Models for Understanding Student Thinking using Data from Complex Computerized Science Tasks

by

Michelle Marie LaMar

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Education

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Sophia Rabe-Hesketh, Chair
Professor Mark Wilson
Associate Professor Tom Griffiths

Fall 2014

Models for Understanding Student Thinking using Data from Complex Computerized Science Tasks

Abstract

Models for Understanding Student Thinking using Data from Complex Computerized Science Tasks

by

Michelle Marie LaMar

Doctor of Philosophy in Education

University of California, Berkeley

Professor Sophia Rabe-Hesketh, Chair

The Next Generation Science Standards (NGSS Lead States, 2013) define performance targets which will require assessment tasks that can integrate discipline knowledge and cross-cutting ideas with the practices of science. Complex computerized tasks will likely play a large role in assessing these standards, but many questions remain about how best to make use of such tasks within a psychometric framework (National Research Council, 2014). This dissertation explores the use of a more extensive cognitive modeling approach, driven by the extra information contained in action data collected while students interact with complex computerized tasks. Three separate papers are included. In Chapter 2, a mixture IRT model is presented that simultaneously classifies student understanding of a task while measuring student ability within their class. The model is based on differentially scoring the subtask action data from a complex performance. Simulation studies show that both class membership and class-specific ability can be reasonably estimated given sufficient numbers of items and response alternatives. The model is then applied to empirical data from a food-web task, providing some evidence of feasibility and validity. Chapter 3 explores the potential of using a more complex cognitive model for assessment purposes. Borrowing from the cognitive science domain, student decisions within a strategic task are modeled with a Markov decision process. Psychometric properties of the model are explored and simulation studies report on parameter recovery within the context of a simple strategy game. In Chapter 4 the Markov decision process (MDP) measurement model is then applied to an educational game to explore the practical benefits and difficulties of using such a model with real world data. Estimates from the MDP model are found to correlate more strongly with posttest results than a partial-credit IRT model based on outcome data alone.

To Brandon, Henry and Geoff

Contents

List of Figures

List of Tables

Acknowledgments

First I'd like to thank my advisors. Sophia Rabe-Hesketh been an inspiration to me throughout my doctoral studies. She has provided me guidance and support, while demonstrating the type of open-minded curiosity which I hope to exemplify as a researcher myself. Sophia's incredible grasp of complexity and simultaneous attention to detail made our discussions ever challenging, exciting, and productive. Mark Wilson has taught me to look beyond the models to understand the effect they have upon the world. He has also been invaluable in pushing me to do my best work; asking me to show, not tell, and to prove, rather than assert. Meanwhile, Tom Griffiths has not only shown me the value of cognitive modeling, he also introduced me to the types of models that populate my dissertation. Further, thanks to him, I now believe that we are all really Bayesian.

I have been extremely fortunate to collaborate with Anna Rafferty. Her exploration into the use of Markov decision processes for modeling student actions within an educational game predates my work. Anna has been generous in sharing her insights and research direction with me. In addition, her quick mind and our shared interests have made our meetings, often over Thai food, some of the most stimulating conversations I've had in grad school.

My fellow students in QME have been a wonderful resource for me, particularly my cohort, Hyo Jeong Shin, Perman Gochyyev, and Rebecca Poon. Over the past five years, they have inspired, consoled and encouraged me. I would like to especially thank Ronli Diakow and David Torres Irribarra, both of whom were always willing to help out their fellow students, whether it was preparing for orals, interpreting complex models, or practicing for conference presentations. They also reminded me, that every once in a while, I should take a break and just have some fun.

The work presented in this dissertation would literally not have been possible without the valuable collaboration with my colleagues at WestEd. I have been fortunate through most of my doctoral years to work with WestEd STEM researchers, Michael Timms, Edys Quellmaltz, Barbara Buckley and Matt Silberglitt, who were already trailblazing in the area of complex computerized assessment. Their projects have provided me with the data that inspired the approaches I've taken in my research, and their perspectives have shown me new ways of envisioning the future of assessment.

Finally I'd like to thank my family. Without the love and support of my husband, Geoff, I would never have been able to start, not to mention complete, this work. Geoff has always encouraged me to dream big and to learn from my failures. My children, Henry and Brandon, have been patient and supportive throughout. From serving as willing test-subjects to debating the value of assessment in eduction, they have assisted me at each stage of my studies.

Chapter 1

Introduction

A strong argument can be made for the use of complex tasks in science assessment. Fundamentally, science is not a collection of facts, but instead a set of practices and a way of viewing the world. Central to the practice of science is inquiry and investigation; posing questions and seeking answers through principled scientific methods. The process of inquiry involves both multiple layers of knowledge and skills and multiple steps to complete. It has been found that testing isolated "inquiry skills" out of the experimental context of real science does not predict student ability in a realistic investigation (McElhaney & Linn, 2011).

The world-view of the scientist, meanwhile, is filled with dynamic models that are continually being tested and revised. Working with these models involves not only understanding their components, but also how the components interact to generate higher level phenomena. Complex tasks are able to tap student thinking about entire systems, getting at the mental models students bring to scientific problem solving (V. J. Shute et al., 2010; Buckley, Gobert, Horwitz, & O'Dwyer, 2010). Evidence centered design suggests that we select assessment tasks based on the evidence they can provide of the competencies we wish to assess (Behrens, Mislevy, DiCerbo, & Levy, 2012). Both for assessing the understanding of complex systems within science and for assessing student ability to engage productively in scientific investigation and reasoning, complex tasks are compelling in how they can provide direct evidence of competency.

Indeed, the recently published Next Generation Science Standards (NGSS, NGSS Lead States, 2013) call for the integration of discipline knowledge and cross-cutting ideas with the practices of science. They explicitly define performance targets that require students to be able to simultaneously demonstrate both science knowledge and science practices. These standards have strong implications for how science should be assessed. In their recent report *Developing Assessments for the Next Generation Science Standards*, the National Research Council (2014) points out that the NGSS requires assessment of three dimensions of science learning, not just within the assessment, but within a individual tasks. They suggest that effective assessments will contain a variety of different item types, with multi-component performance tasks playing a central role. Their selected task examples feature extended activities in which students carry out an investigation, construct an argument or create a representation.

Using complex tasks for assessment is not a new idea, and debate about their use under a variety of names, such as performance assessment and authentic assessment, has been ongoing for decades (Linn, Baker, & Dunbar, 1991). Advocates of complex tasks argue that simple item formats, such as multiple choice or short answer, encourage the measurement of discrete bits of factual knowledge rather than measuring the integretion of diverse knowledge (Resnick & Resnick, 1992). Complex tasks also lend themselves more easily to probing the types of thinking required in real-world problem solving of the modern work place (V. J. Shute et al., 2010).

A number of difficulties arise from the use of complex tasks, however. Due to the length of time such tasks require, the number of tasks that can be presented is limited. This results in few data available for inference about student abilities if the tasks are scored in a traditional manner, impacting the reliability of the measures (Brennan & Johnson, 1995). The extent of such tasks also makes them more expensive to develop and more difficult to administer. For high-stakes assessments, often professional raters are required, making the administration expensive as well. In addition, complex tasks are often embedded in a context or scenario which motivates the task. These contexts can compromise fairness and generalizability, as some students may be more familiar with or more interested in the scenario than others (Shavelson, Baxter, & Gao, 1993).

One way to deal with the issues of feasibility is to make the tasks computer-based. Through the use of technology, multi-component tasks can be presented efficiently and complex inquiries can be accomplished using software simulations (Pellegrino & Quellmalz, 2010). While a computer-based assessment may lack some of the authenticity of a hands-on science activity, there are compensating advantages. A simulation allows students to manipulate variables they would not normally have access to, perform experiments over time periods too large for a classroom experience, and see results at scales unavailable with affordable equipment (Quellmalz, Timms, Buckley, et al., 2012).

In addition to the effect on the assessment task itself, the use of technology can have a large effect on data collection. Computer-based activities provide the opportunity to track individual actions students take as they are working on a complex task. DiCerbo and Behrens (2012) point out that the use of technology for education heralds a paradigm shift in how educators and students interact with data. They coin the terms ‘digital desert,’ to describe the previous data ecology, and ‘digital ocean,’ to describe the world of abundant data in which we will soon find ourselves. Application logfiles can record student actions within a task, eye-tracking software can record where students focus their attention, and chat logs can record how students collaborate with each other. Previously, reliable inferences were dependent upon repeated measures in carefully controlled assessment environments. In a world which provides streams of performance data, we could achieve reliable inferences about student cognition by aggregating the smaller amounts of information contained in a much larger, more diverse, sets of data. The challenge is to build the statistical methods and models that can accomplish this task.

Complex tasks involve more sophisticated cognitive processes than do discreet recall tasks, often tapping multiple dimensions of ability and possibly multiple layers of knowledge and skills. This is one of their advantages from the perspective of authentic assessment, as they are then better surrogates of problem solving in the real world, but it also complicates the psychometric models which allow us to make reliable use of the data from the tasks. As recommended in the National Research Council's book *Knowing what Students Know* (2001), a logical way to deal with cognitive complexity is to more explicitly model cognitive processes. Fortuitously, the data afforded by computerized tasks may enable the estimation of these more advanced models.

To implement the vision set forth in the NGSS we will need to develop models and techniques that allow us to make valid and reliable inferences about students' ability to do science. The psychometric implications of the complex tasks required for these assessments have not yet been fully explored (Fu, Raizen, & Shavelson, 2009; National Research Council, 2014). I propose that the use of more sophisticated cognitive models, driven by the extra information contained in action data collected while students interact with complex tasks, might prove to be a productive research direction.

This dissertation is in a three paper format. Chapter 2 attempts to disentangle the layers of cognitive complexity which might confound measurement in complex tasks. A mixture IRT model is presented that simultaneously classifies student understanding of a task while measuring student ability within their class. The model is based on differentially scoring the subtask action data from a complex performance. Chapter 3 explores the potential of using a more complex cognitive model for assessment purposes. Borrowing from the cognitive science domain, student decisions within a complex task are modeled with a Markov decision process. Chapter 4 then applies the Markov decision process measurement model to an educational game to explore the practical benefits and difficulties of using such a model with real world data.

Chapter 2

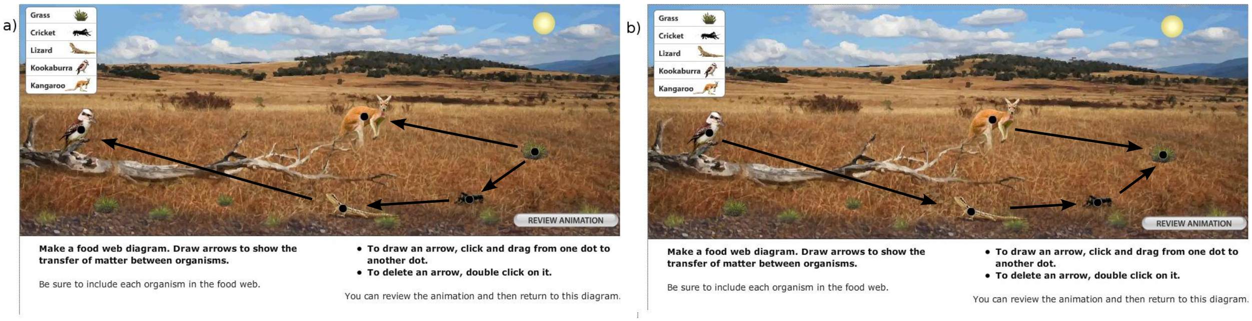
Diagnosing Student Understanding using a Differentially Scored Mixture Model

2.1 Introduction

When students engage in a complex assessment task, a wide variety of cognitive factors come into play. Not only do the students have varying levels of ability in skills that are relevant to the task, but they also have multiple layers of mental models or schema (Siegler, 1976) which guide their performance. While understanding of the content being assessed is a component of these mental models, the models also include conceptions about the goal of the task assigned and beliefs about which strategies should be applied (Bransford, Brown, & Cocking, 2000). These different layers of cognitive factors can complicate measurement of any single ability construct.

For example, a pilot version of the SimScientist ecology assessment (Quellmalz, Timms, Silberglitt, & Buckley, 2012) includes a complex task in which students were asked to draw a food web after having observed the feeding interactions of a handful of organisms. The task was scored based on how many correct and incorrect arrows the students drew. Students generally performed well on this task, with over 50% of them drawing the correct food web (Figure 2.1a). However, the third most common response pattern included no correct arrows at all (Figure 2.1b). This response pattern connected all of the correct organisms, but reversed the directionality of the arrows. While the arrow directionality is important, as it indicates the flow of matter and energy within the food web, the low score which these responses were assigned did not accurately reflect these students' understanding of the ecological system. In this case, the students' concept of arrow directionality in a food web confounded the measurement of their understanding of the feeding relationships.

This chapter proposes the differentially scored diagnostic mixture model (DSDMM) which attempts to both classify students by their conceptions of the task goal and provide a mea-



Make a food web diagram. Draw arrows to show the transfer of matter between organisms.

Be sure to include each organism in the food web.

- **To draw an arrow, click and drag from one dot to another dot.**
- **To delete an arrow, double click on it.**

You can review the animation and then return to this diagram.

Make a food web diagram. Draw arrows to show the transfer of matter between organisms.

Be sure to include each organism in the food web.

- **To draw an arrow, click and drag from one dot to another dot.**
- **To delete an arrow, double click on it.**

You can review the animation and then return to this diagram.

Figure 2.1: a) 54% of students drew the forward-arrow food web; b) 9% drew the reverse-arrow food web

surement within each class of how well the student is able to perform the task given their task conception. The DSDMM models the population as a mixture of sub-populations, each of which hold a specific conception about how the task should be performed and each of which can be modeled using an IRT model in which their conception is considered correct.

Diagnostic models attempting to classify students according to their misconceptions or strategies have a rich history in psychometrics. Tatsuoka (1985) introduced rule-space methodology with the goal of using both ability estimates and fit metrics to cluster students by misconception. The misconceptions were matched to the student clusters by calculating the ability estimate and fit of the response vectors that best represented the performance of a student with each misconception of interest. Mislevy and Verhelst (1990), on the other hand, introduced a mixture LLTM that parameterized item difficulty differently based on the strategy used to solve the item. The mixture LLTM is able to both classify students by strategy use and give a measure of the student's ability to solve the items with their chosen strategy. Wilson extended this approach into the developmental domain with his SALTUS model (1989), in which different classes represent developmental stages.

More recently, Bradshaw and Templin (2013) combined item response and diagnostic classification models to produce the SICM model, intended to both identify student misconceptions and estimate student ability. Their approach is based on the nominal response model (Bock, 1972), thus utilizing the information contained in incorrect as well as correct responses. The misconceptions are treated as person attributes much like skill attributes in a traditional diagnostic classification model (Rupp, Templin, & Henson, 2012). The probability of selecting a particular response is then modeled as a combination of the continuous ability parameter θ and boolean misconception parameters within a logistic framework.

All of these approaches take a confirmatory view of classification, with the assumption that the meaningful strategies, misconceptions or stages have already been identified. As such they emphasize the importance of a strong cognitive model for producing diagnostic estimates that are both interpretable and of educational value.

For the DSDMM the different diagnostic classes represent different ways in which a student conceives that the task should be completed. A student who has no conception of how

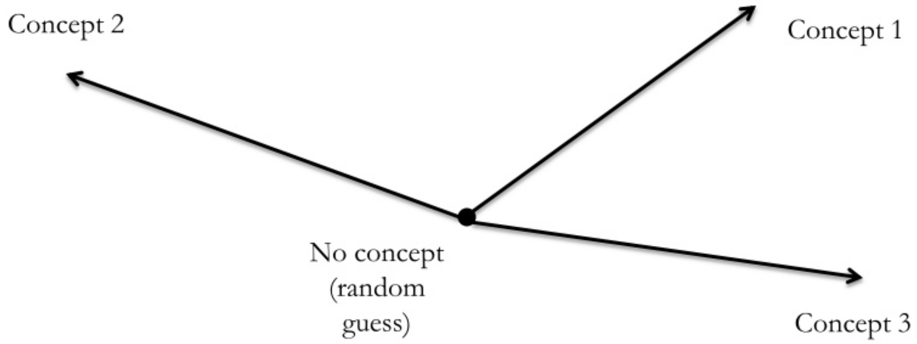


Figure 2.2: The DSDMM classification puts a response pattern on a concept vector, while the ability estimate locates it along the vector.

the task should be completed will likely either produce no answer, in the case of a fully constructed response task, or choose randomly among the response alternatives in the case of a forced choice task. A student who has a task conception will, to the best of their ability, complete the task according to that conception. Thus two layers of understanding are being represented by the model: first, an ability to understand what the task is asking for and second the ability to complete the task according to that understanding. By separating the two layers, diagnostic information about task conceptions, which may include educationally relevant misconceptions, can be obtained while also estimating the students' ability to execute the task as they conceived it.

Geometrically, the DSDMM attempts to place each student's performance within a concept space (Figure 2.2) in which each conceptual class lies along a vector. The vector's orientation defines the concept and the distance along the vector represents the ability of the student to perform well in accordance to that concept. The origin in this space represents random response selection, or no intelligible response in the case of a constructed response task. Thus at the origin, the performance is exhibiting no relation to any conceptual class whatsoever. For a given assessment, defining the cognitive model involves selecting which concepts are of interest and mapping the performance within each conceptual class onto the possible responses for the items on the assessment. This process is similar to the definition of a construct map and outcome space in the BEAR Assessment System (Wilson, 2005). In our model, the concepts of interest are each treated as mutually exclusive constructs. One important distinction between this model and other diagnostic classification models is that the different classes do not converge to a common response vector at high ability levels. Each class will converge to its' ideal response vector at high ability levels, contextualizing performance entirely within the class conception.

For the food web example task, each pair of organisms can be considered as an item. For any pair, Org-A and Org-B, there are three possible responses, an arrow drawn from Org-A to Org-B, an arrow drawn from Org-B to Org-A, or no arrow drawn. Table 2.1 shows the items and ideal responses for two concept classes: the forward-arrow food web and the reverse-arrow food web. For fixed-response dichotomously-scored items, these ideal responses serve as answer keys. Within each class, the students' responses are scored according to the class-specific answer key.

Table 2.1: Ideal responses (answer keys) for the forward-arrow food web and the reverse-arrow food web conceptual classes. (G = grass, Ka = kangaroo, Cr = cricket, L = lizard, Ko = kookaburra)

	G-Ka	G-Cr	C-L	L-Ko	G-L	G-Ko	C-Ka	C-Ko	Ka-L	Ka-Ko
Forw FW	→	→	→	→	-	-	-	-	-	-
Rev FW	←	←	←	←	-	-	-	-	-	-

In this chapter I explore the properties of this model through simulation studies and the practical applicability of the model by fitting it to empirical data from the food web task described above.

2.2 Theoretical Foundation

Model

The differentially scored diagnostic mixture model (DSDMM) is a latent class model in the form of a mixture IRT model (Rost, 1990; Mislevy & Verhelst, 1990). For students who are members of a particular class, it is assumed that their goal can be represented as an ideal response pattern for the corresponding task conception and so their responses are scored relative to that ideal. As class membership is unknown, and thus correct scoring is unknown, each possibly meaningful response must be modeled with the class-specific response scoring function, $b_{ik}(r)$, the score for giving response r to item i by a person in class k. In this study the model will be limited to fixed-response items (e.g. multiple choice) that are dichotomously scored, though extensions may be possible.

Given fixed-response items, we can distinguish knowing the correct answer from selecting the correct answer, as students may select a correct response as a guess (Samejima, 1979; Thissen & Steinberg, 1986). For any given class k, the probability of student j knowing the correct answer for item i is modeled as a one-parameter IRT model with class-specific item difficulty parameters,

$$p(\kappa_{ijk} = 1|z_j = k, \theta_{jk}) = \frac{e^{(\theta_{jk}-\delta_{ik})}}{(1 + e^{(\theta_{jk}-\delta_{ik})})}, \tag{2.1}$$
$$\theta_{jk} \sim N(\mu_k, \sigma_k); \delta_{ik} \sim N(0, \tau_k),$$

where κ_{ijk} is the latent binary variable that indicates whether student j in class k ($k = 1, \ldots, K$) knows the correct response to item i. $z_j \in \{1, \ldots, K\}$ is the class membership for student j, θ_{jk} is student j's ability, δ_{ik} is the difficulty of item i in class k, and K is the total number of classes.

If the student does not know the correct answer, he chooses randomly among the response alternatives according to each response's inherent attractiveness c_{irk}. Which is to say, when the answer is not known, student j's response to item i, x_{ij}, is distributed over the R response alternatives as the multi-category extension to the Bernoulli distribution, $x_{ij}|(\kappa_{ijk} = 0, z_j = k) \sim Cat(c_{i1k}, \ldots, c_{iRk}), \sum_r c_{irk} = 1$.

