of the prior distribution for each parameter, one from the upper tail, and one from the lower tail.

2. Run each chain for length 6,000 and discard the first 1,000 as burn-in.

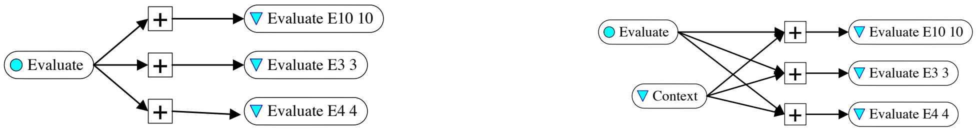
3. Assess convergence by calculating the Gelman-Rubin potential scale reduction Factor (Gelman & Rubin, 1992), often called the Gelman-Rubin $R$. This is calculated for the proficiency model parameters (because if they don't converge the evidence model parameters seldom converge either). If these look reasonable (Gelman and Rubin recommended less than 1.1 as heuristic value for reasonable), then parameters for each observable in each evidence model are checked, and those above 1.1 are flagged.

This procedure was applied to the same data using two different models. The first model was based on the no context design pattern, and each observable variable had only a single proficiency variable as a pattern (with the exception of the two cases of functional dependence retained in the model). The second model was based on the compensatory context design pattern, and the context variable was introduced into the evidence model for each of the 48 tasks. Each observable had two parents, the relevant proficiency variable and the context variable for its task. Note that in the 39 short tasks, only a single proficiency was tapped. Therefore, the structure was still approximately simple, although the context effect is aliased with the proficiency variable, which may cause identifiability issues (Figure 8).

For the medium and long tasks, the situation is a little bit more complicated. Each of these tasks tapped multiple proficiency variables, and the context variables induced dependencies among the proficiency variables that were not in the no context model (Figure 9). This also spoiled the simple structure. One additional complication arose with the longer tasks; there was some indication that the students (for whom the assessment outcome had low stakes) were not as strongly motivated for the longer tasks, which were typically scheduled during the second hour of testing.

When the no context model was fit using the MCMC algorithm, only 4 (out of 782) parameters had $R$ values above 1.1. These were the difficulty and discrimination parameters from two observables from Tasks E34 and E35 (both evaluate tasks). The value of $R$ for the discrimination parameter was 1.878 (Task E34) and 1.690 (Task E35). The value of $R$ for the difficulty parameters was 1.304 (Task E34) and 1.293 (Task E35).

Lack of convergence in the MCMC sampler often means that there are multiple modes in the posterior distribution and that the sampler is not mixing from one mode to the other (one chain gets stuck in one local maxima, and the other chain in a different local maxima). In this case,

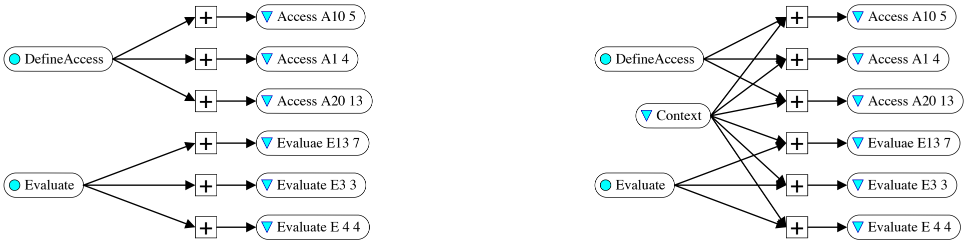


(a) No context EM

(b) Compensatory context EM

***Figure*** **8 Evidence models for short tasks.**

***Note.*** **This shows the evidence models for the two design patterns for the same short task. Other tasks differed as to the number of observable variables but had the same general structure. Note that context is local to this evidence model, while the proficiency variable is shared across all evaluate tasks.**



(a) No context EM

(b) Compensatory context EM

***Figure*** **9 Evidence models for medium length tasks.**

***Note.*** **This shows the evidence models for the two design patterns for the same medium length task. Other tasks differed as to the number of observables tapping each proficiency (these were not necessarily equal) but had the same general structure. Note that context is local to this evidence model, while the proficiency variables are shared across all tasks.**

the two parameters are in the same anchor set, and therefore their sum (along with the sum of other parameters from evaluate tasks) are constrained to be 0.0. Thus, what is happening in this data set is that one chain has a high discrimination for the first task and a lower discrimination for the second. This is reversed in another chain. The chains are not crossing between the two local maxima in the short chain length. The situation with the difficulty parameters is similar, although it could well be that there is a more complex four-way relationship.