If the student does know the answer, we assume he chooses the correct answer with a probability of 1.0. The probability of selecting any given response for an item is then the sum of the probability of knowing the answer and selecting that response given that knowledge, and the probability of not knowing and selecting that response given not-knowing (Samejima, 1979),

$$\begin{aligned} p(x_{ij} = r|z_j = k, \theta_{jk}) \\ &= p(\kappa_{ijk} = 1|z_j = k, \theta_{jk})p(x_{ij} = r|\kappa_{ijk} = 1, z_j = k) \\ &+ p(\kappa_{ijk} = 0|z_j = k, \theta_{jk})p(x_{ij} = r|\kappa_{ijk} = 0, z_j = k). \end{aligned} \tag{2.2}$$

Note that when the student knows the correct answer, their probability of selecting any response r equals 1 if r is the correct answer and 0 if r is not. Thus, $p(x_{ij} = r|\kappa_{ijk} = 1, z_j = k)$ is equal to the class-specific score of $r, b_{ik}(r)$. Similarly, when a student does not know the answer, they guess, so $p(x_{ij} = r|\kappa_{ijk} = 0, z_j = k)$ is the class-specific guessing parameter for response r, c_{irk}. From this we get

$$\begin{aligned} p(x_{ij} = r|z_j = k, \theta_{jk}) \\ &= b_{ik}(r)p(\kappa_{ijk} = 1|z_j = k, \theta_{jk}) + c_{irk}\, p(\kappa_{ijk} = 0|z_j = k, \theta_{jk}) \\ &= b_{ik}(r)p(\kappa_{ijk} = 1|z_j = k, \theta_{jk}) + c_{irk}\{1 - p(\kappa_{ijk} = 1|z_j = k, \theta_{jk})\} \\ &= c_{irk} + \{b_{ik}(r) - c_{irk}\}\{p(\kappa_{ij} = 1|z_j = k, \theta_{jk})\} \\ &= c_{irk} + \{b_{ik}(r) - c_{irk}\}\left\{\frac{e^{(\theta_{jk}-\delta_{ik})}}{1 + e^{(\theta_{jk}-\delta_{ik})}}\right\} \end{aligned} \tag{2.3}$$

which has the form of the IRT 1PL with guessing (1PL-G; San Martín, del Pino, & De Boeck, 2006), extended to model every response rather than modeling only the scored categories.

The random selection portion of the model is important, first because it is central to a "concept-free" response pattern which serves as the origin of our concept space, and second because it prevents identification issues that arise when one concept manifests as an inversion of another concept for a particular assessment. In a Rasch model, as θ moves below zero, there comes a point at which the predicted probability of a student answering correctly becomes smaller than their probability of randomly guessing the correct answer (assuming a fixed response item). This implies that as θ gets smaller, the student is preferentially selecting incorrect responses. Such behavior indicates that he does have a concept of how the items should be answered, but his concept is the antithesis of the "correct" responses. In the DSDMM, such an incorrect conception would be modeled as a separate class, allowing the low end of every class ability spectrum to converge at the random guessing origin of the concept space.

To serve as a common origin and thus an anchor for the various concept scales, the probabilities of choosing different responses should thus converge among all classes as the θ_{jk} become small. For this purpose our guessing parameters, c_{irk}, are not class specific with $c_{irk} = c_{ir}$. One could argue further that at very low concept ability, no one response would be more attractive than any other and thus all guessing parameters should be equal as $c_{ir} = 1/R$ where R is the number of response alternatives. Thus we will consider two different models, the *full model* in which c_{ir} are free parameters, allowed to vary between 0 and 1, given the constraint that $\sum_r c_{ir} = 1,\ \forall i$, and the *restricted model* in which all c_{ir} are fixed to $1/R$.

Putting all the classes together, the overarching mixture model represents the probability of a student record as the marginal probability across class memberships,

$$p(X_j|\theta_j) = \sum_k \left(\prod_i p(x_{ij}|z_j = k, \theta_{jk}) \right) \pi_k \tag{2.4}$$

where π_k is the probability of a person being a member of class k.

Estimation

The model is estimated with the Bayesian Markov Chain Monte Carlo (MCMC) approach using OpenBUGS, an open-source evolution of WinBUGS (Lunn, Thomas, Best, & Spiegelhalter, 2000). Bayesian point estimation involves taking the parameter value associated with either the mode or the mean of the estimated posterior distribution. For the categorical class membership parameter, the mean is not meaningful, so the maximum a-posteriori (MAP) estimator is used, which is the parameter value at the mode of the posterior distribution. The continuous parameters are estimated using the expected a-posteriori (EAP) estimator, which is the mean of the posterior.

The model as expressed in the BUGS code assumes that a student can only be a member of a single class. Thus there is only one θ_j parameter per student. The meaning of this parameter, however, changes as the MCMC algorithm samples different values for z_j, so taking its expectation over the full chain would not produce a correct estimate of θ_j. Instead, after the MCMC run has converged, class-specific estimations, $\hat{\theta}_{jk}$, are calculated from those iterations in which $z_j = k$, for any k which was sampled for student j. The final ability estimate is taken as the modal class estimate, $\hat{\theta}_j = \hat{\theta}_{j\hat{z}_j}$.

For Baysian estimation, prior probability distributions, or "priors", must be specified for each parameter. The θ_{jk} and δ_{ik} parameters are assumed to be normal and take proir distributions as $\theta_{jk} \sim N(\mu_k, \sigma_k^2)$ and $\delta_{ik} \sim N(0, \tau_k^2)$. The hyper parameters μ_k, σ_k, and τ_k require priors of their own. We take μ_k to be normal as $\mu_k \sim N(0, 100)$, while the standard deviations σ_k and τ_k take inverse gamma priors as $\sigma_k \sim InvGamma(0.2, 0.2)$ and $\tau_k \sim InvGamma(0.5, 0.5)$. The categorical group membership parameters, z_j are given a categorical prior $Cat(\pi_1, \ldots, \pi_K)$, where the categorical distribution is a multi-class extension of the bernoulli distribution. The π_k and c_{ir} parameters are both probabilities with ranges constrained to $[0, 1]$. The uniform distribution over [0,1] is used as a prior for both, though it is coded in BUGS as the equivalent dirichlet distribution, $Dir(1, \ldots, 1)$, as the dirichlet is convenient for Gibbs sampling. The full BUGS code can be found in Appendix A.

For each MCMC estimation, three chains were run with a burn-in of 2000 iterations. The Gelman-Rubin statistic, $\hat{R}$ (Gelman & Rubin, 1992), was used to monitor convergence. Due to the large number of parameters in the models, we determine convergence based on the mean $\hat{R}$ across all parameters and use the recommended 1.1 threshold (Gelman & Shirley, 2011), above which an estimation run is considered non-convergent.

While the class membership parameter can be estimated for all response patterns, the concept space also includes a "concept-free" origin at which a pattern is not associated with any concept class. To identify such patterns we will examine the shape of the posterior distribution over z_j. The posterior probability for class membership, $p(z_j|X_j)$ represents the probability that student j is a member of class k given the response data X_j. Thus the posterior probability of the estimated membership, $\hat{z}_j$, provides a confidence metric for the classification. Based on a loss function which incorporates the cost of misclassifying students and the cost of failing to classify students, a threshold, l can be selected such that students for which $p(\hat{z}_j|X_j) < l$ are considered unclassifiable, or members of the "concept-free" group. The specification of the loss function and thus the threshold needed to make these determinations will be explored as part of the simulation studies.

2.3 Simulation Studies

Two simulation studies were conducted to evaluate the performance of the DSDMM. The first study examined the effect of test information and guessing parameterization on parameter recovery. Test information is varied through the number of items and the number of response alternatives per item along with the amount of distinction between the class ideal response patterns. The guessing parameters, c_{ir} were either fixed at $1/R$ or were freely estimated.

The second simulation study examined the performance of the model under more realistic conditions. As it is unlikely that all student conception classes will be identified in practice, we generated data with more classes than were used in the estimating model. We also added students to the data set who are in the 'concept-free' zone, responding based on the guessing parameters alone. Performance was evaluated based on how well we can both classify the concept-oriented students and identify the concept-free students, along with parameter recovery for person ability and item difficulty parameters.

Simulation Study 1

For this first study all students were assumed to possess a task concept and thus be a member of one of the concept classes, both for data generation and estimation. Further, all of the classes used to generate the data were represented in the fitted model.

Simulation Design

The goal of this study was to evaluate recovery of population parameters, π_k, μ_k and σ_k, student parameters, z_j and θ_{jk}, and item parameters, δ_{ik} and c_{ir}, under conditions that are likely to affect parameter estimation. The simulated conditions (Table 2.2) vary features of the assessment and features of the conceptual classes. For the assessment, the effect of test length is examined with 10, 20 and 50 item tests as well as the effect of the number of response alternatives per item, with 2, 3, 4 and 6 response categories. For the conceptual classes, this study varies the number of classes (2 or 4) and the amount of distinction between the classes. The class distinction is measured in terms of what proportion of items differ in their ideal response between the classes. Class distinctions of .25, .5 and .9 were simulated. For the four-class conditions, $R = 2$ was not included, as two response alternatives were deemed insufficient to distinguish four classes.

Table 2.2: The factors that were varied in the generation of simulation data for the first simulation study

Factor	Description	Values
I	Number of Items	10, 20, 50
R	Number of Response Alternatives	2, 3, 4, 6
K	Number of Classes	2, 4
D	Amount of Class Distinction	.25, .5, .9
G	Guessing Behavior	uniform, preferential

As we hypothesized earlier that guessing behavior might be driven by either response-specific attractiveness or by choosing uniformly at random from the available responses, the guessing behavior is our final simulation factor. For the uniform-guessing condition, the generating c_{ir} were all fixed at $1/R$, while in the preferential-guessing condition, the generating c_{ir} were allowed to vary between 0 and 1, constrained such that $\sum_r c_{ir} = 1$ for each item i. In all cases it was assumed that the response attractions were equal across classes. The full model, in which c_{ir} are free parameters, was estimated for the preferential-guessing data so that recovery of all item parameters could be assessed. The restricted model, with $c_{ir} = 1/R$, on the other hand, was estimated for both the uniform-guessing data, for which the restricted model is the same as the generating model, and for the preferential-guessing data to evaluate the consequences of mis-specifying the guessing behavior. For all conditions, a sample size of 300 subjects was used and for each condition 20 replicate data sets were generated.

The data were generated by first creating the test specifications for the first class, which is considered to be the primary class. An answer key was generated by drawing I values uniformly from the possible responses $\{1, \ldots, R\}$. Item difficulty parameters were drawn from a standard normal distribution, $\delta_{ik} \sim N(0, 1)$. The remaining $K - 1$ classes are variants of the primary class, with the amount of variation being controlled by D, the class distinction factor. For each non-primary class, $I \times D$ items (rounding down) are randomly selected to differ in conceptual meaning from that of the primary class. For these items, new correct answers are drawn from the remaining $R - 1$ responses, and new item difficulty parameters are drawn from the standard normal. The guessing parameters, which are equal across classes, are either set to be $1/R$ for the uniform-guessing datasets or they were drawn from a uniform distribution and normalized to ensure $\sum_r c_{ir} = 1$.

Next the student parameters were generated. Students were assigned to classes randomly based on the mixing parameters π_k. These parameters were fixed such that the first, or primary class, would be the largest, and the remaining classes would have equal probabilities. Thus for $K = 2$, $\pi_1 = .6$ and $\pi_2 = .4$, while for $K = 4$, $\pi_1 = .4$ and $\pi_2 = \pi_3 = \pi_4 = .2$. The student class membership parameters were drawn from the categorical distribution as $z_j \sim Cat(\pi_1, \ldots, \pi_K)$. For each class, the mean of θ_{jk} was drawn from a normal distribution, $\mu_k \sim N(0, 0.5)$. Student abilities were then drawn from the distribution associated with their class, $\theta_j | z_j = k \sim N(\mu_k, 1)$.

Finally the response data were generated in a two-step procedure. For each student j in class k and each item i, we draw κ_{ijk}, the indicator that the student knows the answer to item i, from a bernoulli distribution $\kappa_{ijk} \sim Bern(p_{ijk})$ where $p_{ijk} = p(\kappa_{ijk} | z_j = k, \theta_{jk})$ as given by Equation 2.1. If $\kappa_{ijk} = 1$, the student response is generated as the correct response for item i in class k. If $\kappa_{ijk} = 0$, the response is selected from the categorical distribution over all responses with the probability of each response selection given by c_{ir}.

The class membership estimates were evaluated using the misclassification rate, or the proportion of students whose membership was incorrectly estimated,

$$\text{MissclassRate} = \frac{1}{N} \sum_{j}^{N} I(\hat{z}_j \neq z_j). \tag{2.5}$$

The recovery of the student ability, θ_{jk}, the item difficulties δ_{ik} and the guessing parameters, c_{ir} was evaluated using the root mean squared error (RMSE). The RMSE is calculated

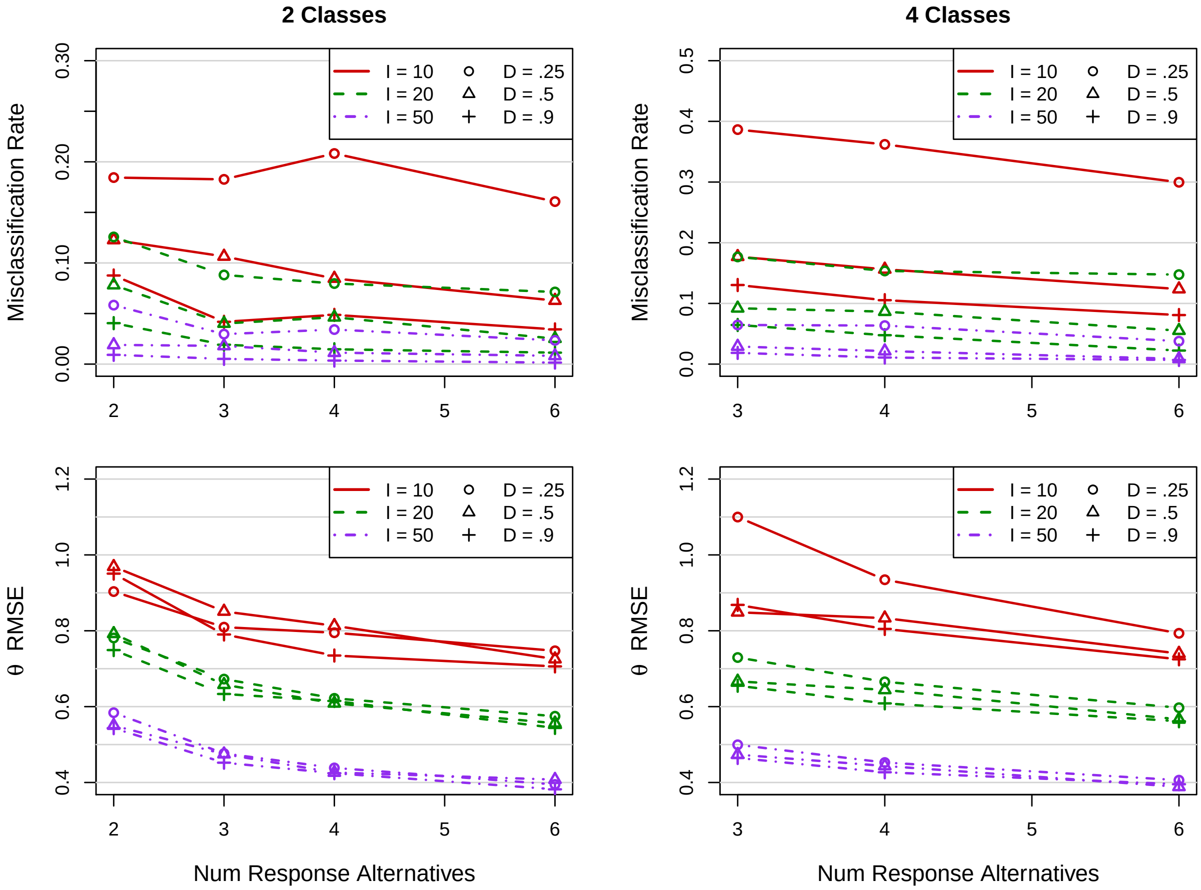


Figure 2.3: Misclassification rate and RMSE for $\hat{\theta}$ by number of response alternatives (R), number of items (I) and class distinction (D).

by taking the square-root of the mean squared error taken over all persons or items and all replications. For example the RMSE for θ is

$$\text{RMSE}(\theta) = \sqrt{\frac{1}{N \times M} \sum_{m}^{M} \sum_{j}^{N} (\hat{\theta}_j^m - \theta_j^m)^2}, \tag{2.6}$$

where N is the number of students and M is the number of replications.

Results

Full model, preferential-guessing data For the full model, which includes c_{ir} as free parameters, run on the preferential-guessing data sets, the MCMC estimation of the 1260 simulation runs converged 97% of the time, where the convergence criterion was taken to be mean $\hat{R} < 1.1$ over all parameters (Gelman & Shirley, 2011). Convergence problems were primarily associated with low information, with 73% of the problem runs having $D = .25$ and 48% having $I = 10$ and $R = 2$. Non-converging runs were removed from the results and only those estimates which converged are analyzed here.

The estimated bias and RMSE of the population parameters, π_k, μ_k, and σ_k, are shown in Table 2.3 for the two class conditions. There appears to be a small negative bias in the $\hat{\mu}_k$ parameters and a small positive bias in the $\hat{\pi}_1$ parameter. The negative bias in $\hat{\mu}_k$ is likely due to the effect of misclassified students who would have lower estimated abilities and thus bring the mean down. The $\hat{\pi}_1$ bias implies that the larger class ($k = 1$) tended to be overpopulated, which would be expected as the true $\pi_1 > \pi_2$, so borderline students would be drawn into the larger class.

Table 2.3: Population parameter recovery for the two-class full model

Parameter	Bias	RMSE
π_1	0.0046	0.054
μ_1	-0.0557	0.205
μ_2	-0.0235	0.309
σ_1	0.0000	0.077
σ_2	-0.0314	0.110

Looking at the recovery of the person parameters, we found, unsurprisingly, that the misclassification rates and the RMSE for $\hat{\theta}$ both improved as more information was added to the assessment in the form of more items, more response alternatives and more class distinction (Figure 2.3). The misclassification rate was most strongly affected by the number of distinct items between the classes, where distinct items are defined as items whose ideal response differs between classes. In fact, as seen in Figure 2.4, the relationship between the log misclassification rate and the number of distinct items ($I \times D$ rounded down) is almost linear. The RMSE for $\hat{\theta}$, on the other hand, appears to be most strongly affected by the number of items (Figure 2.3), similar to what one would expect from a single class IRT model. Under high information conditions ($I = 50, R = 6, D = .9$), the person parameter recovery was quite good with a mean misclassification rate of 0.0015 for the two class model and 0.0068 for the four-class model while the RMSE of $\hat{\theta}$ was 0.38 and 0.39 respectively.

For item parameter recovery, there appears to be a strong dependency between the estimates of δ_{ik} and c_{ir}. When we calculate a per-replication RMSE for $\hat{\delta}_{ik}$ and $\hat{c}_{ir}$, taking the mean over the items, we find that the errors for the two parameters are highly correlated ($\rho = .71$). The correlation of these errors is likely due to the known weak identifiability between the difficulty and guessing parameters in the 1PLG models (San Martín, Rolin, & Castro, 2013). For example, when many students in class k choose the correct response to an item, the resulting probabilities can be modeled either by an easy item with low $\hat{\delta}_{ik}$ or a very high chance of guessing the correct response with $\hat{c}_{ir} \approx 1.0$. As the number of response alternatives increases however, the number of c_{ir} parameters increase, with the result that

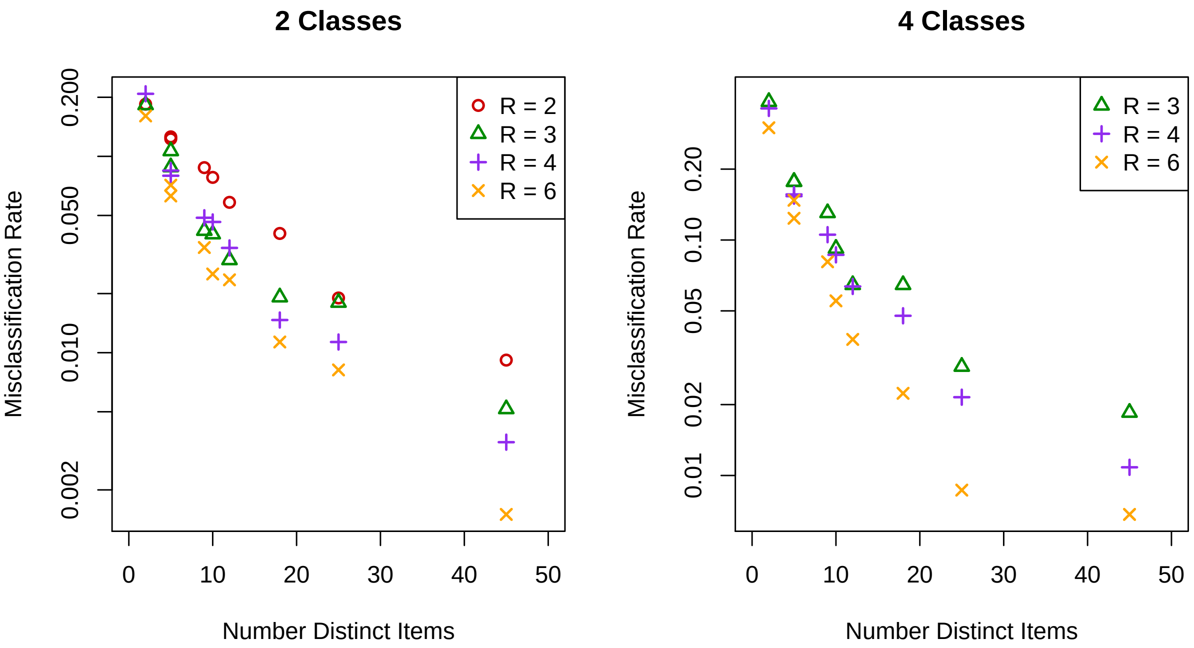


Figure 2.4: Misclassification rate, on log scale, by number of distinct items between classes and number of response alternatives (R).

any given $\hat{c}_{ir}$ is less likely to have a large error. This interpretation is consistent with the simulation results, as the number of response alternatives had the largest effect on $\hat{\delta}_{ik}$ and $\hat{c}_{ir}$ RMSE values (Figure 2.5). Interestingly, the next most significant factor appears to be the class distinction, D. For $R = 6$ and $D = .9$, the mean RMSE of $\hat{\delta}_{ik}$ over all other conditions was 0.29 logits (std. dev. 0.08). The mean RMSE for $\hat{c}_{ir}$ was 0.035 (std. dev. 0.007) over the same conditions.