It is possible that if the chain were run longer, the two chains would eventually mix; however, it is probably easier to fix the problem by dropping one or both observables from the assessment. (Just dropping them from the anchor set may be sufficient.) In general, the no context model could be successfully fit to the data. (A comparison of the results of this analysis to the parallel IRT analysis will be described in a future report.)

The second model used a context variable in each evidence model to model the local dependence. Compensatory distributions were used for all observables, making this model the compensatory context design pattern described above. The results from this analysis were substantially worse. The Gelman-Rubin $R$ value reached as high as 9 for some parameters in the proficiency model. This resulted in roughly half (446 of 993 parameters) of the parameters failing to converge. Running the model for an additional 10,000 iterations did not improve the convergence rate.

Recall that lack of convergence in this case means that there are multiple modes in the posterior distribution. The three chains (starting at different starting points) have found at least three of them. The marginal distributions for the proficiency variables in each chain tell a large part of the story. Table 3 gives the distributions by chain for the proficiency variable with the largest values of $R$. This is essentially the proportion of students estimated to fall in that category. Note that the proportion of students in the pretest data classified at the high level level differs by about 0.05 across the three chains. One disturbing explanation for the failure of the compensatory context pattern to converge is that different assumptions about the strength of the dependence in the tasks is leading to different conclusions about the presence of the skill in the population. This could happen because the relative sizes of the discrimination parameters for the evaluate proficiency and context latent variable are only weakly identified by the data.[1] Thus, one chain could have a higher discrimination for evaluate than for the context variable while in another chain the discriminations could be approximately equal. This could shift the population estimates for the evaluate variable in the two chains.

**Table 3**

***Marginal Distributions for the Evaluate Proficiency Variable***

| Chain | High | Medium | Low |
|---|---|---|---|
| Chain 1 | 0.314 | 0.420 | 0.270 |
| Chain 2 | 0.306 | 0.432 | 0.263 |
| Chain 3 | 0.261 | 0.421 | 0.316 |

Note that all tasks have a context variable, so that the collection of all context variables for evaluate tasks effectively alias (block) the evaluate proficiency variable. (The models in Section 4 did not have this problem because the proficiency was assumed to be known during estimation.) The model probably needs stronger constraints either through tighter priors or eliminating one or more of the context variables. However, this means that the final model is potentially quite sensitive to modeling choices.

The complex design of the assessment presents a number of other issues with these data that may also lead to problems with convergence. In particular, on five of the seven forms, the evaluate proficiency variable is measured through only two tasks; one long, and one medium in length. Thus, shifts on the parameters of those two tasks may have been enough to create shifts in the population distribution. There may also have been motivational differences in the shorter and longer tasks. (The subjects all knew that the test outcome would not affect their grades or academic standing.)

To sort some of these issues out, we generated data known to be consistent with the model. We estimated the parameters of the model from the last 5,000 cycles of just the first chain. We then generated 5,000 simulees from the distribution with these parameters. Finally, we tried to fit this using the same procedure and priors as the operational data.

The convergence statistics for this run were considerably better. Only 6% (58 of 993) parameters were flagged with high values of $R$. In the proficiency model, the highest value of $R$ was 1.124 (with 1.1 the threshold generally used for convergence). It seems that with the simulated data there are still multiple modes in the posterior; however, the problem is not nearly as pronounced as in the real data. It would be reasonable to conclude that the model is underspecified and that tighter priors or other assumptions about the nature of the context effect might be necessary to get the model to converge. The hierarchical models used by Bradlow et al. (1999) are one approach.

A large part of the issue is the symmetry between the context and the proficiency variable in the equation for the compensatory model. If there is a moderate degree of correlation between the context and the proficiency variable, this will result in collinearity between the two variables, producing a ridge in the likelihood (and hence the posterior). This could explain the observed difficulties with convergence.

One way to proceed would be to review each task and try and put a sensible prior on the strength of the context effect. This has not been done for a number of reasons. As mentioned previously, the ICT Literacy program has chosen a different method to use for scoring the assessment, one that should be less sensitive to assumptions about local dependence. Second, the long run time would make this process rather time consuming. Finally, the compensatory context model may not be the appropriate one. The cascading model seems more appropriate for the kinds of tasks described in Section 6.2. Task-by-task modeling for all 48 tasks would take a great deal of effort.

## 7 Preliminary Recommendations

The simple experiments here have only scratched the surface of what can be done with these kinds of models. There are still a lot of unanswered questions, especially about the interplay of models for local dependence with multiple proficiency variables. However, these experiments have shed some light into the use of the four design patterns.

- *No context.* This pattern is simpler than the others, and with calibration it provided a surprisingly good approximation to the other design patterns. This suggests that if the model is to be calibrated from data, the no context pattern may be good enough for practical application, provided the local dependence does not bias the calibration.

- *Compensatory context.* This pattern does the most towards damping down the evidence from dependent observables. However, it appears as if data from this pattern is easily approximated by other models. Furthermore, it enters into the equations symmetrically with the effect it is trying to measure; for that reason is seems to produce a great deal of collinearity, making the resulting models difficult to fit. Its role is probably more important when the model is to be built primarily from expert opinion, to help the experts not overcount the evidence.

- *Inhibitor context.* This pattern differs from the no context model only in the case where are observables are incorrect. Basically, it reduces the amount of negative evidence provided by the model when the examinee is off-task. This may provide better measurement, but it runs contrary to expectations to reward off-task behavior. In particular, it produces a dissonance between the notion of the test as measurement and the test as a contest.

- *Cascading pattern.* This is the most interesting of the patterns because it seems to be doing something substantially different from the other patterns. It would be particularly useful for multistep tasks or simulation tasks where two or more observables are based on the same work product. However, the compensatory distribution used with the cascading model in the examples may not be appropriate.

As an alternative to the use of the compensatory distribution in the cascading pattern, consider the inhibitor distribution. This seems particularly attractive in the case of a multistep problem. Examinees who get the first step correct are given a normal IRT-like model for the second step. Examinees who get the first step wrong behave like the lowest scoring group. A variant on this idea worth exploring is to use a mixture of two DiBello-Samejima models, one for examinees who get the first step correct, and one (with higher difficulty and lower discrimination) for examinees who get the first step incorrect. It remains to be seen whether this design pattern would exhibit the same kind of estimation problems that the compensatory context pattern had.

Given that the no context design pattern could do a good job of approximating the other design patterns with fewer parameters, our current recommendation is to use this model whenever the model will be calibrated to data, but to check for possible local dependence problems that might affect the results. One way to check is to use the Mantel-Haenszel statistic; however, another method is to perform a calibration (either IRT or Bayes net with the DiBello-Samejima model) and look for large values of the discrimination parameter for observables from the same task. These may indicate that local dependence has caused a problem with the estimation of the proficiency variables, which will, in turn, affect the estimation of all of the parameters in the model.

When these simple procedures indicate a lack of fit, more complex design patterns may be warranted. In the case where two locally dependent observables are measuring the same proficiency variables, it may be fairly easy to drop one or combine them into a single variable. In either case, the test developers should look carefully at the scoring rules as they may be able to make a combination that better fits the purposes of the assessment than the one created by fitting the model to data. This is essentially the testlet observable proposed by Wainer and Kiely (1987). If the task taps a single dimension, scoring using multiple observables may not add much for the purposes of evidence accumulation (however, it may be useful for the purposes of providing task level feedback).