Restricted Model, uniform and preferential-guessing data The restricted version of the model with c_{ir} fixed at $1/R$ was run both on the uniform-guessing and the preferential-guessing sets of data.

The recovery of person parameters was found to be fairly resilient to the handling of the guessing parameters. For the correctly specified models, an ANOVA analysis found that there was little difference between the error rates of the full and restricted models, holding $I, R, D,$ and K constant. Use of the restricted model decreased the misclassification rate by 0.005 ($p = .03$) and decreased the RMSE for $\hat{\theta}$ by 0.016 logits ($p < .001$). When the restricted model was applied to the preferential-guessing data sets, as a misspecified model, the misclassification rate showed a small increase of 0.014 ($p < .001$), while the RMSE of $\hat{\theta}$ increased by 0.024 logits ($p < .001$).

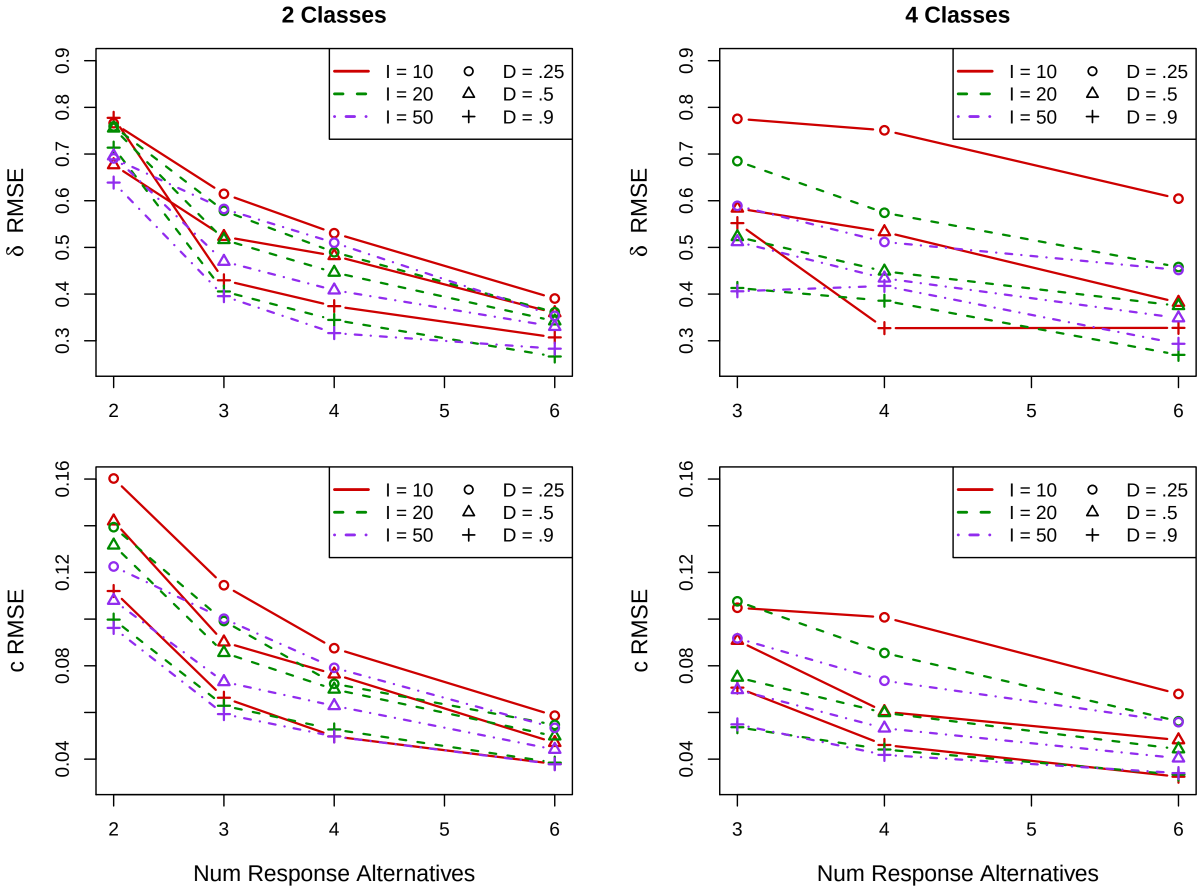


Figure 2.5: RMSE of $\hat{\delta}_{ik}$ and $\hat{c}_{ir}$ by number of response alternatives (R), number of items (I) and class distinction (D).

Not surprisingly, the handling of the guessing parameters had a large effect on item parameter recovery. The restricted model used on the uniform-guessing data decreased the RMSE of $\hat{\delta}_{ik}$ by 0.198 logits ($p < .001$). When the restricted model was used on the prefered response data sets, the RMSE of $\hat{\delta}_{ik}$ increased by 0.267 logits overall. The amount of $\hat{\delta}_{ik}$ RMSE increase was very dependent, however, upon the number of response alternatives. For $R = 2$ the RMSE of $\hat{\delta}_{ik}$ increased by 0.797 logits while for $R = 3, 4$, and 6 the increase was only $0.303, 0.141$ and 0.053 respectively.

Simulation Study 2

In the second simulation study, parameter recovery was examined under more realistic conditions. First the simulated data included a group of students who did not fall into any of the conceptual classes as a “concept-free” class. This group simulated students who had no idea how to solve the problem. In addition, the model included one fewer conceptual class than was present in the simulated population.

Simulation Design

The data sets for this simulation are all based on a medium information profile, with $I = 20$ items, $R = 4$ response alternatives per item and $D = .5$ class distinction. Fifty replicate data sets were generated with four different conceptual classes along with a fifth class of concept-free (CF) students. The class proportions were set at $\pi = \{.3, .2, .2, .2, .1\}$ to allow for a primary class as $k = 1$, and the concept-free class as $k = 5$ which takes up %10 of the population. The sample size was set at $N = 330$, so that the number of students in concept classes would be comparable to the data sets in the first simulation study (300).

Given these parameters, the data were generated using the same distributions as in the first simulation study. Concept-free students were modeled by automatically setting $\kappa_{ij5} = 0$ and drawing responses from the guessing distribution only $x_{ij} \sim Cat(c_{i1}, \ldots, c_{i4})$. The data sets were then copied and the concept-free students were removed to provide a performance baseline without the complication of concept-free students.

Two models were run on all 100 data sets. The first was the four-class full DSDMM model (i.e. with estimated guessing parameters). The second model was a three class full DSDMM model in which the fourth generating class was dropped. For both models and all data sets, "classifiable" students were identified based on a threshold, l, for the maximum posterior probability of $\hat{z}_j$. The threshold was calculated based on an arbitrary criterion that no more than 2% of the students should be misclassified. Using results from the first simulation study, l was set at .85 as it was found that for both the two and four-class models, approximately 2% of the students with $p(\hat{z}_j|X_j) > .85$ were misclassified. We expect the concept-free students to fall below this threshold and so be unclassifiable, but very low ability students within the concept classes are also likely to be considered unclassifiable.

Performance was evaluated by comparing misclassification rates and RMSE for $\hat{\theta}$ between the data sets that included no concept students, and those that did not, as well as between the data sets that were estimated using only three classes versus those that included all four conceptual classes.

Results

The MCMC estimation of the four-class model appeared to converge well, with the mean $\hat{R}$ (Rubin-Gelman statistic) for all runs falling below 1.1. For the three-class model, however, 19 of the simulation runs failed to converge based on a mean $\hat{R}$ exceeding 1.1. The data sets with convergence problems were almost equally divided between those with and without concept-free students. In three cases the corresponding concept-free and no-concept-free data sets both failed to converge. As in the first simulation study, non converging runs were excluded from the analyzed data.

Table 2.4: Performance of the three and four-class models with and without concept-free students.

	Misclass Rate				
Model/Dataset	All Students	Classifiable Only	% Classifiable	RMSE $\hat{\theta}$	RMSE $\hat{\delta}_{ik}$
4 class; no CF	.073	.014	83.0%	0.597	0.466
4 class; with CF	.172	.045	76.4%	0.612	0.401
3 class; no CF	.257	.156	79.8%	1.282	0.667
3 class; with CF	.332	.177	73.8%	1.142	0.551

The four-class model run on the data sets without concept-free students was intentionally similar to a set of simulation runs from the first simulation study. This condition serves as a baseline since it represents a correctly specified model without the complication of concept-free students. The parameter recovery for this condition was, as expected, similar to that obtained by equivalent data sets in the first simulation study (Table 2.4). The overall misclassification rate, calculated with the best class membership estimate for every student, was .073 while the misclassification rate among students who were considered classifiable, based on $p(\hat{z}_j|X_j) > .85$, was .014.

The addition of concept-free students increased the mean overall misclassification rate by about 10%, which is the proportion of concept-free student added. The classifiable student misclassification rate however remained below 5%. Note that recovery of θ (for classifiable students) and δ_{ik} did not significantly decrease based on the addition of the concept-free students.

Table 2.5: Population parameter recovery for the three and four-class models with and without concept-free (CF) students

	Bias			
Model/Dataset	$\hat{\mu}_1$	$\hat{\mu}_2$	$\hat{\mu}_3$	$\hat{\mu}_4$
4 class; no CF	-0.033	-0.071	-0.034	-0.020
4 class; with CF	-0.072	-0.392	-0.302	-0.285
3 class; no CF	-0.665	-0.517	-0.635	-
3 class; with CF	-0.598	-0.706	-0.831	-

When the three-class model was estimated on the same data sets, misclassification increased considerably, as would be expected. However, given that 20% of the students could not be correctly classified for the no-CF data sets and 30% could not be correctly classified for the with-CF data sets, the overall misclassification rates are surprisingly good. Unfortunately, the rise in RMSE for $\hat{\theta}$ implies that students from the missing class negatively impact the per-class population ability estimates. In fact the bias for μ_k (Table 2.5) clearly show how both the concept-free students and the missing class students bring the population mean ability estimates down, especially for the smaller classes ($k = \{2, 3, 4\}$).

2.4 Application Study

The DSDMM was applied to the food web task described at the beginning of this chapter to determine the feasibility and utility of the approach. Exploratory analysis of this task revealed both the frequent misconception about the directionality of arrows in a food web and a less frequent, but educationally important misconception about the distinction between food webs and food chains. The DSMRM was run with two classes to distinguish forward-arrow from reverse-arrow students. While ideally we would also use a four-class model to add the food-web versus food-chain distinction, in reality these classes were insufficiently distinct to make identification feasible. Evidence of validity was drawn from an evaluation of the classification and estimated ability of several common response patterns along with comparison with student performance on an ecology posttest.

Methodology

Data from 3455 middle school students were analyzed. The students had completed the food web task as part of a larger ecology assessment (Quellmalz, Timms, Silberglitt, & Buckley, 2012). These students also completed an ecology posttest which consisted of relevant ecology multiple choice items. In the food web task (Figure 2.1), students first watched an animation of an Australian ecosystem, and then were instructed to "Make a food web diagram. Draw arrows to show the transfer of matter between organisms." The students were allowed to freely draw arrows connecting any of the five organisms displayed on the screen: grass, kangaroo, cricket, lizard and kookaburra.

Each pair of organisms (a,b) were coded as a separate item with three response choices: 1) no arrow, 2) b→ a, and 3) a→ b, giving ten items in total. For the two classes, the forward and reverse food webs, the answer keys used are shown in Table 2.1 giving a class distinction factor of $D = .4$. The posited third and fourth classes, forward and reverse food *chains*, have answer keys that differ by only one item from their corresponding food web classes ($D = .1$). As the first simulation study has shown that even two class models with $I = 10$ and $D = .25$ yield poor classification and ability estimates, the four-class model was deemed impractical for these application data.

Initial attempts to fit the full model, with c_{ir} treated as free parameters, to the application data using the MCMC approach failed to converge even after 6000 iterations. As the parameters of interest here are the person parameters, and the simulation studies found that person parameter estimates are fairly robust when simplifying to the restricted model, in which all c_{ir} are fixed at $1/R$, this restricted model was used for the application study. The classification threshold established in the simulation studies of $p(\hat{z}_j|X_j) > .85$ was used to distinguish classifiable from unclassifiable students.

To provide validity evidence for the application results, we examine face validity by looking at how particular response patterns were classified and what ability estimates they produced as well as external construct validity by comparing the person estimates with estimates of content ability from the ecology posttest. The posttest estimates came from a two-dimensional IRT analysis of the posttest scores in which content and inquiry were modeled as two ability dimensions. The DSDMM estimates will be compared only with the content ability estimates, as those reflect the skills needed for the food-web task.

Results

Using the restricted model, the estimation took about four hours and converged with mean $\hat{R} = 1.037$ and max $\hat{R} = 1.293$.

As seen in Table 2.6, most of the students were estimated to be forward food web students. This class also had the highest estimated class-specific ability mean of $\hat{\mu}_1 = 4.3$ logits. This seems reasonable as 54% of the students drew the forward food web exactly. The reverse food web students also performed well, with a mean ability estimate of $\hat{\mu}_2 = 3.0$ logits. Only 4.5% of the students were considered unclassifiable based on our posterior probability threshold of .85. These students, unsurprisingly, had the lowest mean ability estimate of 0.8 logits.

Table 2.6: Population characteristics by classification for the FoodWeb task analysis.

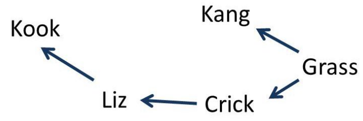
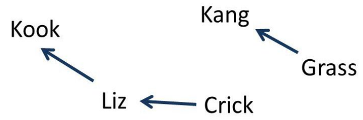
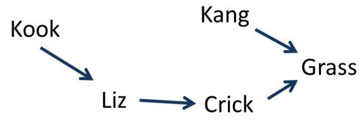
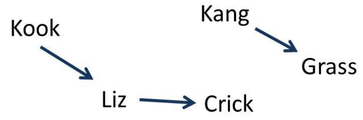
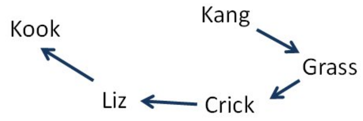
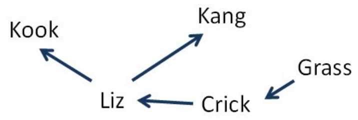
Class	Proportion of Students	Mean $\hat{\theta}$	SD $\hat{\theta}$
Forward Food Web	.768	4.30	0.92
Reverse Food Web	.186	3.01	1.36
Unclassifiable	.045	0.80	1.18

To examine face validity, the estimates for students with particular response patterns were analyzed. While there were 780 different response patterns, 77.4% of students responded with one of the top six patterns, which are shown in Table 2.7. Each response pattern was classified as we would expect. The ability estimates are also reasonable, with the class ideal response patterns generating a very high estimate, and patterns that deviate from ideal ranking lower dependent upon the amount of deviation. Note that the ability estimates of specific patterns are not directly comparable between classes as both the mean and standard deviation are class-specific.

To understand how a student gets to be labeled as "unclassifiable," we looked at the top response patterns for the unclassifiable students. They are grass→kang, cric→liz, liz←kook (12 students), grass←kang, grass→cric, cric←liz, liz→kook (9 students) and grass→kang, grass←cric, cric←liz, liz→kook. These patterns mostly connect the correct organisms, but arrow directionality appears fairly random. The unclassifiable status thus seems appropriate.

Student performance on the posttest was compared with estimated class and class-specific ability for evidence of external validity. The mean estimated content ability of students, based on the posttest, was found to be significantly different between those students classified as forward food web, 0.522 logits and those classified as reverse food web −0.433 logits, (t-value = 17.31, df = 990, $p < .001$). As the forward food web is more correct, the

Table 2.7: Population characteristics by response pattern for the FoodWeb task.

Response Pattern	Num Stud.	Classified	Mean $\hat{\theta}$	SD $\hat{\theta}$
	1842	forward	4.877	0.028
	310	forward	3.440	0.021
	298	reverse	4.240	0.032
	96	reverse	2.829	0.022
	75	forward	3.274	0.022
	55	forward	2.155	0.020

significant difference in the posttest ability estimates demonstrates both that the DSDMM can classify students into cognitively distinguishable groups and that the groups align with construct ability in the expected direction. The unclassifiable students, meanwhile, had a mean estimated posttest ability of -0.661, which was significantly lower than the reverse food web students (t-value = -2.30, df = 231, p-value = .022). Further, the class-specific ability estimates for both forward and reverse classes correlate with posttest content-ability estimates. For students estimated to be forward food web, the correlation between estimated ability from the food-web task and estimated ability from the posttest was .287 ($p < .001$) while the reverse food web students showed a correlation of .254 ($p < .001$). While both of these correlations are fairly low, the posttest covered a large range of ecology topics only one of which was tested by the food-web task. The fact that both classes show a significant correlation with posttest results supports the hypothesis that the class-specific ability estimates are evidence of the construct, even for the class exhibiting a misunderstanding.

2.5 Discussion

The first simulation study showed that we can get good recovery of person parameters, including class membership and class-specific ability, given enough test information. Clearly the numbers of items and the number of response alternatives are within the test designer's control, and so can be increased to produce acceptable results. Perhaps less obviously, the amount of class distinction can also be manipulated by the test designer, assuming that the class conceptions of interest are known in advance. If diagnosing student conception is a primary goal of the assessment, items can be designed to distinguish between the concepts, increasing the class distinction and the total number of items which differ between classes.

The second simulation study showed that failing to model significant student conceptions can result in decreased accuracy of the ability estimation as well as some decrease in the classification accuracy. This finding reinforces the conclusion that the DSDMM is best used as a confirmatory model based on a strong cognitive model of student performance on the task at hand. The addition of concept-free students did not greatly impact the classification accuracy of the classifiable students nor did it greatly increase the RMSE for $\hat{\theta}$. Thus the model seems resilient to these students near the origin of the concept space provided that an appropriate classification threshold is used.

This study explored a method for diagnosing student systemic task conceptions using an IRT mixture model. There are a number of limitations to this current work and opportunities for further research. One limitation of this study was the assumption that selection among incorrect responses is not affected by student ability. The work could be extended to include ability in the probability of selecting an incorrect response, either using the nominal response model (Bock, 1972) or the 1PL-AG (San Martín et al., 2006). It would be interesting to see if these parameters displayed any patterns between classes that might indicate that student of higher abilities transition from one class to another. In this study we considered only fixed response type items for which the probability of each possible response could be individually modeled. Future work might include an extension to fully constructed response tasks. Finally, the classification of the concept-free students by identifying unclassifiable students in a post-estimation procedure may have distorted the distribution of those students who properly belonged to a concept class. We could add a concept-free class directly to the model, perhaps in a nested classification structure, so that large proportions of uninformed students can be cleanly estimated separately from the concept-class students.

Chapter 3

Using a Markov Decision Process for Measurement of Student Capability in Complex Tasks

Complex tasks are, by their nature, made up of sub-steps. Students engaging in such tasks are required to plan out a problem-solving approach and take multiple actions to reach the final outcome. Some examples of complex tasks include laboratory experiments, questions on high level physics exams and strategy games like chess. It is usually easy to assign a value or score the final outcome – did the student draw the correct conclusions? get the right answer? win the game? But there are many ways to go wrong in a complex task and a single end result is insufficient to draw reliable conclusions about student ability. Focusing only on outcomes also ignores the wealth of data contained in the actions that students take within the task. When we can capture such sub-task action data, as is straightforward in a computerized setting, we should be able to use those data to improve our estimates of student ability.

To make use of any data in a latent trait measurement model, we need to map the observed data onto the construct that we wish to measure. In assessment, this mapping is done through scoring; a higher score shows evidence of more of the construct being measured (Wilson, 2005). Thus for the sub-task action data, the first challenge is to assign values or scores to the data provided by the actions. In a multi-step task, especially one that can be solved using different strategies, it is not clear how to assign these values. An action taken at one stage of the process might be of high value while the same action taken at a different point would be useless. A second problem with sub-task action data is the dependence within the data. Because we assume that students form and implement a plan, their actions cannot be considered independent of one another. Further complicating matters, the paths students take through the problem space may differ in length and difficulty. One student may end up facing a decision another student avoided by choosing different initial actions. This type of data is difficult to model with either classical test theory or item response theory, requiring a more structured and dynamic model (Mislevy et al., 2002).

The most common approach to dealing with data from complex tasks is to employ feature extraction in which salient indicators of the intended construct are selected and aggregated across the full student performance. Such feature extraction might involve hand coding log files to find indicators of productive behaviors (Buckley et al., 2010) or using complex algorithms to identify effective patterns of actions (Vendlinksi & Stevens, 2002). Once the features are extracted, they are fequently modeled using Bayes nets, which allow for more complex dependency relationships than are commonly found in item response theory (IRT) approaches (Mislevy et al., 2002). In some intelligent tutoring systems, actions are scored by comparison to an "ideal student model" which is based on a set of production rules that specify, for each decision point, what action would be appropriate using a series of if-then statements (Corbett, Anderson, & O'Brien, 1995). These systems, once created, can be used for assessment of complex tasks (Draney, Pirolli, & Wilson, 1995) but the production rules themselves are laborious to generate and are not generalizable.

Ideally we would like to find a feasible and generalizable method for making use of the information in each decision, or action, that the student takes within the task. One approach comes from recognizing that each decision is made within the context of the current state of the problem. For example, in chess, the state of the problem is represented by the pieces on the chess board. Any particular move only makes sense in relation to the current arrangement of the pieces. In general, the meaning, and therefore value, of each action is dependent upon the state of the problem at that moment. Further, as the problem state contains the manifest results of previous actions, each decision can be considered independent of other decisions given the current problem state. For the chess example, this implies that based only on the current state of the chess board a player can choose an appropriate move, even with no knowledge of exactly which moves brought the board to that state. This is, of course, a simplification, as in reality the game history would inform the player of which moves their opponent is likely to make, and thus which of their available moves is most likely to be successful. For now we set aside such within-task learning, though it could be an interesting extension of this work.

As the problem state gives the task actions both value and independence, the state can be considered analogous to an item in a traditional assessment framework. Thus, rather than identifying responses by sequence, we associate them with the problem state in which they were chosen. The number of possible "items" would then be as large as the state space and each student would only have responses for a small number of them. Given this approach, we could model the probability of a student taking an action in a particular state using an IRT framework by including the scoring parameter within the model as in the nominal response model (Bock, 1972)

$$p(x_{sj} = r_s|\theta_j) = \frac{\exp(b_{rs}\theta_j + \xi_{sr})}{\sum_{m=1}^{R_s}\exp(b_{ms}\theta_j + \xi_{sm})}, \quad \sum_{r=1}^{R_s}\xi_{sr} = 0, \sum_{r=1}^{R_s} b_{sr} = 0, \theta_j \sim N(\mu, \sigma^2), \quad (3.1)$$

where b_{rs} is the score parameter for response r in state s, R_s is the total number of response alternatives possible in state s, ξ_{sr} is the intercept for response r in state s, and θ_j is the ability of person j. To estimate this model we would need to estimate $2R_s - 2$ parameters for each state contained in the data. If we fix the score parameters by hand, as is traditionally done in a partial credit model, the number of parameters would be reduced to $R_s - 1$ per state, but scores would need to be specified for every action in every state. To put this in perspective, a simple strategy game such as tic-tac-toe has 5,478 legal game states (board positions), with each state allowing 2 to 9 possible actions. Specifying scoring for each state-action pair by hand would be a monumental task. Estimating the remaining parameters would require massive amounts of data, given that each game would only provide data for nine game states.

Instead, we propose explicitly modeling the student's decision making process to express the scoring parameters in terms of a smaller, more tractable, set of parameters, similar to the approach taken in an LLTM (Fischer, 1973). For this purpose we use a Markov decision process which links the probability of choosing a particular action in a particular state to the likelihood of achieving a predefined goal, such as succeeding in the task at hand. In this chapter we develop a measurement model for complex assessment tasks by combining the IRT approach with a Markov decision process cognitive model.

3.1 Markov Decision Process as a Cognitive Model

A Markov decision process (MDP) is a method for choosing optimal actions based on a longitudinal cost-benefit analysis (Puterman, 1994). The basic MDP depends upon the principle of rationality, assuming that agents will tend to choose the most beneficial action based on their goals and understanding of the system.

Formally, an MDP is defined by $\{S, A, T, R, \gamma\}$ where S is the set of possible states of the system and A is the set of actions that one can take. T represents the transition model, $p(s'|s,a)$, the probability of transitioning to a state s' given that action a was taken in state s. R corresponds to the reward structure $r(s,a,s')$ which specifies the immediate reward for taking action a in state s and entering state s', while $\gamma \in [0,1]$ is the discount parameter, representing the relative value of future versus immediate rewards. From this specification, one can calculate the Q function, which is the expected sum of discounted rewards obtained by taking action a while in state s:

$$Q(s,a) = \sum_{s' \in S} p(s'|s,a) \left(r(s,a,s') + \gamma \sum_{a' \in A} p(a'|s') Q(s',a') \right), \tag{3.2}$$

where $p(a|s)$ is the decision rule, or policy, by which actions are chosen given a particular state. The Q function essentially assigns a value to each action in each state. In Equation 3.2, $r(s,a,s')$ is the immediate reward for taking action a in state s, assuming it results in state s', while $\sum_{a' \in A} p(a'|s')Q(s',a')$ is the expected value of the next state, marginalized over the possible next actions. Thus the quantity inside the large parentheses is the sum of the immediate reward and the discounted value of the future state. The expectation of this sum is then taken over all possible states s' that might result from action a in state s. Note that the function is recursive, as the value of a state is defined using the Q function itself. The Q function can be calculated using dynamic programming (Howard, 1960).

Historically, MDPs have been used in the context of robotics or computer artificial intelligence to find optimal paths through a problem space. The optimal policy, $\pi(s)$, is the set of actions for a given state s which maximize a specified criterion (Puterman, 1994). When the MDP problem has a definitive end point, the optimization criterion is usually taken to be the maximal expected total rewards. The optimal policy therefore can be found by maximizing a modified Q function, $Q^*(s,a)$, over all possible actions (Ng & Russell, 2000). Q^* assumes that an optimal action will also be taken in all future states:

$$\begin{aligned} Q^*(s,a) &= \sum_{s'\in S} p(s'|s,a)\left(r(s_i,a_i,s') + \gamma Q(s',\pi(s'))\right) \\ \pi(s) &= \arg\max_{a\in A}(Q^*(s,a)). \end{aligned} \tag{3.3}$$

Note that for optimal policies, it is assumed that an optimal action is taken with a probability of 1, $p(a \in \pi(s)|s) = 1$, thus the last term in Equation 3.2 reduces as

$$\begin{aligned} \sum_{a'\in A} p(a'|s')Q(s',a') &= \sum_{a'\in\pi(s)} p(a'|s')Q(s',a') \\ &= \max_{a\in A}(Q(s',a)) \sum_{a'\in\pi(s)} p(a'|s') \\ &= \max_{a\in A}(Q(s',a))p(a\in\pi(s)|s) \\ &= \max_{a\in A}(Q(s',a)) = Q(s',\pi(s')). \end{aligned} \tag{3.4}$$

More recently, Markov decision processes have been used as a cognitive model to describe not only human decision making, but also people's ability to infer the goals and beliefs of others. Baker, Saxe and Tenenbaum, (2011), describe a "Bayesian theory of mind" in which cognition is modeled as a partially observed MDP. They hypothesize that people act based on their beliefs, modeled by the state space, action set and transition functions, and in accordance to their desires, which are modeled by the reward structure. In their earlier work (2009), they used an MDP to show how one person might infer the goals of another through the observation of their actions. When modeling human decision making, the policy is not assumed to be optimal, as humans make mistakes. Frequently a Boltzmann policy is used (Baker et al., 2009),

$$p(a|s) \propto e^{\beta Q(s,a)} \tag{3.5}$$

where $\beta \in [0, \infty)$ represents the decision maker's capability to optimize actions according to the Q function. As β increases, the probability of taking the action with the highest Q value, i.e. $\pi(s)$, increases. When β goes to zero, the action probabilities become equal, and actions are selected uniformly at random.

Note that under this model the decision maker is at worst performing randomly. As a cognitive model, the MDP specifies the individual's actual goals and beliefs, and we assume that the individual's actions are consistent with those goals and beliefs based on the principle of rationality (Baker et al., 2009). Thus while an individual might make mistakes in the pursuit of their goals, represented by lower β values, they will not consistently act contrary to their interests, based on their understanding of the situation. On the other hand, if we were to specify an alternative MDP $M' = \{S', A', T', R', \gamma'\}$ which did not match the decision maker's internal model, then it would be possible for the probability of optimal action under M' to fall below random chance.

In the cognitive modeling domain, however, the intent is to infer what MDP is being used internally by a decision maker. As such, all the MDP parameters represent the individual's understanding of the problem space. The reward structure R reflects that person's personal goals and values; the action set A includes only the actions they know are available; the state space S includes only system states they understand as possible; and T represents their understanding of the system dynamics. This subjective quality of the model allows it to be used for making inferences about different elements of an agent's cognition, based upon their actions (Baker et al., 2011; Rafferty, LaMar, & Griffiths, in press). In particular, inverse reinforcement learning utilizes MDPs to infer discrete goals based on action traces by estimating the most probable values of the reward function (Ng & Russell, 2000), while inverse planning algorithms infer student understanding of the effects of their actions by estimating parameters in the transition function (Rafferty et al., in press).

3.2 Markov Decision Processes for Assessment

While the Markov decision process model contains numerous parameters that could be allowed to differ by person, for this study we will focus primarily on the single variable β, as a measure of a student's capability to optimally solve a specific problem. As is appropriate for an educational assessment, we will fix the state space, action set and transition parameters to objectively correct values so that β represents the capability to solve the problem accurately, rather than to solve the problem given misconceptions about the problem space. Parts of the reward structure, on the other hand, may depend upon student motivation. We will explore how such motivation parameters interact with student problem solving capability and whether or not they are distinguishable. In previous work using the MDP model for inference (Baker et al., 2011; Rafferty et al., in press), person-specific goals and beliefs were estimated, but β was assumed to be common across participants, and was either fixed or estimated at the population level. Here we propose to make β person specific and so give it a person-specific subscript, β_j. The formulation of the Q function remains as in Equation

3.2, except that we note explicitly the dependency upon the capability parameter β_j. The conditional probability of student j selecting action a when in state s now becomes

$$p(a|s,\beta_j) = \frac{\exp\left(\beta_j Q(s,a|\beta_j)\right)}{\sum_{a' \in A} \exp\left(\beta_j Q(s,a'|\beta_j)\right)} \tag{3.6}$$

As the MDP model for assessment reflects the correct, rather than the student, conception of the problem, it is now possible for students to consistently choose a non-optimal action. In this study, however, we will retain the minimum limit of zero for β_j as that is traditional for the MDP model. Students who have misconceptions will then tend to cluster near that lower limit.

It will be noted that Equation 3.6 is very similar to the nominal response model discussed earlier (Equation 3.1), and is part of the family of item response theory models known as divide-by-total models (Thissen & Steinberg, 1986). There are two significant differences from traditional IRT models. First, the Q functions, which might be seen as discrimination or scoring parameters, depend themselves upon the value of β_j. Functionally, $Q(s,a|\beta_j)$ acts very much like a response score as it gives a value for each possible action in each state. However, because the action probability is part of the Q function (Equation 3.2), $Q(s,a|\beta_j)$ does not fit the traditional concept of an objective score. For the MDP measurement model, we refer to $Q(s,a|\beta_j)$ as the *action value* and note that it is dependent upon both person and state.

The second difference from standard IRT formulations is that Equation 3.6 lacks an intercept parameter, which in the IRT framework represents item difficulty for a dichotomous model or response attractiveness in a partial credit or nominal model. The lack of intercept implies that for all states, as β_j goes to zero, the probability of selecting any action a goes to $1/R$ where $R = |A_s|$.

While the lack of intercept suggests that the model ignores differences in decision difficulty, within the MDP framework decision difficulty is instead represented by the contrast among the Q values for the available actions of a given state. When a decision is 'easy,' one action will have a much higher Q value than the others, making the selection of the correct action quite probable. For a more challenging decision, the distinction between the actions will be more subtle and the Q values will be closer together. The implications of interpreting Q value differentials for decision difficulty will be further explored in Section 3.5.

The remaining MDP parameter space includes the reward-structure parameters and the discount parameter. The reward structure is usually simple to parameterize. For example, rewards can often be specified by a large positive end reward for achieving the correct final state R_{goal} and a small negative cost for taking any action that does not result in the goal state R_{move}. This reward structure will always be describable with two parameters, no matter how large the state space or action set. We will discuss the implications of the reward parameters for model identification in Section 3.5 and explore the interplay between reward and capability at the population level in Section 3.6. The final parameter, the discount parameter, modulates the value of rewards over time, in particular allowing future rewards to be less valuable than present rewards. Depending upon the specific problem being modeled, the discount parameter might be considered to be subjective and therefore a person-specific parameter. On the other hand, for many short-term problems in which particular outcomes result in specified rewards, such as a scored test or a straight-forward strategy game, the discount parameter is logically set to 1.0 as the final score is considered to be equally valuable at any point in the task. For this study we shall consider the discount parameter to be fixed and known.

3.3 Estimation

The observed data for student j consist of a sequence of state-action pairs,

$$O_j = \{(s_{1j}, a_{1j}), (s_{2j}, a_{2j}), \ldots (s_{T_j j}, a_{T_j j})\}. \tag{3.7}$$

where T_j is the total number of actions taken by the student. Each pair indicates a state and the action taken in that state. When it is necessary to capture the final state after the last action is taken, an extra action such as $STOP$ is added to form the last pair in the observed data and T_j is incremented to include this last pair.

The Markov property applies to this model, allowing us to take each action to be conditionally independent, conditioned upon student capability and the system state in which the action was taken. Thus the probability of the observed data can be written as

$$\begin{aligned} p(O_j|\beta_j) = &\prod_{t=1}^{T_j} p(a_{tj}|s_{tj}, \beta_j) = \prod_{t=1}^{T_j} \frac{\exp\left(Q(s_{tj}, a_{tj}|\beta_j)\beta_j\right)}{\sum_{a' \in A} \exp\left(Q(s_{tj}, a'|\beta_j)\beta_j\right)}. \\ &\beta_j \sim \ln N(\mu, \sigma^2) \end{aligned} \tag{3.8}$$

We estimate the model by taking the β_j parameters as random effects with a parametric log-normal distribution, $\ln N(\mu, \sigma^2)$. A variable is log-normally distributed when a log transformation of the variable would be normally distributed, thus we can alternatively define $\beta_j = \exp(\lambda_j)$ with $\lambda_j \sim N(\mu, \sigma^2)$. The use of the log-normal distribution ensures that β_j is restricted to be non-negative, as is desirable for our interpretation of the latent trait, but it also makes sense conceptually. The log-normal distribution is found naturally as the distribution of growth metrics (Limpert, Stahel, & Abbt, 2001). In particular when growth is best modeled by a multiplicative rather than an additive process the log-normal distribution results. As learning is arguably a multiplicative growth process, the log-normal is attractive among the positive-bound probability distributions. Log-normal ability distributions have also been used previously in psychometric models to achieve a lower bound to the probability of response selection without the introduction of guessing parameters (Bradshaw & Templin, 2013).

We define the set of all model parameters to include the capability distribution parameters, μ and σ, along with the Q function parameters, R which are the set of parameters needed to define the reward structure. To estimate the model parameters, we use marginal maximum likelihood (MML), marginalizing over the β_j distribution. The marginal likelihood for the model parameters is

$$L(\mu, \sigma, R) = \int_{\beta_j} \prod_{j}^{N} p(O_j|\beta_j; R) G(\beta_j; \mu, \sigma) d\beta_j, \tag{3.9}$$

where $G(\beta_j; \mu, \sigma)$ is the log-normal probability distribution. This likelihood cannot be evaluated analytically. Not only is the integral intractable, but the Q-function must be calculated through iterative approximation. To evaluate the integral, Gaussian quadrature is used for integration over $\lambda_j = \ln(\beta_j)$ with nine quadrature points. The MDP Q-function is calculated using policy iteration methods (Howard, 1960) in which the state values and action probabilities are iteratively updated until they are changing less than a specified convergence criterion.

The maximization could then be performed either by an iterative optimization algorithm over the parameter space, or through an MCMC approach if prior distributions are specified for all model parameters. Numerical approximation is used for this study. The maximization over the parameter space is implemented using the nlopt C++ library in two stages, with a "global" search conducted over a large range followed by a localized optimization using the global results as starting values. These algorithms introduce some randomness to avoid local optima, so multiple runs over the same data do not always produce the exact same results.

After the model parameters have been estimated, β_j parameters are predicted using empirical Bayesian estimation, with the prior set to the empirically estimated distribution $\ln N(\hat{\mu}, \hat{\sigma}^2)$. The point estimates are taken as the maximum a-posteriori (MAP) estimates, which correspond to the β_j values at which the posterior distribution is maximized,

$$\hat{\beta}_j = \underset{\beta_j}{\operatorname{argmax}}\, p(O_j|\beta_j) G(\beta_j; \mu, \sigma). \tag{3.10}$$

All of the estimation code was custom written using the C++ programming language.

3.4 An Illustrative Example

To illustrate the functioning and psychometric properties of the model, as well as to study parameter recovery in the simulation studies, we will use as an example the popular board puzzle game known as "Peg Solitaire." The peg solitaire game consists of a board with holes, some of which are filled with pegs. The holes can be arranged in a rectangular, triangular or hexagonal grid, though only rectangular grids are considered here. Legal moves involve one peg jumping over an adjacent peg into an empty hole on the other side, after which the jumped peg is removed from the board. The goal is to leave as few pegs on the board as possible. Often leaving a single peg is considered a win, while leaving more than one peg is a loss. The complexity of the game can vary depending upon the size and configuration of the board and starting position.

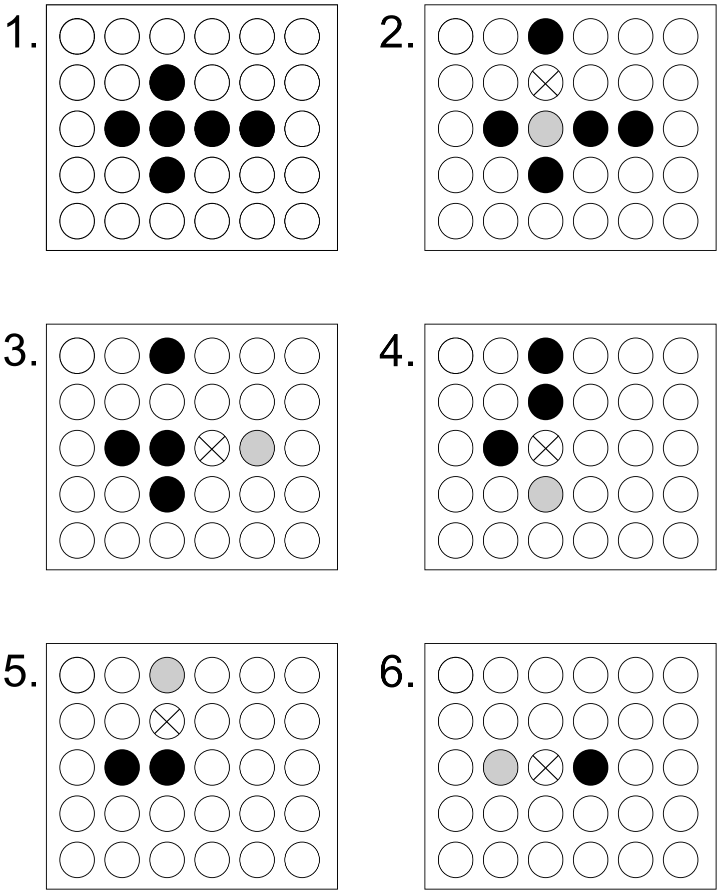


Figure 3.1: A simple Peg Solitaire game. The empty circles are holes and black filled circles are pegs. The upper left represents the starting board. Gray circles show where pegs were moved from, while the X shows where pegs were jumped over and removed. Each numbered board configuration corresponds to a game state.

As an example, a very simple game is shown in Figure 3.1. Reading from the starting state 1 on the top left, a 5-move winning solution is shown. Each jump is shown in the following diagram, so from board position 1, we see that the middle peg is moved up to the top row, and the peg that was jumped is removed, as shown in board position 2. Completing this puzzle takes exactly five moves, but the solution is not unique. Note that the starting position is symmetric, and an equivalent winning solution can be obtained by jumping the middle peg down in the first move rather than up. Similarly, the very last move of the game can be taken in either direction.

To describe the peg solitaire game as a Markov decision process (MDP), each element of the MDP, (state space, action set, transition function, reward function and discount parameter) must be specified for the game. The general state space includes all possible configurations of the pegs on the board. For a particular game however, the state space S can be reduced to all reachable peg configurations given the starting configuration. For the very simple game shown in Figure 3.1, there are only 22 reachable states, six of which are shown. Note that a game state involves only the actual position of the pegs, not the move which preceded it, although the move is shown in the diagram for clarity. Many specific states might be achieved through more than one path. The full action set A contains all possible peg jumps at any point in the game. For a particular state, however, the number of legal moves is limited. Therefore we define subsets $A_s \subset A$ as all possible actions from state s. The possible actions include all legal peg jumps in state s along with the *reset* and *score* actions. The *reset* action resets the board back to the starting state and is allowed in any state other than the starting state or the final winning state. The *score* action ends the game and assigns the final rewards for the board. It is allowed from any state.

The transition model, $p(s'|a, s)$ is deterministic for the peg solitaire game. A legal move will transition to the board state which has the jumping peg moved over and the jumped peg removed with a probability of 1.0. The probability of transitioning into any other state is 0.0. Also the probability of an illegal move transitioning into any state what-so-ever is 0.0. The *reset* action transitions to the starting position with a probability of 1.0, while the *score* action does not change the game state.

The reward function could either reward only a complete win, leaving only one peg on the board, or it could give 'partial credit' by assigning decreasing rewards for final positions that include more pegs. It is interesting to note that the different reward structures would change the behavior of skilled players. If we reward only single-peg solutions, a player who has made a mistake would most likely reset the game or quit, depending upon the cost of each move and the length of the current path. Under the partial-credit reward structure, however, it might make sense to continue on to a near-win result. For this study we define a partial-credit reward structure with four parameters as described in Table 3.1.

Finally, for this game we take the discount parameter γ to be fixed at 1.0, as the value of winning is not smaller earlier in the game than it is later in the game.

Using the example reward values from Table 3.1 for the example game pictured in Figure 3.1, we will now illustrate how the action values and probabilities could be calculated. Pegs and holes will be referenced by row and column, (r,c), where $(1, 1)$ is the hole in the upper

Table 3.1: Reward parameterization for the Peg Solitaire MDP.

Parameter	Description	Example Value
R_{win}	Reward for scoring with only one peg left on the board	5.0
R_{peg}	Adjustment to score for each additional peg remaining	-1.0
R_{move}	Reward (cost) for each game move	-0.1
R_{reset}	Reward (cost) for reseting the game	-1.0

left corner of the board. It is often easiest in MDPs to work backwards from an end state (Puterman, 1994). Thus we will start with state 6, shown in Figure 3.1. State 6 has only one remaining peg and so the only action available is *score*. The *score* action ends the game, so there are no future actions and there is no change of state. Thus the Q function reduces to only the immediate reward for scoring the game in state 6, $Q(s_6, a_{score}) = r(s_6, a_{score}, s_6)$ which is $R_{win} = 5.0$.