If it is important to retain multiple locally dependent observables in a task, the cascading design pattern offers the best hope of realistically modeling the dependence. The compensatory context pattern, in addition to being difficult to estimate, does not seem to change the distribution in a way that produces high Mantel-Haenszel statistic values. However, using the cascading design pattern requires more work than the others as the experts must think hard about the nature of the relationship among the observables. ECD provides a mechanism for communicating these constraints across models.

The realization that psychometricians and test developers must think hard about models for local dependence and not just rely on off-the-shelf solutions may be the most important result of the study. In situations where local dependence is a potential issue, it is important for psychometricians and test developers to have good lines of communication about potential problems and possible solutions. The documentation required for the ECD process was very helpful in maintaining those lines of communication.

## References

Almond, R. G. (1995). *Graphical belief modeling.* London: Chapman and Hall.

Almond, R. G., Dibello, L., Jenkins, F., Mislevy, R. J., Senturk, D., Steinberg, L. S., & Yan, D. (2001). Models for conditional probability tables in educational assessment. In T. Jaakkola & T. Richardson (Eds.), *Artificial intelligence and statistics 2001* (pp. 137–143). San Mateo, CA: Morgan Kaufmann.

Almond, R. G., & Hemat, L. A. (2005). *ICT model repository.* Unpublished manuscript.

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. (in press). *Bayesian networks in educational assessment.* New York: Springer.

Almond, R. G., Mislevy, R. J., & Yan, D. (2005). *Using anchor sets to identify scale and location of latent variables.* (Manuscript in preparation)

Almond, R. G., Yan, D., & Hemat, L. A. (2006). *Simulation studies with a four proficiency Bayesian network model.* (Manuscript in preparation)

Almond, R. G., Yan, D., Matukhin, A., & Chang, D. (2006). *Statshop testing* (ETS Research Memorandum No. RM-06-05). Princeton, NJ: ETS.

Bishop, Y., Fienberg, S., & Holland, P. (1975). *Discrete multivariate analysis.* Cambridge, MA: M.I.T. Press.

Bradlow, E., Wainer, H., & Wang, X. (1999). A Bayesian random effects model for testlets. *Psychometrika*, *64*, 153–168.

Chen, W., & Thissen, D. (1997). Local dependence indices for item pairs using item response theory. *Journal of Educational and Behavioral Statistics*, *22*, 265–289.

Feller, W. (1968). *An introduction to probability theory and its applications* (3rd ed.). New York: Wiley.

Gelman, A., & Rubin, D. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, *7*, 457–511.

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

Jenkins, F., & Qian, J. (2005, April). *IRT analysis of the ICT Literacy Assessment.* Paper presented at the annual meeting of the National Council on Measurement in Education, San

Francisco.

Katz, I. R., Williamson, D. M., Nadelman, H. L., Kirsch, I., Almond, R. G., Cooper, P. L., & et al. (2004). *Assessing information and communications technology literacy for higher education.* Paper presented at the 30th Annual Conference of the International Association for Educational Assessment (IAEA), Philadelphia, PA.

Mislevy, R. (1994). Evidence and inference in educational assessment. *Psychometrika*, *12*, 341-369.

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.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference.* San Mateo, CA: Morgan Kaufmann.

Samejima, F. (1969). Estimation of latent ability using a response pattern of graded scores. *Psychometrika Monograph No. 17*, *34*(4), (Part 2).

Sireci, S. G., Thissen, D., & Wainer, H. (1991). On the reliability of testlet-based tests. *Journal of Educational Measurement*, *28*, 197–219.

Wainer, H., & Kiely, G. (1987). Item clusters and computerized adaptive testing: A case for testlets. *Journal of Educational Measurement*, *24*, 185–201.

Williamson, D., Bauer, M., Steinberg, L., Mislevy, R., & DeMark, S. (2004). Design rationale for a complex performance assessment. *International Journal of Testing*, *4*, 303–332.

Yen, W. (1993). Scaling performance assessments: Strategies for managing local item dependence. *Journal of Educational Measurement*, *30*, 187–213.

## Notes

[1] Technically, in a Bayesian model all parameters are identified by the prior distribution, if nothing else. However, Bayesian models that are not identifiable in the classical sense often have multiple modes in the posterior distribution, which make them difficult to estimate.