Iterating backward, we now look at state 5. There are four actions available in state 5, but we will ignore the *reset* action for now, as it produces loops and so complicates the calculations. Of the three remaining actions, two are peg moves: moving the $(3, 2)$ peg to $(3, 4)$ or moving the $(3, 3)$ peg to $(3, 1)$, and the third action is *score*. Both peg moves result in a winning state and thus are accorded equivalent value. The real choice here is between making one of those moves, or quitting early. Table 3.2 shows the Q values for each move and the action probabilities for two sample values of β_j. For the peg move $(3, 2) \rightarrow (3, 4)$, the Q value is calculated as

$$\begin{aligned} Q(s_5, a_{3234}) &= r(s_5, a_{3234}, s_6) + \sum_{a' \in A} p(a'|s')Q(s', a') \\ &= r(s_5, a_{3234}, s_6) + Q(s_6, a_{score}) \\ &= R_{move} + R_{win} = -0.1 + 5.0 = 4.9 \end{aligned}$$

As there is only one possible action in the next state, the action probability is 1, and so the Q value is not differentiated by β_j value. The action probabilities, however, show that the lower capability student has a higher probability of quitting before reaching the highest score, as the β_j values do come into that calculation,

$$p(a_{score}|s_5, \beta_j) = \frac{\exp(\beta_j Q(s_5, a_{score})}{\exp(\beta_j Q(s_5, a_{3234}) + \exp(\beta_j Q(s_5, a_{3331})) + \exp(\beta_j Q(s_5, a_{score}))}$$

State 4 has only two possible actions, move $(1, 3) \rightarrow (3, 3)$ or *score* (Table 3.3). For this state, however, the value of the Q function is now slightly different for $\beta_j = 0.5$ vs. 2.0. Because a student with $\beta_j = 0.5$ is less likely to take the correct move if they get to state 5, the move that takes them to that state is less valuable. For example the value for the move

Table 3.2: Values and probabilities for the example game, state 5.

Move	$Q(s,a\|\beta_j)$		Action Probabilities	
	$\beta_j = 0.5$	$\beta_j = 2.0$	$\beta_j = 0.5$	$\beta_j = 2.0$
$(3,2) \rightarrow (3,4)$	4.9	4.9	.38	.46
$(3,3) \rightarrow (3,1)$	4.9	4.9	.38	.46
score	4.0	4.0	.24	.08

$(1,3) \rightarrow (3,3)$ is

$$\begin{aligned} Q(s_4, a_{1333}|\beta_j) &= r(s_4, a_{1333}, s_5) + \sum_{a' \in A} p(a'|s', \beta_j) Q(s', a'|\beta_j) \\ &= r(s_4, a_{1333}, s_5) + p(a_{3234}|s_5, \beta_j) Q(s_5, a_{3234}|\beta_j) + \\ &\quad p(a_{3331}|s_5, \beta_j) Q(s_5, a_{3331}|\beta_j) + p(a_{score}|s_5, \beta_j) Q(s_5, a_{score}|\beta_j). \end{aligned} \tag{3.11}$$

If we plug in the values for $\beta_j = 0.5$ or $\beta_j = 2.0$ from Table 3.2, we see how the Q function begins to differ by capability,

$$Q(s_4, a_{1333}|\beta_j = 0.5) = -0.1 + .38 \times 4.9 + .38 \times 4.9 + .24 \times 4.0 = 4.6$$
$$Q(s_4, a_{1333}|\beta_j = 2.0) = -0.1 + .46 \times 4.9 + .46 \times 4.9 + .08 \times 4.0 = 4.7$$

Table 3.3: Values and probabilities for the example game, state 4.

Move	$Q(s,a\|\beta_j)$		Action Probabilities	
	$\beta_j = 0.5$	$\beta_j = 2.0$	$\beta_j = 0.5$	$\beta_j = 2.0$
$(1,3) \rightarrow (3,3)$	4.6	4.7	.69	.97
score	3.0	3.0	.31	.03

Looking at game state 3, we see four different available moves (Table 3.4). The optimal move leads to state 4, while the other three are dead end moves which must be scored in the next action for a value of 3.0. Their move value is thus 2.9, as the cost of making a move is -0.1. Here we see much greater differences in the value of the correct move, 4.0 vs. 4.6 for the β_j values of 0.5 and 2.0. We also see large differences in the action probabilities. The student with $\beta_j = 0.5$ is predicted to have a 32% chance of choosing the optimal action, while the $\beta_j = 2.0$ student has a 90% chance of selecting it.

Going all the way back to the beginning of the game we see that State 2 has one optimal move out of three possible actions while state 1 has two optimal moves out of five possible actions. The Q values and probabilities could be similarly calculated for these states, but will not be shown here. It should be noted that this method of calculating the Q values via backward propagation is not possible in our actual MDP setup as the *reset* action produces

Table 3.4: Values and probabilities for the example game, state 3.

Move	$Q(s,a\|\beta_j)$		Action Probabilities	
	$\beta_j = 0.5$	$\beta_j = 2.0$	$\beta_j = 0.5$	$\beta_j = 2.0$
$(4,3) \to (2,3)$	4.0	4.6	.32	.90
$(3,2) \to (3,4)$	2.9	2.9	.19	.03
$(3,3) \to (3,1)$	2.9	2.9	.19	.03
$(3,3) \to (5,3)$	2.9	2.9	.19	.03
score	2.0	2.0	.12	.005

possible loops in the state space. In the actual code, policy iteration methods (Howard, 1960) are used.

Our example showed action probabilities for two different values of β_j. To see how the probabilities change over the capability range, Figure 3.2 shows the probability of selecting an optimal move for states 2, 3, 4 and 5 as a function of β_j. One interesting consequence of having the lowest value of β_j correspond to uniform random selection is that the probabilities of a correct move near $\beta_j = 0$ are dominated by the ratio of correct choices to total choices. This suggests that greater distinction between action probabilities, and thus between predicted β_j values in the lower range, will be achieved by giving students more choices at each decision point.

3.5 Psychometric Properties of the Model

Monotonicity and Decision Difficulty Ordering

As a measure of capability within a problem space, we desire specific relationships between β_j and the probability of selecting an 'optimal action' for any given state. Recall that within the MDP framework, $\pi(s)$ is defined as the set of optimal actions for state s. If a person selects action a while in state s, we can define the probability that they selected an optimal action as $p(a \in \pi(s)|s, \beta_j)$.

For a continuous measurement model, one expects monotonicity between β_j and $p(a \in \pi(s)|s, \beta_j)$, meaning that as β_j values increase, the probability of selecting an optimal action should be non-decreasing. If we define $\Delta Q(s, a'|\beta_j) = Q(s, \pi(s)|\beta_j) - Q(s, a'|\beta_j)$, we note

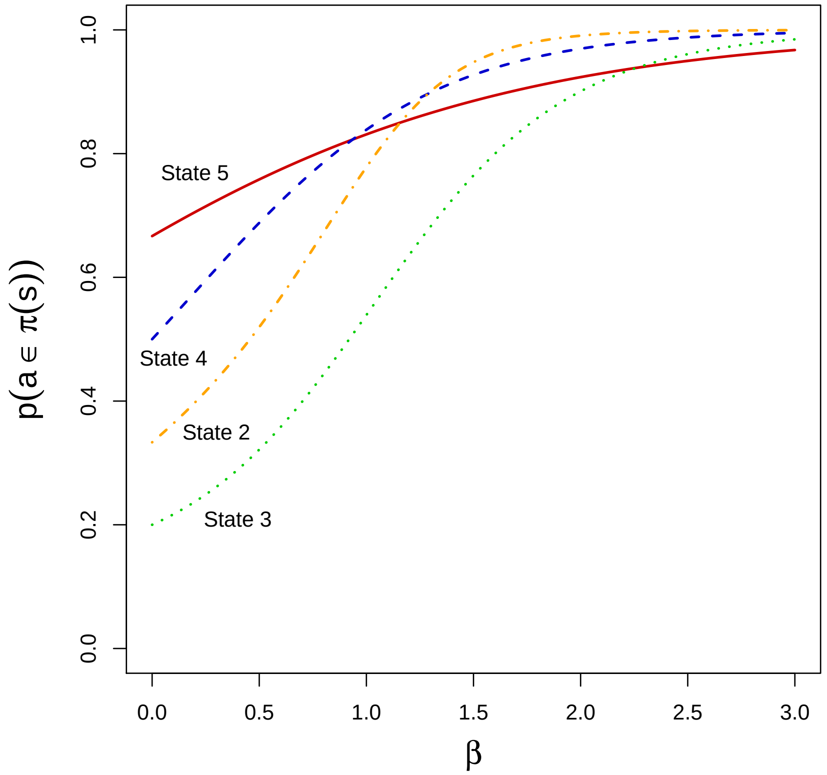


Figure 3.2: The probability of selecting an optimal response as a function of capability, β_j, for the game states 2 through 5 corresponding to Figure 3.1

that the log odds of choosing an element of $\pi(s)$ rather than a' is

$$\begin{aligned}
log\left(\frac{p(\pi(s)|s,\beta_j)}{p(a'|s,\beta_j)}\right) &= log\left(\frac{\exp(\beta_j Q(s,\pi(s)|\beta_j))}{\exp(\beta_j Q(s,a'|\beta_j))}\right) \\
&= log\left(\frac{\exp(\beta_j[Q(s,a'|\beta_j)+\Delta Q(s,a'|\beta_j)])}{\exp(\beta_j Q(s,a'|\beta_j))}\right) \\
&= log\left(\frac{\exp(\beta_j Q(s,a'|\beta_j))\exp(\beta_j \Delta Q(s,a'|\beta_j))}{\exp(\beta_j Q(s,a'|\beta_j))}\right) \\
&= log\left(\exp(\beta_j \Delta Q(s,a'|\beta_j))\right) = \beta_j \Delta Q(s,a'|\beta_j)
\end{aligned} \tag{3.12}$$

For a state s whose value does not depend upon future actions, i.e. a terminal state, the Q function does not depend upon β_j and the log-odds of choosing $\pi(s)$ increases as β_j increases. For any other state, ΔQ depends upon β_j only because β_j affects the probability of selecting optimal actions in the future. It can be shown that increasing β_j always increases the value of future states, but the interplay of the transition probabilities and the reward structure complicates the proof that ΔQ would always increase as β_j increases in every state. Instead, we suggest that the weaker criterion of convergence to an optimal action as β_j increases should suffice to ensure that β_j can be interpreted as a measure of the capability to find an optimal solution. Formally then we require that as $\lim\limits_{\beta_j \to \infty} p(a \in \pi(s)|s, \beta_j) = 1.0$.

This requirement is easily satisfied by the MDP model. We note that the probability of selecting an action from a finite set of actions A is

$$p(a|s, \beta_j) = \frac{\exp\left(\beta_j Q(s, a|\beta_j)\right)}{\sum_{a' \in A} \exp\left(\beta_j Q(s, a'|\beta_j)\right)}.$$

The set of actions $\pi(s)$ is defined as actions which maximize $Q(s, a|\beta_j)$. We refer to an element of this set as $a_{\pi(s)}$. For each $a' \notin \pi(s)$ there exists a constant $c_{a'} > 0$ such that

$$Q(s, a'|\beta_j) = Q(s, a_{\pi(s)}|\beta_j) - c_{a'}.$$

Out of the action set A, we define the number of optimal actions in state s to be $N_\pi = |\pi(S)|$, now the probability of selecting a particular $a_{\pi(s)} \in \pi(s)$ is

$$\begin{aligned} p(a_{\pi(s)}|s, \beta_j) &= \frac{\exp\left(\beta_j Q(s, a_{\pi(s)}|\beta_j)\right)}{\sum_{a' \in A} \exp\left(\beta_j Q(s, a'|\beta_j)\right)} \\ &= \frac{1}{\sum_{a' \in A} \dfrac{\exp\left(\beta_j Q(s, a'|\beta_j)\right)}{\exp\left(\beta_j Q(s, a_{\pi(s)}|\beta_j)\right)}} \\ &= \frac{1}{N_\pi + \sum_{a' \notin \pi(s)} \dfrac{\exp\left(\beta_j Q(s, a_{\pi(s)}|\beta_j) - \beta_j c_{a'}\right)}{\exp\left(\beta_j Q(s, a_{\pi(s)}|\beta_j)\right)}} \\ &= \frac{1}{N_\pi + \sum_{a' \notin \pi(s)} \dfrac{\exp\left(\beta_j Q(s, a_{\pi(s)}|\beta_j)\right)}{\exp\left(\beta_j Q(s, a_{\pi(s)}|\beta_j)\right) \exp \beta_j c_{a'}}} \\ &= \frac{1}{N_\pi + \sum_{a' \notin \pi(s)} \dfrac{1}{\exp \beta_j c_{a'}}} \end{aligned}$$

As $\beta_j \to \infty$, $\dfrac{1}{\exp\left(\beta_j c_{a'}\right)} \to 0$ for each $a' \notin \pi(s)$. Thus

$$\lim_{\beta_j \to \infty} p(a_{\pi(s)}|s, \beta_j) = \frac{1}{N_\pi}$$

The probability of selecting an action that is contained in the set of optimal action is

$$p(a \in \pi(s)|s, \beta_j) = \sum_{a_{\pi(s)}} p(a_{\pi(s)}|s, \beta_j)$$

so

$$\lim_{\beta_j \to \infty} p(a \in \pi(s)|s, \beta_j) = \sum_{a_{\pi(s)}} \frac{1}{N_\pi} = 1.$$

While it is reassuring that increased capability converges to optimal action choice, we must be careful when we define the MDP model for a problem space, that the mathematical definition of optimal matches the intended best path(s) through the problem. For example, while it is tempting to interpret the discount parameter γ cognitively, as a decreased ability to foresee the value of future actions, within the traditional settings for the MDP framework, γ is not a person parameter but instead represents real decrease in future rewards, such as are common in economics. If γ is set below 1, then there can exist situations in which the optimal policy is to take a shorter path to a lower final reward rather than taking the longer path to the highest final reward. If we expect the highest capability students to achieve the highest score, then we need to either ensure that all path lengths are equal or set $\gamma = 1.0$. A similar problem can arise from setting a negative reward value for actions that move the student closer to the reward. This will be discussed in more depth under parameter interpretation.

Another interesting property of psychometric models is the ordering of decisions, or items, by difficulty. Here we define decision difficulty as the probability of giving an optimal response, or $p(a \in \pi(s)|s, \beta_j)$, with a lower probability indicating a more difficult decision. When difficulty ordering is invariant across capabilities, the model is said to possess the property of “double monotonicity” (Skrondal & Rabe-Hesketh, 2007). Rasch models, for example, exhibit double monotonicity and the property is considered desirable for parameter interpretation. For 2PL IRT models, on the other hand, difficulty ordering can vary by capability. For our MDP model, double monotonicity would imply that if $p(a \in \pi(s_A)|s_A, \beta_j = c) > p(a \in \pi(s_B)|s_B, \beta_j = c)$ then for all c', $p(a \in \pi(s_A)|s_A, \beta_j = c') \geq p(a \in \pi(s_B)|s_B, \beta_j = c')$. From the example probability graph (Figure 3.2) we see that the MDP model does not give decision difficulty ordering invariance. However the crossing of the $p(a \in \pi(s)|s, \beta_j)$ lines is largely due to the fact that decision difficulty near zero is dominated by the number of options, while decision difficulty at large values of β_j is dominated by the differences in the $Q(s, a|\beta_j)$ value.

To understand better how the Q values are related to decision difficulty, we will examine some hypothetical decisions, all of which have the same number of options. First it should be noted that the decision difficulty is not affected by the overall magnitude of the Q function. This is clear from the fact that one could add an arbitrary constant C to the all Q values in Equation 3.6, and it would factor out of both the numerator and denominator, thus canceling. The difficulty, then, is determined by the difference between the Q values of the available options, as defined by $\Delta Q(s, a'|\beta_j)$ above. Given a decision with only two options and for which $\Delta Q(s, a'|\beta_j) = \Delta Q(s, a')$ we can compare the decision characteristic curves for four such decisions of differing values of $\Delta Q(s, a')$ (Figure 3.3). High contrast decisions, such as $\Delta Q(s, a') = 4.0$ provide high distinction between capabilities near zero, but do little to distinguish the higher capability students. The low contrast decisions, on the other hand,

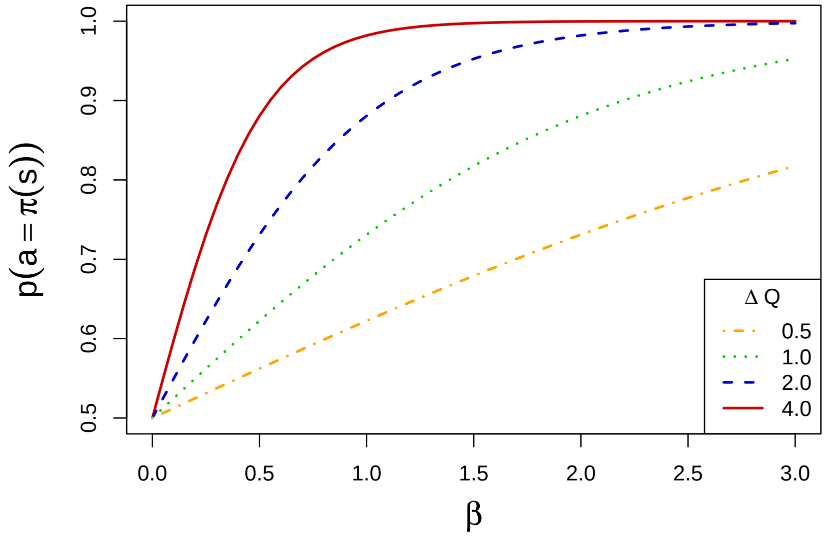


Figure 3.3: The probability of selecting an optimal response as a function of capability, β_j, for 2-option decisions of differing $\Delta Q(s, a')$

result in smaller differences in the action probabilities, but continue to distinguish students into the high β_j values.

Parameter Interpretation

Three elements affect the overall probability of choosing an optimal action. First the β_j value directly affects choice probability in the decision model, Equation 3.6. Second the differences between the Q values for the available actions affect the choice probabilities. These differences are themselves affected by two parts of the model, the β_j values as they affect the probability of future optimal action, and finally the reward values which provide a scale for Q. Note that the capability parameter β_j has two mechanisms by which it affects the action probabilities. We refer to the effect of β_j as it acts through the Q function as the *indirect* effect of β_j, contrasted with the *direct* effect of β_j as a parameter in the decision model. Each of these three elements can be interpreted cognitively. The direct effect of β_j models the student's ability to make a correct decision given their understanding of the problem. The indirect effect of β_j models the student's ability to understand what the effects of their actions will be in the future. Note that the farther the away goal is, the more diluted the Q values become for a low β_j student. This maps to an intuitive understanding that higher capability students would be able to 'read' farther into the problem than lower capability students. Finally, the reward structure models the motivation of the students.

As reward differences increase, the ΔQ values increase leading to an increased probability of choosing an optimal action. Further, a larger move penalty will cause lower capability students to gravitate to shorter paths through the problem space, even if they do not result in the highest final score.

Identification

If we wish to attach meaning to point estimates from a statistical model, we need to ensure that the model is identified, which is to say that there exists a single best set of estimates for the various parameters. Our model has two sources of potential identification problems. First, in divide-by-total models, the logit is invariant to translation, meaning that an arbitrary constant c could be added to all values $\beta_j Q(s, a|\beta_j)$ for a given decision without changing the decision probabilities. Second, because of the multiplicative relationship between β_j and $Q(s, a|\beta_j)$, an arbitrary factor could also be introduced to both values as

$$\beta_j Q(s, a|\beta_j) = (c\beta_j)\left(\frac{Q(s, a|\beta_j)}{c}\right). \tag{3.13}$$

As we are using a log-normal β_j, multiplying β_j by a constant is equivalent to an additive translation of the log-transformed normal variable λ_j

$$c\beta_j = c\exp(\lambda_j) = \exp(\ln c + \lambda_j) = \exp(\lambda_j + c'), \tag{3.14}$$

thus resulting in a shift of the μ parameter but not affecting σ. Note that multiplying β_j by a factor c would also affect $Q(s, a|\beta_j)$ in many states due to the dependence on β_j. Thus this is not a pure identification problem, but may result in weak identifiability, causing problems with the estimation algorithms.

To produce an clearly identified model, therefore, we will need to constrain either the scale and location of the β_j distribution or the scale and location of the $Q(s, a|\beta_j)$ distribution. If we were to constrain the β_j distribution, we would have to constrain two parameters, to achieve both the additive and multiplicative identification, which would leave little of interest to estimate about the student capability distribution. Therefore, we choose to apply the constraints to the distribution of the $Q(s, a|\beta_j)$ values. First we fix the maximum $Q(s, a|\beta_j)$ to give us the location constraint. This is straight-forward in an MDP, as generally there is an end goal with a large pay-off which drives the process. For the peg solitaire game, for example, we would fix R_{win} to a set value to provide this constraint. Next we need to constrain the scale of $Q(s, a|\beta_j)$ by fixing an interval within the range of possible values. Again, depending upon the parameterization of R, this can usually be easily handled by setting a second parameter within the reward structure. For the peg solitaire game, we would fix the penalty for leaving an extra peg, R_{peg}, which fixes the difference between scoring the present board state and taking another move when one is available.

When we consider the identifiability of a model, we should note that the issue is not a mere statistical technicality. Model identification problems indicate the limits of the inference we can make from data. The linear identification problem, present in many IRT based models, indicates that we cannot infer absolute student capability, but only relative student capability. Our second identification problem is interesting because it involves an interplay between the reward parameters and the capability parameter. If these were each single parameters, we would conclude that we cannot distinguish between capability and motivation. In fact, most educational assessment models are unable to distinguish between capability and motivation. While motivation is not even parameterized in common assessment models, there is an implicit assumption that either students are highly motivated so that we can be sure we are measuring ability, or that we are actually measuring the combination of ability and motivation. In particular, the highly-motivated low-ability student and the poorly-motivated high-ability student are generally indistinguishable. An MDP model, however, might be able to make that distinction. While we are forced to constrain two parameters of the peg solitaire reward structure, there remain two free parameters, R_{move} and R_{reset}. As the R_{move} parameter signifies the subjective cost of taking an action, it seems particularly related to motivation. A student who is uninterested in playing the game might consider each addition move a significant expenditure of effort. This would be modeled by a large negative R_{move} value and would result in the student tending to choose shorter paths through the problem space. These students would be more likely to quit early and less likely to find the highest final reward state. Highly motivated students, on the other hand, would be modeled by small negative R_{move} values, resulting in play that takes more actions to achieve a better final reward. We will evaluate the ability of our MDP model to distinguish between capability and motivation, as indicated by R_{move}, in the second simulation study.

3.6 Simulation Studies

Simulation studies were performed to evaluate the performance of the MDP model for measurement by examining parameter recovery under controlled conditions. The peg solitaire game explained in Section 3.4 was used for all the simulations. This task is appealing because it is a pure strategy game with no hidden information and can easily be scaled in complexity by increasing the board size or changing the starting configuration. There is no stochastic component to the game, in that all state transitions are deterministic. While the MDP approach works well with probabilistic systems which include an element of chance in the outcomes, a deterministic game is more comparable to current assessments in which the outcomes are determined directly by student actions.

Simulation study 1 evaluated basic recovery of the population capability parameters, μ and σ and the student capability parameter, β_j, under ideal conditions. For these simulations the model reward parameters, R are all fixed at favorable, consistent values for both the generating and estimating models. In study 2, we evaluated the distinguishability of capability and motivation at the population level by estimating both the capability distribution and the free reward parameters in four distinct simulated samples.

For all of the simulations, the model parameter and capability estimates were compared to the generating parameters and evaluated using the bias and root mean squared error

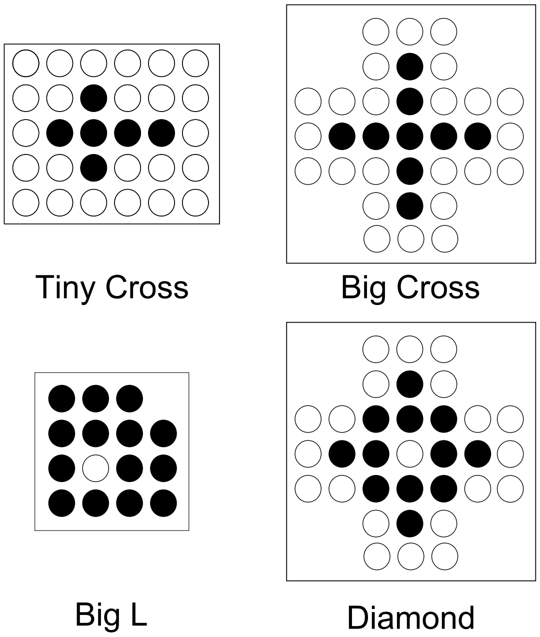


Figure 3.4: Starting positions for the game boards used for the simulation study.

(RMSE).

Simulation Design

Four different game boards were used to examine the effect of task complexity on parameter recovery (Figure 3.4). They varied in game length from the Tiny Cross board, which can be solved in 5 moves, as was shown in Figure 3.1, to the Big-L board which requires 13 moves to win (Table 3.5). While longer solution paths are an indication of complexity, the number of meaningful choices at each move also contributes to complexity. For example, a long path with no branching will probably be easier to solve than a shorter path with multiple options at each choice point. In game theory, complexity is often measured using either the size of the reachable state space, which is the total number of game states that can be reached from the initial position, or the number of leaf nodes, which is the total number of game states in which no further moves (other than *score* and *reset*) are possible. We can also look at the total number of distinct allowable moves as an indication of game complexity. Table 3.5 shows that of the selected game boards, the Big-L board has the longest path to solution, but the Diamond board has higher values on every other complexity measure.

Play records were simulated by using the MDP as a generating model. For all simulations, γ was fixed at 1.0 to reflect the temporal invariance of the value of winning. R_{win} was fixed

Table 3.5: Complexity measures for the game boards used in the simulation studies.

Board Name	Solution Path Length	Reachable States	Leaf Nodes	Move Actions
Tiny Cross	5	22	11	12
Big Cross	8	153	42	22
Big-L	13	807	47	30
Diamond	11	5923	1454	70

at 5.0 and R_{peg} was fixed at 1.0 for identification purposes. Given generating parameter values for μ, σ, R_{move} and R_{reset}, student samples were generated as

$$\beta_j = \exp(\lambda_j); \quad \lambda_j \sim N(\mu, \sigma). \tag{3.15}$$

To simulate student game records on each board, the MDP was solved for each simulated value of β_j, yielding the appropriate Q values for each possible game state. For each game board, our simulated students started in the starting state, the original board configuration, and then drew their next action from the available actions according to the probabilities given in Equation 3.6. If the action was to score the game, the game play was finished, if not, the next state was calculated based on the selected action. This was repeated until game completion.

Study 1: Recovery of Population and Person Parameters

This simulation study evaluated the recovery of the population parameters, μ and σ and the person parameters β_j. A single sample of 200 students was simulated from a population with $\mu = 0.0$ and $\sigma = 0.75$. The reward parameters were set to $R_{move} = -0.1$ and $R_{reset} = -1.0$. The students in this simulated sample played each of the four game boards 50 times, giving a total of 4,000 game records. This design allows us to more easily evaluate the bias and standard error of the β_j estimates as well as to directly compare the information gained from the differing game boards. As multiple samples were not simulated, the study does not provide information about how the estimates might be affected by differing sample characteristics.

As described in Section 3.3, the population parameters, μ and σ were estimated using maximum marginal likelihood, then the individual capability parameters, β_j were estimated using maximum a-posteriori (MAP) with the estimated population distribution $\ln N(\hat{\mu}, \hat{\sigma}^2)$ used as the prior. For all estimations the reward parameters, R, were fixed to the same values as in the generating model. Estimated biases and RMSE values were calculated relative to the values from generated sample.

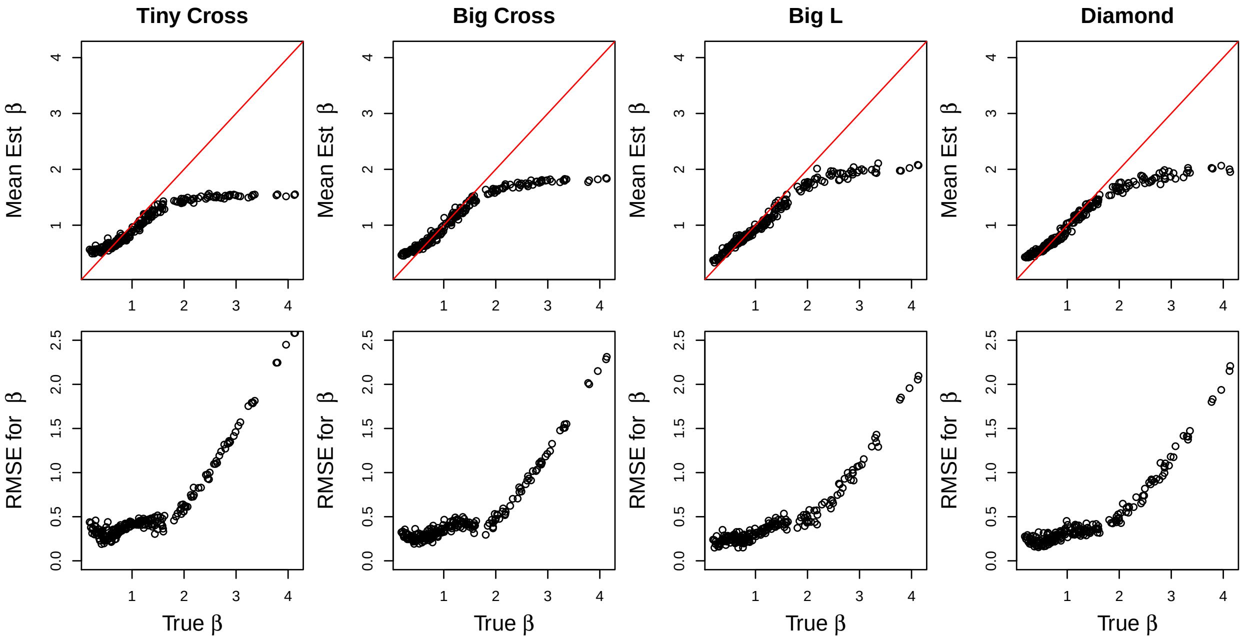


Figure 3.5: Top row: Mean $\hat{\beta}_j$ by true β_j. Bottom row: RMSE of β_j by true β_j.

Results

Population parameter recovery was fairly good as shown in Table 3.6. There was a slight positive bias in both $\hat{\mu}$ and $\hat{\sigma}$, but the RMSE for both parameters remained small. Both the bias and RMSE for μ remained fairly consistent across different board complexities, but the errors for σ do show a decrease for the two more complex boards.

Table 3.6: Population parameter recovery with fixed reward values for four different game boards, standard errors are shown in parentheses.

	μ		σ	
Board	Bias	RMSE	Bias	RMSE
Tiny Cross	0.017 (0.008)	0.055 (0.023)	0.103 (0.029)	0.217 (0.102)
Big Cross	0.028 (0.007)	0.053 (0.024)	0.076 (0.014)	0.121 (0.053)
Big-L	0.019 (0.008)	0.058 (0.034)	0.056 (0.011)	0.093 (0.049)
Diamond	0.012 (0.009)	0.058 (0.033)	0.056 (0.013)	0.105 (0.046)

Recovery of the person parameters, β_j can be seen as a function of true β_j in Figure 3.5. For the lower values of β_j, the mean of the estimates track the true value of β_j quite well, as they generally lie upon the identity line as shown on the graphs. As β_j gets large however, the estimates level off, showing increasing bias and RMSE. The error in the estimates for these high capability students appears to be due to ceiling effects as students who perform perfectly on the task cannot be distinguished.

To determine the point at which β_j values give a high probability of perfect play, we analyzed the simulated play records for all of the simulation runs. Plays which resulted in the maximum score, without once resetting the board, were considered ‘perfect play.’ We calculated a ceiling threshold based on the true β_j value above which students produced perfect plays at least 60% of the time. These thresholds varied by game board, but ranged between 2 and 2.6 logits, as shown in the fourth column of Table 3.7. The simulation results data were then truncated by removing students that fell above the threshold on each board. The number of students removed in this process was $\leq$ 20%, as shown in column five of Table 3.7.

Table 3.7 shows the $\hat{\beta}_j$ simulation results for the full population and the truncated populations. While the bias and RMSE for the full population are quite large, the truncated population show little bias and reasonable RMSE for β_j. The difference in ceiling thresholds among the boards suggests that the ceiling effect is task dependent. The highest threshold is for the Big-L board which is also the board with the longest solution path, indicating that record length, rather than state space size more closely affects the amount of ceiling effect in the estimates.

Table 3.7: Recovery of β_j with fixed reward values for four different game boards, with and without truncation for perfect play.

	Full Population		Ceiling	Students	Truncated Pop.	
Board	Bias	RMSE	Thresh.	Remaining	Bias	RMSE
Tiny Cross	-0.318	0.738	2.03	0.80	-0.064	0.395
Big Cross	-0.228	0.631	2.33	0.84	-0.036	0.362
Big-L	-0.208	0.571	2.62	0.88	-0.072	0.365
Diamond	-0.220	0.587	2.28	0.84	-0.045	0.327

Study 2: Recovery of Reward and Population Model Parameters

For this study we examined separability of capability from motivation at the population level. In particular, we were interested in whether the model could distinguish between highly-motivated low-ability populations and poorly-motivated high-ability populations.

Design

Four samples were generated to simulate populations that were of differing capability and motivation. The generating parameters are shown in Table 3.8. For capability, we varied μ between the groups. A μ value of 0.5 gives a median β_j value of 1.65 while $\mu = -0.5$ gives a median β_j of 0.61. There is still considerable overlap between the distributions of the high and low capability groups. For motivation, we varied the move cost, R_{move}. The value of

R_{move} should be considered relative to the potential gain for taking a single move, R_{peg} which is fixed at 1.0. The high motivation groups have a very low move cost of -0.05 which would encourage them to keep trying after making a mistake. For the low motivation students, on the other hand, the cost of a move is half of the expected gain from the move, given perfect future play. For low capability students, perfect play is unlikely, so this reward value should discourage continued attempts after errors. R_{reset} was fixed to -1.0 in all conditions. In all cases $N = 200$ students were simulated and each simulated student played each of the four game boards 25 times.

Table 3.8: Generating parameters for the four samples used in simulation study 2

Sample	Capability	Motivation	μ	σ	R_{move}	R_{reset}	R_{win}	R_{peg}
1	High	High	0.5	0.75	-0.05	-1.0	5.0	1.0
2	High	Low	0.5	0.75	-0.75	-1.0	5.0	1.0
3	Low	High	-0.5	0.75	-0.05	-1.0	5.0	1.0
4	Low	Low	-0.5	0.75	-0.75	-1.0	5.0	1.0

For each set of game records, the population parameters and reward parameter were simultaneously estimated. Person β_j parameters were then estimated using the MAP estimators as described previously.

Results

The differences between the four populations were evident immediately in the simulated game records. As predicted, the higher motivation students reset the game more often (Table 3.9), especially among the lower capability group, and thus had longer game records. The mean total score, however, is less distinctive. In particular the mean total score of the high-ability low-motivation group is quite similar to that of the low-ability high-motivation group.

Table 3.9: Mean number of resets and total score per dataset by capability and motivation conditions.

Sample	Capability	Motivation	Mean # Resets	Mean Total Score
1	High	High	85.7	3.77
2	High	Low	3.6	0.73
3	Low	High	105.7	0.95
4	Low	Low	19.9	-1.07

Looking at results for individual data sets (Figure 3.6), we see that mean total score is insufficient for distinguishing between the high-ability low-motivation (HALM) and the low-ability high-motivation (LAHM) groups. In fact there is even considerable overlap between the low-ability low-motivation (LALM) group and the other two. Only the high-ability high-motivation (HAHM) group stands out as truly distinct. The estimates from the MDP

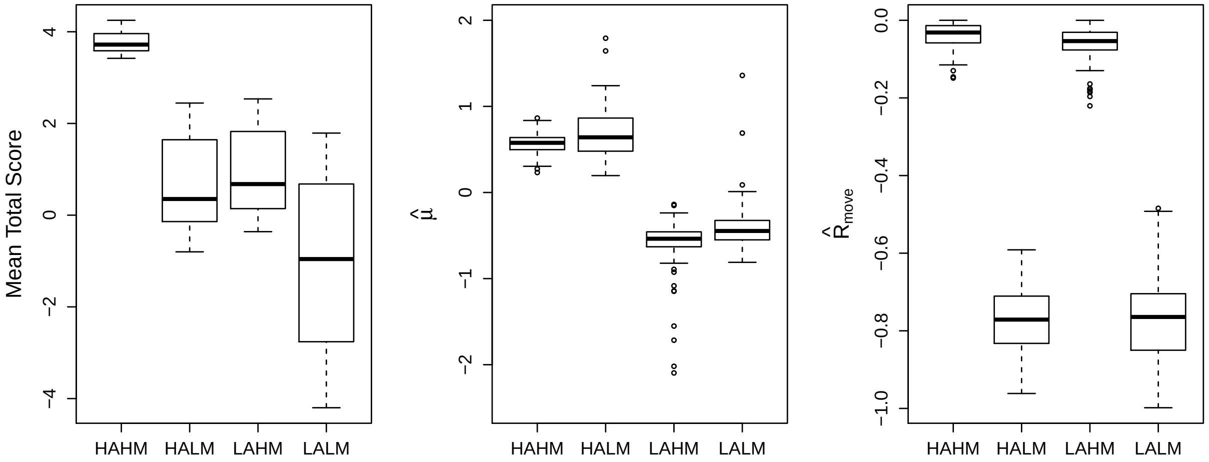


Figure 3.6: Distribution of per-dataset mean total score, estimated μ and estimated R_{move} by experimental condition, where HAHM is "High Ability High Motivation, HALM is "High Ability Low Motivation", LAHM is "Low Ability High Motivation" and LALM is "Low Ability High Motivation".

model, on the other hand, are able to distinguish all groups. Figure 3.6 shows how the distributions of $\hat{\mu}$ are mostly non-overlapping between the high and low ability groups while the distributions of $\hat{R}_{move}$ are completely distinct between the high and low motivation groups. In Figure 3.7 we can see how the two parameters together are sufficient to cleanly distinguish the sample data sets into the four experimental groups. The only estimates which might be misclassified are the few LALM students with very low $\hat{R}_{move}$ who also had high $\hat{\beta}_j$ values. These can be seen in the upper left part of Figure 3.7 closer to the HALM cluster than to the LALM cluster.

The mean estimates for the population level parameters (μ, σ, and R_{move}) are shown in Table 3.10. These aggregated estimates are very close to the true values from the sample. For the high-ability sample, mean$(\beta_j) = 0.56$ and sd$(\beta_j) = 0.69$ while for the low-ability sample, mean$(\beta_j) = -0.48$ and sd$(\beta_j) = 0.72$. The high-motivation data sets were generated with $R_{move} = -0.05$ and the low-motivation sets had $R_{move} = -0.75$.

While the recovery of R_{move} is fairly consistent across conditions, the recover of μ and σ vary considerably by experimental condition (Table 3.11). The bias and RMSE of μ appears to be best for the HAHM group and worst for the LAHM group. For σ, however, both low ability groups perform considerably better than the high ability groups, with HALM giving the most accurate estimates for σ. Interestingly, while the estimation of R_{move} was good overall, the estimation performed poorest for the HALM group. This may indicate that there is still some amount of scale trade-off between the β_j distribution and the reward

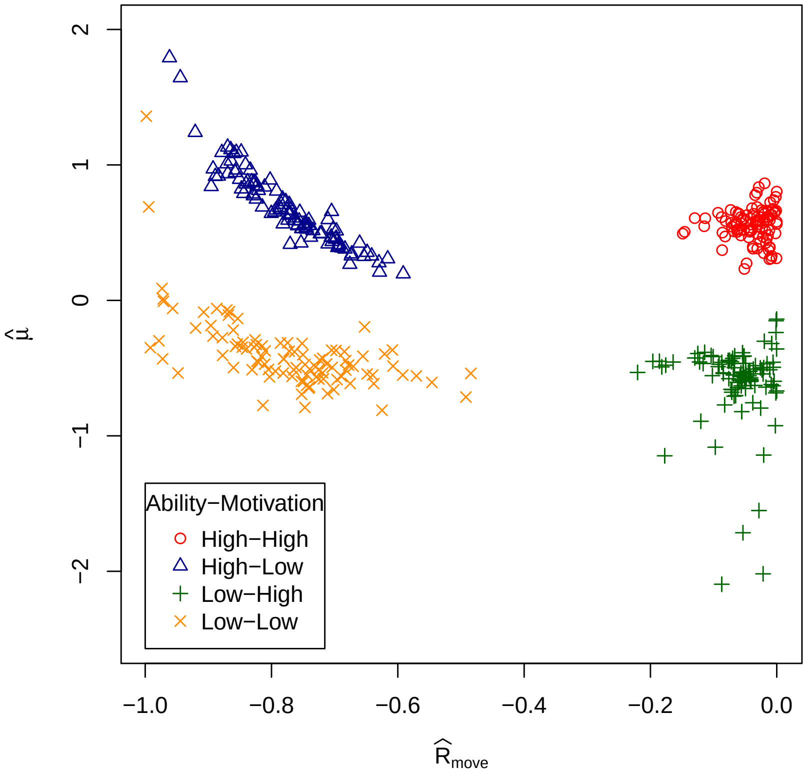


Figure 3.7: Sample group clustering shown in $\hat{\mu}$ and $\hat{R}_{move}$ space. Each condition includes 200 data sets.

Table 3.10: Mean parameter estimates across all four boards and 25 runs, by experimental group, with standard deviation in parentheses.

Sample	Capability	Motivation	μ	σ	R_{move}
1	High	High	0.559 (0.122)	1.016 (0.879)	-0.039 (0.032)
2	High	Low	0.681 (0.279)	0.743 (0.192)	-0.774 (0.077)
3	Low	High	-0.639 (0.482)	0.908 (0.590)	-0.062 (0.047)
4	Low	Low	-0.400 (0.276)	0.900 (0.514)	-0.774 (0.111)

parameters. The difference in complexity introduced by the four different game boards turned out to not significantly affect parameter recovery.

We note that for even the best performing condition in this study, the estimation errors are higher than those achieved in study 1. This is to be expected as we are estimating more parameters, but it is a concern as we attempt to expand the role of the measurement model to encompass measurement of motivation and perhaps beliefs through estimation of reward

Table 3.11: Parameter recovery over all replications and all boards by experimental group.

			μ		σ		R_{move}	
Sample	Capability	Motivation	Bias	RMSE	Bias	RMSE	Bias	RMSE
1	High	High	-0.005	0.122	0.324	0.932	0.011	0.034
2	High	Low	0.117	0.301	0.050	0.197	-0.024	0.080
3	Low	High	-0.155	0.504	0.183	0.615	-0.012	0.048
4	Low	Low	0.084	0.288	0.176	0.540	-0.024	0.113

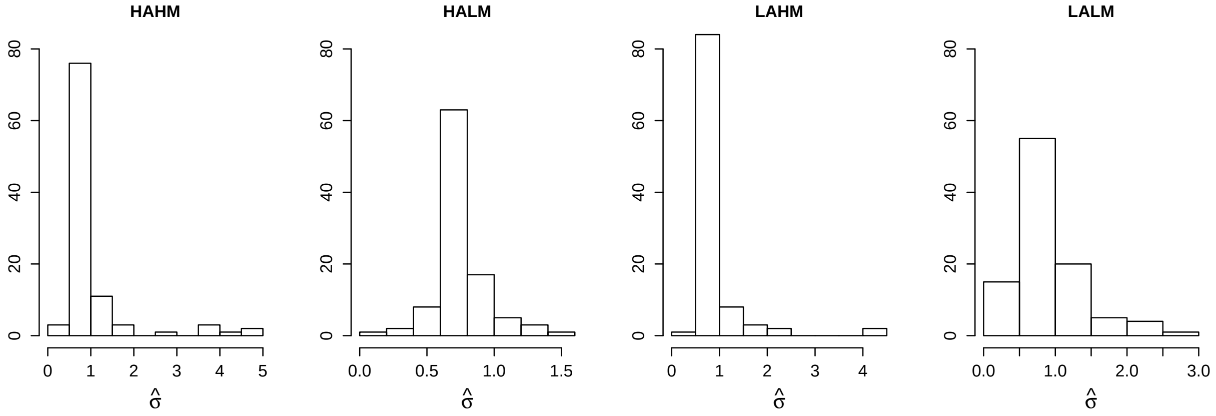


Figure 3.8: Histograms of $\hat{\sigma}$ for the four different conditions.

and transition function parameters.

An examination of the results from individual estimation runs uncovered an interesting phenomenon in the estimation. For the high-motivation groups, a handful of data sets were estimated to have a very large σ value (Figure 3.8). These data sets were also accompanied by a large error in the estimation of μ. The σ parameter for a log-normal distribution controls the shape of the distribution. As σ gets large, the distribution becomes more and more concentrated near zero. Thus a gross over-estimation of σ will have the effect of pushing the $\hat{\beta}_j$ to near zero when using Bayes modal estimators. Figure 3.9 shows the estimates of β_j versus the true β_j values for two different runs on the Diamond board for the HAHM group. Repeated estimation runs on one of the data sets which produced a high σ estimate did not consistently reproduce the erroneous estimate, due to randomness in the estimation algorithms. In fact an estimate of $\hat{\sigma} > 3$ was produced less than 1 in 5 runs and always resulted in a much smaller log-likelihood than the estimation runs that produced smaller estimates for σ. This seems to indicate that oddly high $\hat{\sigma}$ may be failure of the estimation algorithm, akin to a convergence failure, rather than a valid parameter estimate for the data.

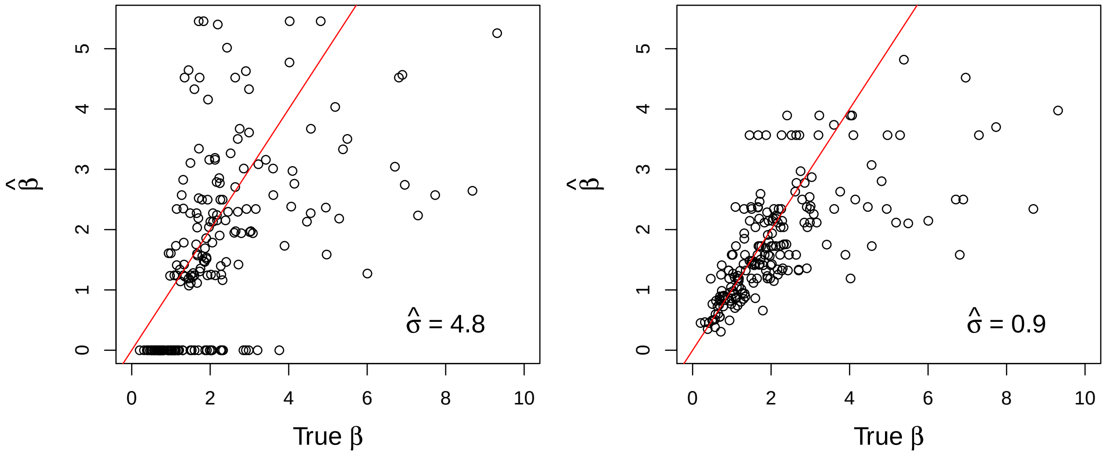


Figure 3.9: Estimated versus true β_j for two different runs of the high-ability high-motivation condition on the Diamond board.

3.7 Discussion

In this chapter we've explored the use of a Markov decision process as the basis for a measurement model in the context of complex strategic problems. The model departs from standard psychometric models in several aspects. Decision difficulty is encoded in the differences between the values of the available options rather than in a separately estimated parameter. Further, this decision difficulty has a multiplicative relationship with the latent capability parameter, rather than a linear relationship. These factors lead to the constraint that the latent capability parameter be non-negative, and we have chosen here to model it with a log-normal distribution. Perhaps most unusual, the capability parameter is itself a component of the decision difficulty calculation, making decision difficulty ability-specific.

As a cognitive model for decision making, the MDP includes elements that correspond to motivation, beliefs and problem-solving capability. Because these elements each affect action probability in different ways, estimation of motivation and beliefs are naturally separable from an estimation of problem-solving capability. Both goals and beliefs have been estimated using MDP models before (Ng & Russell, 2000; Baker et al., 2011; Rafferty et al., in press), but in those cases the focus was on classifying agents by their goals and beliefs and the Boltzmann parameter, β, was taken to be a nuisance parameter at most. This present study focused on the Boltzmann parameter as a measure of problem-solving capability and further explored how the MDP might be used to separate commonly the confounded latent traits of capability and motivation.

As a consequence of restricting β_j to be non-negative, the probability of choosing an optimal action was constrained to be uniform random chance at worst. When used as a cognitive model, such a constraint makes sense, as it is assumed that the student goals and beliefs are correctly represented by the MDP specification. In an educational setting, however, it is more than likely that some students will have misconceptions about the content, mechanics or goals of the task. These misconceptions could cause the student to consistently select non-optimal actions if the MDP has been specified to reflect a correct interpretation of the task. Thus one limitation of the current model is that it does not properly model the actions of students with misconceptions. The model might be extended to allow for different MDP specifications for different sub-populations, as a mixture model, thus classifying student by misconception while measuring their problem solving capability, similar to the model presented in Chapter 2.

The first simulation study showed that the MDP model was able to recover both the population distribution of β_j and individual β_j values up to a ceiling threshold at which students were indistinguishable due to perfect responses. This finding presents one piece of evidence that an MDP might be used as an informative measurement model, provided that the MDP is, in fact, a good generative model for student performance. Further validity evidence must rely on analysis of actual student data so that estimates can be compared to external measures and student self-reports. If the validity of the MDP measurement models can be fully established, they may proof to be of value in analyzing performance tasks, for which traditional approaches require many replicate outcome measures to produce a reliable estimate.

One puzzling result from the first simulation study was how little game complexity mattered in the recovery of the parameters. While the larger boards had lower RMSE for σ, the RMSE and bias for μ and β_j were little affected by game board. Complexity did appear to affect the difficulty of the task however, as the ceiling thresholds for perfect-play were higher for the more complex boards. This finding seems to indicate that the measurement gain from increasing the complexity of tasks may lie primarily in accurately measuring more of the upper capability students, though further exploration of this issue is needed.

The second simulation study showed that at the population level, both μ and R_{move} can be fairly accurately estimated. If these parameters are valid representations of student capability and motivation, such estimation could prevent confusion between the low-motivated and the low-ability students. In particular, we were able to clearly separate data sets that were generated under 'high-ability but low-motivation' conditions from those that were generated under 'low-ability and high-motivation' conditions – a separation that was not possible using the final outcome metric alone. The next step in this line of research would be to classify students into sub-populations based on motivation while still estimating their capability.

In the second simulation it was found that the estimation algorithm occasionally produced spurious results in the form of an unusually large estimation for the capability distribution parameter σ. This problem can likely be ameliorated through an improvement of the estimation methodology. For example multiple estimations might be run from different starting values so that the global maximum of the likelihood can be more reliably selected. Another possibility would be to use a Bayesian approach to the estimation of population parameters, such as MCMC. This would allow the specification of a prior distribution on σ which would have the effect of reducing unusually large estimates. A similar result could be achieved without using MCMC by adding a penalty term associated with high σ values. Estimation methodology was not the focus of this study, and future work could do much to make it more efficient and reliable.

While all demonstrations and simulations of the model in this study has occurred in the context of a simple board game puzzle, the MDP model is generalizable to many different problem-solving tasks. To model a new task with an MDP, the state space, action set, transition function and reward structure need to be specified. As a part of this study, a generalized MDP C++ library has been created which facilitates the specification of a task as an MDP, and provides an interface for parameter estimation. Still, modeling new tasks with this library requires writing C++ code. Future work might include producing an R package for MDP models, which would provide an interface within an environment generally more familiar to statisticians.

Chapter 4

Applying the Markov Decision Process Measurement Model to an Educational Game

There is increasing interest in combining educational games and assessment, either by including assessments seamlessly within games intended for learning (V. Shute, 2011) or by using the game primarily as an assessment, exemplified by the recent work at Glass Lab (Mislevy et al., 2014). In particular, simulation games which require students to solve complex problems within a somewhat realistic environment can be both challenging and engaging for the student. When used as an assessment, such programs can provide more data than the correct/incorrect scores supplied by traditional assessments - they enable access to a wealth of additional data about student performance, in the form of actions that the students take within the game (Shaffer & Gee, 2012). Not only do these tasks provide an opportunity for students to demonstrate higher level skills in planning and system thinking (V. J. Shute et al., 2010) but discrete actions within the complex performance reveal students' understanding of domain specific mechanisms.

Used as assessments, games are examples of complex performance tasks. While outcome metrics are available in terms of achieving predefined goals or "winning" different game levels, much of the available data relates to actions performed as part of the problem solving within a particular game challenge. Sometimes referred to as log-file data (Buckley et al., 2010), making use of the information contained in this type of data is an area of active research. When the game or simulation contains clearly definable "right" actions, measurement evidence can be collected by merely aggregating the number of occurrences of such desirable behaviors. In many cases, however, it is not the individual action, but the relationships between actions or the context within which an action was taken that contain possible evidence of student expertise.

As presented in Chapter 3, Markov decision processes (MDP) can be combined with item response theory to create a measurement model from these internal action data. While the model was developed and studied via simulation in that chapter, the complications of applying the model to an actual educational game have yet to be evaluated.

This chapter explores an application of the MDP measurement model to empirical data from an actual educational game. The study focuses on the feasibility and utility of the approach for real world data as well as providing a demonstration of how the model can accommodate the quirks and requirements of such a project. In the following sections I describe the application game, Microbes, review the MDP measurement model and show how the model can be applied to the Microbes game. I then fit the model to data from a pilot study and estimate person specific abilities. The ability estimates are compared to post-test scores for evidence of external validity. The performance of the MDP model is also compared to IRT models which rely solely on outcome data from the game. Finally I discuss the overall outcome, including what was successfully achieved, the difficulties encountered, how such difficulties might be overcome and the generalizability of the approach to similar educational games.

4.1 Microbes

Microbes is an educational game in which students learn about cell biology by playing the part of a microbe navigating through increasingly challenging environments (Red Hill Studios, 2009). Publicly available on the pbsKids website, the game is part of the Lifeboat to Mars series played by hundreds of students each month. For each of the ten levels, the student receives a description of the environment they are about to enter, is given an opportunity to configure their microbe by "buying" upgrades, and then attempts to navigate the environment to reach the goal without being eaten or running out of energy. To succeed at each level, the student must understand both the challenge presented by the environment and the features or behaviors that will enable the microbe to survive those challenges. If the student succeeds, they are given ten game tokens and move on to the next level. If they fail, they are encouraged to try again, reconfiguring their microbe if they wish.

The educational focus of the game is cellular energy production, so the microbe features of most interest are mitochondria, which facilitate the conversion of food to energy, and chloroplasts, which generate food from sunlight. In the configuration phase of the game, the student is able to buy mitochondria and chloroplasts using the tokens they have earned from previous levels. The first four levels of the game are tutorials in which the students learn to drive their microbe around the tank using the keyboard's arrow keys and recognize food and predators. These levels also ensure that the student has earned some tokens by the time they reach levels that require mitochondria (at level 5) and chloroplasts (at level 6).

The data analyzed here were collected as part of a pilot study in which 233 students played the game and then took a multiple choice post-test. For each student, play records were collected that list actions taken and results of the actions. Recorded actions include buying microbe features or playing a level, while the results record the outcome from each play-level attempt. The outcome from playing a level might be success, as when the microbe made it to the goal post, or failure if it died. In cases of failure, the cause of death is recorded

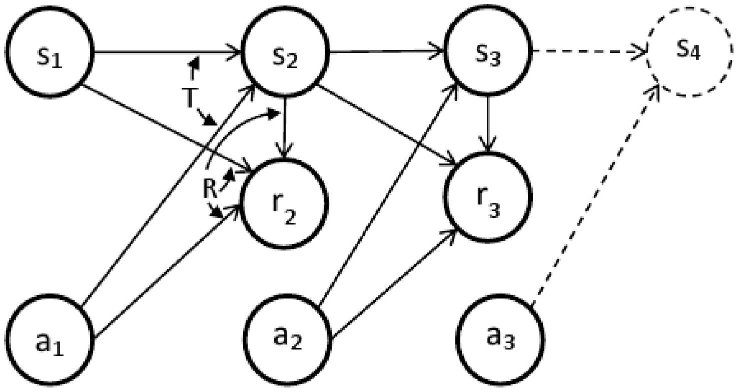


Figure 4.1: Markov decision process model.

as is the microbe's distance from the goal post at death. Information on exactly how the student navigated through the level is not available. The multiple-choice post-test covered topics of cell biology deemed to be similar in content to the Microbes game.

4.2 Markov Decision Processes

Markov decision processes (MDPs) model sequential decision making in a non-deterministic setting (Puterman, 1994). MDPs are often used to find an optimal sequence of actions through a state space, optimizing rewards that result from taking specific actions in specific states. The states are collections of variables that describe the relevant factors pertaining to the problem at hand, while the rewards reflect either cost or benefit gained from taking an action in a particular state and transitioning to the next state. The model is probabilistic in that the progression from one state to the next need not be determined completely by the actions taken in the previous state, but instead can be influenced by random effects. Thus the transition function between states is a probability distribution over the state space. The MDP model (Figure 4.1) is expressed in discrete time increments, $t \in 1, 2, 3, \ldots N$. When $N < \infty$ the model is known as a finite horizon MDP. At each time interval t, the system is in a particular state s_t and the agent takes an action a_t. The system then moves to state s_{t+1} and the agent incurs the cost or receives the benefit of that transition as r_{t+1}. Cognitively, the MDP depends upon the principle of rationality, assuming that agents will tend to choose the most beneficial action based on their goals and understanding of the system.

An MDP can be formally defined by it's components, $\{S, A, T, R, \gamma\}$. S is the set of possible states of the system and A is the set of actions that one can take. T represents the transition model, $p(s_{t+1}|s_t, a_t)$, the probability of transitioning to the state s_{t+1} given that action a_t was taken in state s_t. R corresponds to the reward structure $r(s_t, a_t, s_{t+1})$ which specifies the reward for taking action a_t in state s_t and entering state s_{t+1}, while $\gamma \in [0, 1]$ is the discount parameter, representing the relative value of future versus immediate rewards. From this specification, one can calculate the Q function, which is the expected sum of discounted rewards obtained by taking action a_t while in state s_t,

$$Q(s_t, a_t) = \sum_{s_{t+1} \in S} p(s_{t+1}|s_t, a_t) \left(r(s_t, a_t, s_{t+1}) + \gamma \sum_{a_{t+1} \in A} p(a_{t+1}|s_{t+1}) Q(s_{t+1}, a_{t+1}) \right). \quad (4.1)$$

$p(a_t|s_t)$ is the decision rule, or policy, by which actions are chosen given a particular state. In Equation 4.1, $r(s_t, a_t, s_{t+1})$ is the immediate reward for taking action a_t in state s_t assuming it results in state s_{t+1}, while $\sum\limits_{a_{t+1} \in A} p(a_{t+1}|s_{t+1}) Q(s_{t+1}, a_{t+1})$ is the expected value of the next state, marginalized over the possible next actions. Thus the quantity inside the large parentheses is the sum of the immediate reward and the discounted value of the future state. The expectation of this sum is then taken over all possible states s_{t+1} that might result from action a_t in state s_t. Note that the function is recursive, as the value of a state is defined using the Q function itself. The Q function can be calculated using dynamic programming (Howard, 1960).

Historically, MDPs have been used in the context of robotics or computer artificial intelligence to find the best actions to take in any given state, known as the optimal policy. More recently, Markov decision processes have been used as a cognitive model to describe not only human decision making, but also people's ability to infer the goals and beliefs of others. Baker, Saxe and Tenenbaum (2011), describe a "Bayesian theory of mind" in which cognition is modeled as a partially observed MDP. They hypothesize that people act based on their beliefs, modeled by the state space, action set and transition functions, and in accordance to their desires, which are modeled by the reward structure. In their earlier work, (2009), they used an MDP to show how one person might infer the goals of another through the observation of their actions. When modeling human decision making, the policy is not assumed to be optimal, as humans do make errors. Frequently a Boltzmann policy is used (Baker et al., 2009),

$$p(a_t|s_t) \propto e^{\beta Q(s_t, a_t)} \quad (4.2)$$

where $\beta \in [0, \infty)$ represents the decision maker's capability to optimize actions according to the Q function. As β increases, the probability of taking the action with the highest Q value increases. When β goes to zero, the action probabilities become equal, and actions are selected uniformly at random.

One important aspect of the MDP when used as a cognitive model is that the parameters S, A, T, R, γ all represent the decision maker's understanding of the problem space. If, for example, there are available actions unknown to the decision maker, those actions will not be be part of the action set in his personal formulation of the problem. This subjective quality of the model allows it to be used for making inferences about an agent's understanding of the problem based upon their actions (Baker et al., 2011; Rafferty et al., in press). In particular, the agent's goals are encoded in the reward structure, while their beliefs about system dynamics are encoded in the transition function. Inverse reinforcement learning utilizes MDPs to infer discrete goals based on action traces through an estimation of the most probable values of the reward function (Ng & Russell, 2000), while inverse planning algorithms infer student understanding of the effects of their actions based an estimation of parameters in the transition function (Rafferty et al., in press).

4.3 Markov Decision Processes for Assessment

To use the Markov decision process model for assessment, we are interested primarily in β as a measure of a student's capability to optimally solve a specific problem. All other parameters will either be fixed, when it can reasonably be assumed that the parameters affect all participants in a well understood way, or estimated at the population level with the assumption that all participants share a common understanding of the problem. The formulation of the Q function remains as in Equation 4.1, except that we note explicitly the dependency upon the capability parameter, transforming β into the person-specific latent variable β_j. The conditional probability of student j selecting action a_t when in state s_t now becomes

$$p(a_t|s_t,\beta_j) = \frac{\exp\left(\beta_j Q(s_t, a_t|\beta_j)\right)}{\sum_{a'_t \in A} \exp\left(\beta_j Q(s_t, a'_t|\beta_j)\right)}. \tag{4.3}$$

In practice we treat β_j as a random parameter when we are estimating the model from the population data. Because we wish β_j to remain positive we take $\beta_j \sim lnN(\mu, \sigma)$ (see Chapter 3 for an extended discussion). Thus our model parameters include the distributional parameters μ and σ.

The remaining MDP parameter space includes the transition function parameters, the reward-structure parameters and the discount parameter. These parameters might be fixed or estimated, depending upon the intended inference. As was discussed in Chapter 3, the values in the reward structure determine the magnitude and scale of the Q function, which must be constrained to produce an identifiable model. Thus even when the reward parameters are estimated, at least two of the parameters must be fixed for identification purposes.

4.4 Modeling the Microbes Game

To apply the MDP measurement model to the Microbes game, we must first model the Microbes game play as a Markov decision process. This involves defining the state space S, action set A, reward structure R, and the transition probabilities T. For a complex game which includes many different variables, the immediate question is how much of that complexity to include in the model. As our goal in this application is to provide a measurement of student understanding of microbe energy production, we chose to focus on state variables and actions that directly relate to those concepts. Another issue is how much of the game record to model at one time. We initially attempted to model complete game play records over all six game levels with a single MDP. This seemed attractive because these records would capture both decisions about actions taken within a level and decisions about the order in which the levels were played. Modeling these complete game records, however, proved problematic for two reasons. First, students often took breaks between the levels, sometime breaks of several days, making the continuity of the play over the full game questionable. Second, the MDP model over six game levels required an extremely large state space, which made the calculation of the Q function and thus the estimation of parameters very slow. Instead we decided to model each game level with a separate MDP, but combine the models when making inference (see Section 4.5). This produced a more reasonable state space while losing only the information about a student's choice to switch game levels, choices that were unlikely to be relevant to our intended inference.

The state space S must capture the meaningful differences in the state of the game as students play. For Microbes, the game state primarily consists of two parts, the environment through which the microbe must navigate and the current configuration of the microbe. Given our assessment goal, the relevant variables for the environment are how much food and how much sun are available, while the relevant variables for the microbe are the number of mitochondria and number of chloroplasts contained within the microbe. During the play of the game these variables interact to determine the amount of energy available to the microbe and thus heavily influence its chance of survival. Other variables that were available in the game records but deemed not relevant to the inferences we wished to make included the amount of predators and barriers in the environment and the type of locomotion organelle with which the microbe was equipped.

As the availability of food and sun are constant for each game level and we are modeling each level with a separate MDP, we can consider food and sun to be fixed within the models, determined by game level as shown in Table 4.1. The amounts are coded as discrete integers ranging from 0, no food or no light, to 3 for abundant food or abundant light. Note that levels $1 - 4$ are tutorial levels and are not modeled as part of the MDP for assessment. Because food, F_l, and sun, S_l, are fixed within each level's model, they are not actually variables within the state space.

The number of mitochondria, m_s, and number of chloroplasts, c_s, within the microbe are state variables as they can change due to student interaction with a game level. As optimal numbers of mitochondria and chloroplasts were around 4 or 5 of each, we model these variables as integers ranging from 0 to 10. To give appropriate rewards, we also need to track when the student was successful in playing a level and whether they have already beaten this level. Thus we include two additional state variables, *just-won*, a boolean that is 1 if the previous action resulted in a win and 0 otherwise, and *won-level*, a boolean that

Table 4.1: Availability of food and sun in the microbe's environment for each modeled game level.

Level (l)	Food (F_l)	Sun (S_l)
5	2	0
6	0	3
7	0	2
8	1	1
9	1	0
10	0	1

is 1 if the student has ever won this level. Overall we have four tracked state variables and $11 \times 11 \times 2 \times 2 = 484$ game states per level.

For the MDP action set we again focus on content relevant actions, defining $A = \{$buy-mito, buy-chloro, play-level, stop$\}$. The first two actions directly affect the configuration of the microbe, incrementing the respective counts. The *play-level* action occurs when the student decides to play the level with the current microbe configuration. This action either does not change the state (when the student fails to reach the goal) or results in a state in which *just-won* and *won-level* are set to 1, based partly on the configuration of the microbe and partly on the student's ability to play the maze-navigating part of the game. Finally the *stop* action occurs when the student decides to stop playing the game level. This often occurs after they have succeeded in winning the level, but may occur after any other action.

The reward function R is set up to reflect the values that students might give to particular action-result pairs. While the game has an internal monetary system which might be considered as a natural reward structure, the "game tokens" are of no use other than for buying microbe features and thus the students probably don't value them intrinsically. Instead we base our reward structure on the estimated psychological value of each transition. We assume that students value making progress in the game, which involves winning the game level they are working on. They may wish to play that level again, even after having won, but we assume that a second win is less rewarding than the initial one. Thus we assign two parameters for the reward of winning a level: $R_{highWin}$ when the student wins a level for the first time, and R_{lowWin} for a repeated win. For non-winning actions, we parameterize the reward (cost) for a buy action as R_{buy}, and for playing without winning as R_{lose}.

The transition function, $p(s_{t+1}|a_t, s_t)$, for purchase actions is deterministic. If a student buys a mitochondrion they will transition to the state in which their microbe configuration includes one more mitochondrion, up to the limit of the state space. If they try to buy a feature that would exceed the limits of the state space, the action has no effect. The play-level action, on the other hand, is probabilistic, resulting either in success or failure based in part on how well the microbe configuration is suited to the environment.

The probability of success is a function of the interactions between the environment (food and sun) and the microbe (mitochondria and chloroplasts). We can thus express the probability of successful play for a level as a logistic model based on the game variables for environment and microbe configuration,

$$\text{logit}(p(success|play, s, l)) = \alpha_0 + \alpha_1 F_l + \alpha_2 F_l m_s + \alpha_3 S_l + \alpha_4 S_l c_s + \alpha_5 S_l r_s \tag{4.4}$$

where F_l is the available food, S_l is the available sunlight, m_s is the current number of mitochondria and c_s is the current number of chloroplasts. The final factor, r_s, is intended to capture the balance between mitochondria and chloroplasts, as a balanced microbe is more efficient than an unbalanced microbe in the presence of sunlight. Thus we define

$$r_s = 1 - \frac{|c_s - m_s + 0.0001|}{(c_s + m_s + 0.0001)} \tag{4.5}$$

with 0.0001 added to prevent discontinuity when both c_s and m_s are zero. Note that r_s will be approximately zero when the microbe has one type of organelle but not the other and it will be approximately 1 when $c_s = m_s$.

The coefficients in Equation 4.4 express the utility of each factor for microbe survival. α_0 is the easiness of the level with no food or sun. α_1 represents the utility of food in the absence of mitochondria and α_3 represents the utility of sunlight in the absence of chloroplasts. α_2 represents the increased utility of food for each additional mitochondria, and similarly α_4 represents the increased utility of sunlight for each additional chloroplast. Finally α_5 represents the increased utility of sunlight when the microbe contains a balanced number of mitochondria and chloroplasts.

Once the elements of the MDPs are defined, the Q functions can be calculated. The student response probability is then modeled using the MDP models for each game level within the action probability equation defined in Equation 4.3.

4.5 Estimation

Estimation proceeds in two phases. First the model parameters are estimated and then individual student abilities are predicted, given the estimated model parameters. The observed data for student j on game level l consist of a sequence of state-action pairs,

$$O_{jl} = \{(s_{1jl}, a_{1jl}), (s_{2jl}, a_{2jl}), \ldots (s_{T_{jl}jl}, a_{T_{jl}jl})\}. \tag{4.6}$$

where T_{jl} is the total number of actions taken by the student on level l. Each pair indicates the game state for that interval and the action taken in that state. The full record for student j includes $\{O_{j1}, O_{j2}, \ldots O_{jL}\}$ where L is the total number of possible game levels. If a student did not play a particular level l, then O_{jl} would be the empty set.

The Markov property applies to this model, allowing us to take not only each level as conditionally independent, but also each action within a level to be conditionally independent, conditioned upon student capability and the system state in which the action was taken. Thus the probability of the observed data can be written as

$$
\begin{aligned}
p(O_j|\beta_j) = & \prod_{l=1}^{L}\prod_{t=1}^{T_{jl}} p(a_{tjl}|s_{tjl},\beta_j) = \prod_{l=1}^{L}\prod_{t=1}^{T_{jl}} \frac{\exp\left(Q_l(s_{tjl},a_{tjl}|\beta_j)\beta_j\right)}{\sum_{a'\in A}\exp\left(Q_l(s_{tjl},a'|\beta_j)\beta_j\right)}. \\
& \beta_j \sim \ln N(\mu,\sigma^2)
\end{aligned}
\tag{4.7}
$$

We estimate the model by taking the β_j parameters as random effects with a parametric log-normal distribution, $\ln N(\mu,\sigma^2)$. A variable is log-normally distributed when a log transformation of the variable would be normally distributed, thus we can alternatively define $\beta_j = \exp(\lambda_j)$ with $\lambda_j \sim N(\mu,\sigma^2)$. The use of the log-normal distribution is discussed in more detail in Chapter 3.

We define the set of all model parameters to include the capability distribution parameters, μ and σ, along with the Q function parameters, R and T, where R is the set of parameters needed to define the reward structure and T is the set of parameters needed to define the transition function. To estimate the model parameters, we use marginal maximum likelihood (MML), marginalizing over the β_j distribution. The marginal likelihood for the model parameters is

$$
L(\mu,\sigma,R,T) = \int_{\beta_j}\prod_{j}^{N} p(O_j|\beta_j;R,T)G(\beta_j;\mu,\sigma)d\beta_j, \tag{4.8}
$$

where $G(\beta_j;\mu,\sigma)$ is the log-normal probability distribution. This likelihood cannot be evaluated analytically. Not only is the integral intractable, but the Q-function must be calculated through iterative approximation. To evaluate the integral, Gaussian quadrature is used for integration over $\lambda_j = \ln(\beta_j)$ with nine quadrature points. The Q_l functions are calculated using policy iteration methods (Howard, 1960) in which the state values and action probabilities are iteratively updated until they are changing less than a specified convergence criterion.

The maximization could then be performed either by an iterative optimization algorithm over the parameter space, or through an MCMC approach if prior distributions are specified for all model parameters. Numerical approximation is used for this study. The maximization over the parameter space is implemented using the nlopt C++ library in two stages, with a "global" search conducted over a large range followed by a localized optimization using the global results as starting values.

Once the model parameters have been estimated, individual student capabilities are predicted. They can be predicted based on a straight-forward maximal likelihood estimation (MLE) or using an empirical Bayesian approach in which the estimated β_j distribution is used as a prior to determine the posterior distribution for β_j. As was noted in the discussion of Chapter 3, the Bayesian estimation using a log-normal distribution for β_j has shown a tendency to push estimates into the lowest part of the distribution, especially when σ is estimated to be large. For this application study, therefore, we will predict β_j using both the MLE and MAP and explore the effects of the differing methods on the resulting estimation.

The MDP modeling software used in Chapter 3 was customized to model the Microbes game. The customization included writing code for the transition functions and the reward functions. The state space and action sets were defined using the standard modules included in the MDP builder.

4.6 Application

Data

The data come from a pilot study in which the Microbes game was evaluated as an assessment instrument. In the study, middle school students played the game, either in class, in an after-school program or at home. Students were then given an in-class multiple-choice post-test which covered appropriate topics in cellular biology. The game play by the students was largely unsupervised, and so while some students completed all of the ten game levels, many students played only a few levels. We are not modeling the first four game levels, as they function more as game-interface tutorials, so we are only interested in the 238 students who completed at least one of the levels from five to ten. Of these students, 148 also completed the post-test.

Game play data

The game play data consisted of a sequence of action steps where actions included buying mitochondria, buying chloroplasts and playing the level. At each action step, the pre-action state of the game was recorded in the form of the game level, the number of mitochondria and the number of chloroplasts in the microbe. For play actions, the data also indicate the results of the play as either 'success' or 'failed.' When the student moved to a new level or stopped playing the game a 'STOP' action was inserted to indicate their decision to stop playing that game level. Excluding the STOP actions, the initial data file recorded 5970 student actions. A sample game record for a single student is shown in Table 4.2. This student played levels 5 and 6, buying two mitochondria on level 5 before choosing to play the level. The first play attempt failed, but the student succeeded the second time. Then the student moved on to level 6, bought a chloroplast and successfully played the level before stopping.

In the actual student data complications arise. Some students bought excessive amounts of organelles, well beyond the ten mitochondria and chloroplasts that were considered reasonable to model. One student, for example, managed to produce a microbe with 50 mitochondria and 157 chloroplasts. To keep our model within reasonable bounds, any actions taken in a game state that exceeded the maximum numbers of mitochondria and chloroplasts were removed from the game record. This truncation removed 710 actions and affected 28 of the students.

A less straight-forward issue with the data arose from students who continued to play a level after they had already succeeded in winning that level. It is unclear whether the

Table 4.2: A sample game record.

Level	Num Mito	Num Chloro	Action	Result
5	0	0	Buy Mito	-
5	1	0	Buy Mito	-
5	2	0	Play	Failed
5	2	0	Play	Success
5	2	0	STOP	-
6	2	0	Buy Chloro	-
6	2	1	Play Level	Success
6	2	1	STOP	-

actions students take after winning a level are useful in a capability measurement context. Up to the point of their first win, it is reasonable to assume that students are motivated by trying to win the game level and thus can be reasonably modeled by the Markov decision process described previously. After they have beaten the level, we expect them to move on to the next level or quit the game. When they continue interacting with the completed level, however, it is unclear what they are trying to accomplish. These may be students who wish to beat the level faster than their previous win, or they may just be exploring the consequences of different paths. As an MDP is a goal oriented model of behavior, we cannot expect to fit data for which the goals are completely unknown. Thus, inferences about student capability in cell biology may be more valid if we trim the data after the first win for each level, eliminating these ambiguous actions from the play records. To evaluate the utility of this post-win data, we will fit our models to two versions of the data: one in which post-win actions are included, and one in which they are removed.

Excluding stop actions, the minimum record length was 1 action while the maximum record length was 199 actions for both the full and trimmed data. The 199 action record was something of an outlier however, with the next longest record being 80 actions for the full data and 60 actions for the trimmed data. The median record length for the full data was 21.5 actions while for the trimmed data it was 19 actions. Students played between one and six levels with 128 students playing all six levels and 64 students playing three or fewer levels.

Game outcome data

To compare our MDP capability estimates to more traditional measures, we also processed the raw data into a more standard, outcome-oriented format. For this data we retain only the results of the student's *play-level* actions during the game. Each level can then be considered as an 'item' with the item score based on the student's win record on that level. As students are allowed to play any level multiple times, however, there is not a one-to-one correspondence between levels and play results. For example, the sample game record shown in Table 4.2 would result in the outcome record shown on the left side of Table 4.3. The student played level 5 twice, failing on their first try but succeeding on their second try, and played level 6 only once, succeeding on that try.

Table 4.3: The sample record from Table 4.2 processed for outcome statistics only.

			Scoring	
Level	Try Num	Success	First Try	Partial Credit
5	1	0		
5	2	1	0	2
6	1	1	1	3

Some students were very persistent, trying a level as many as 28 times before either winning or quiting. To deal with these repeated attempt data, we scored the levels in two different ways. First, using dichotomous scoring, we recorded only the outcome for a student's first try on each level. By this scoring, the sample student would get a 0 on level 5 and 1 on level 6, as shown on the right side of Table 4.3. First-try scoring lumps all students who failed their first attempt together, even though some succeeded on their second attempt and others took 20 tries to succeed. As an alternative, we also used partial credit scoring in which students receive scores from 0 to 3 for each level, depending upon how many tries they required to succeed. Success on the first try was awarded a score of 3, on the second try a score of 2, and on the third try a score of 1. Students who could not win by their third try were given a score of 0 for that level. Using the partial credit scoring, the sample record would be scored 2 for level 5 and 3 for level 6 as shown in the last column of Table 4.3 .

Post-test data

Our final data set comes from the multiple-choice post-test used as part of the pilot study. There were 24 items which dealt with topics such as the basic needs of microbes and the functionality of special organelles such as mitochondria and chloroplasts. As the game is particularly focused on the functionality of mitochondria and chloroplasts, we note that 10 of the items were specifically targeted at those organelles. The items were all scored dichotomously.

Not all students completed all questions on the post-test, with 20 students out of 148 skipping one or more question. One student skipped ten questions, all others completed at least 19 of the 24 questions. All skipped questions were coded as missing data. The post-test scores ranged from 1 to 24 with a mean of 13.0 correct answers.

Models

MDP

The game play records were modeled with the Markov decision process measurement models detailed in sections 4.3 and 4.4. Each game level was modeled with an MDP for that specific level but with shared parameters across all levels. This approach allows the state space for each MDP to be kept small while utilizing the information from the complete game records for parameter estimation.

In the Microbe MDP, the transition parameters represent a cognitive model of the interaction between the microbe's organelles and its environment. If we were to estimate the transition parameters based on the student action data, we would be estimating the students' beliefs about how the system worked (Rafferty et al., in press), as it is their internal representations of the interaction between organelles and environment that determine their actions. Such a diagnostic model could have uses, particularly in a formative assessment, but such is not the goal here. Instead, to produce a capability-oriented measurement model, we fix the transition parameters to represent the correct belief and use the estimation of β_j as a measure of how closely their actions match optimal actions given that correct belief. The question then arises, how do we determine the correct values for the transition parameters? Two different approaches were compared for making that determination. First a set of values were assigned based on theoretical considerations of how the microbe's configuration and the environmental factors should affect the probability of success. The theoretical values were set as shown in Table 4.4. These values reflect that food can be used by the microbe directly, but is more efficiently converted to energy in the presence of mitochondria, while sun is not useful to the microbe without chloroplasts and is increasingly useful when chloroplasts are matched with mitochondria. The last coefficient, α_5 is set to 2 rather than 1 because as a ratio, r_s ranges between 0 and 1, while the other factors are integer value ranging up to 3 for food and sun and up to 10 for mitochondria and chloroplasts.

Table 4.4: Theoretical and empirical transition parameter values for the MDP measurement model. Standard errors are in parentheses.

Parameter	Function	Theoretical	Empirical
α_0	intercept	-5	-2.70 (0.14)
α_1	coefficient for food alone	1	0.67 (0.09)
α_2	interaction between food and mitochondria	1	0.12 (0.02)
α_3	coefficient for sun alone	0	0*
α_4	interaction between sun and chloroplasts	1	0.23 (0.03)
α_5	interaction between sun and the ratio of mitochondria to chloroplasts	2	0.97 (0.14)

*Constrained parameter.

The theoretical values for the transition parameters are fairly imprecise, expressing only the positive value of the different combinations of microbial and environmental factors without resolving the relative value between the different factors. One way to achieve a closer approximation of those coefficients is to estimate the empirical value of each factor based on the outcome data. As many students have played the game with different configurations of microbes in different environments (different levels) we have enough data to determine the effect of each factor on the probability of winning the game level. Note that this is different from estimating the transition parameters as part of the MDP model, in which case the estimation would be based on when the students choose the play action rather than when they were successful in their play. To empirically determine the effect of the organelles and environmental factors on the probability of successful play, we modeled the play outcomes using a hierarchical logistic regression with plays nested in students.

The regression model is the same as Equation 4.4 excepting that we add a random intercept, ζ_j at the person level:

$$\begin{aligned} \text{logit}\{p(Y_{tj}=1|x_{tj},\zeta_j)\} =& \alpha_0 + \alpha_1 F_{l_t} + \alpha_2 F_{l_t} m_{s_t} + \alpha_3 S_{l_t} + \alpha_4 S_{l_t} c_{s_t} + \alpha_5 S_{l_t} r_{s_t} + \zeta_j \\ & \zeta_j \sim N(0,\sigma') \end{aligned} \tag{4.9}$$

where Y_{tj} is 1 if student j won their tth play of the game, and $x_{tj} = \{F_{l_t}, S_{l_t}, m_{s_t}, c_{s_t}, r_{s_t}\}$ where l_t is the game level being attempted, and s_t is the game state in which student j played the game, at try t. The different game levels are deterministically described by the explanatory variables F_{l_t} and S_{l_t}, so game level does not separately enter into the model. While the student results do provide empirical evidence of which game states produced a higher probability of success, it should be noted that the selection of states in which attempts were made was far from random. As students would favor particular game states for level-play, the data suffer from selection bias and therefore the estimates may exhibit some bias. The model was estimated using `lmer` in `R` using adaptive Gauss-Hermite quadrature with 15 nodes and produced the coefficient in the 'Empirical' column of Table 4.4. The full regression results can be found in Appendix B.

In addition to the transition parameters, the reward parameters must also be specified or estimated. For the model to be identified, the reward structure must include two fixed parameters which constrain the location and scale of the Q function (see Chapter 3). We constrain the reward for winning, $R_{win} = 1$ and the reward for stopping, $R_{stop} = 0$ to serve this function. The only remaining reward parameters of interest are R_{buy}, the reward (or cost) of buying organelles and R_{lose} the cost of losing a play attempt. We can estimate R_{buy} and R_{lose} at the population level as an estimate the subjective cost that students generally assigned to time spent configuring their microbe and to losing their play attempt. To maintain meaning of the parameters, however, we restrict the range of $R_{buy} \in (-R_{win}, 0)$, as values of R_{buy} outside of this range would fundamentally change the cognitive model represented by the MDP. Similar to the transition function, we will fit a model in which all of the reward parameters are constrained to their theoretical values, and one in which the R_{buy} and R_{lose} are estimated as model parameters.

IRT

For the comparison IRT model we used a Rasch model, estimated with the item parameters constrained to sum to zero. Student abilities were predicted using both maximum likelihood (MLE) and expected a-posteriori (EAP) estimation methods. We fit the 'first try' outcome data using a dichotomous model (IRT FT) and the partial credit outcome data using a partial credit model (IRT PC). The post-test data were also fit with a dichotomous Rasch model. All IRT models were run using ACER's ConQuest software (Wu, Adams, & Wilson, 1998).

Analysis

Four different MDP models were compared (Table 4.5), varying whether the transition parameters are set to theoretical or empirical values and whether the reward parameters are fully constrained or partially estimated. The model fits were compared based on their log-likelihood and AIC fit statistics. The student capability parameters were then estimated using both the MLE and MAP estimation methods. For validity evidence, we examined the pair-wise correlation of these capability estimates with estimates from the students' post-test results. The MDP estimates were also compared to those generated by the outcome-only IRT models to determine if the models utilizing the game action data provide any additional information beyond that available in the win/lose data. These analyses were performed twice, once on the full data and once on the trimmed data in which actions after the first win per level have been removed.

Table 4.5: MDP models to be compared.

	transition	estimated
model	parameters	model parameters
1	theoretical	μ, σ
2	empirical	μ, σ
3	theoretical	$\mu, \sigma, R_{buy}, R_{lose}$
4	empirical	$\mu, \sigma, R_{buy}, R_{lose}$

Results

For all models, the estimates from the full and trimmed data were similar, but the person estimates from the trimmed data correlated better with the post-test results. Thus the results from the trimmed data will be reported here. Similarly, while the person capability parameters, β_j were estimated using both the MAP and MLE methods, we found that the MAP estimates were very sensitive to the estimated capability distribution parameters. In particular, high values of $\hat{\sigma}$ tended to force the capability estimates close to zero, because for log-normal distributions, the σ parameter is a shape parameter and high values create a distribution that is highly peaked at the low end of the range. These estimates did not correlate as well with the post-test capability estimates as did the MLE estimates. For the remainder of this section, we will use the MLE estimates.

We first look at model fit, comparing the theoretical versus empirical transition parameters and the completely fixed versus estimated reward parameters. Table 4.6 shows that model 4, with empirical transition parameters and estimated reward parameters, produced the best fit by all criteria. The estimated distribution of β_j varied widely, with μ estimated near 0 by model 1 and around 1.4 for the other three models. The σ parameter was generally estimated to be large, with 1.755 being the lowest estimate, given by model 3.

Table 4.6: Comparison of MDP models running on the trimmed data.

	model 1	model 2	model 3	model 4
Parameter Estimates				
μ	0.051	1.445	1.410	1.467
σ	2.503	1.933	1.755	3.024
R_{lose}	-0.20*	-0.20*	-0.006	-0.067
R_{buy}	-0.10*	-0.10*	-0.969	-0.675
Model Fit				
-2 log(L)	15461.87	14887.86	13444.96	12694.45
AIC	15465.87	14891.86	13452.96	12702.45

*Fixed parameters.

For the models which estimated the reward parameters, we note that R_{lose} was estimated to be close to zero while R_{buy} was nearly the negative of R_{win}, the reward for winning. Given such estimates, the MDP would predict that in almost any game state, students would choose to play the level rather than buy an organelle, as there would be little cost for playing and losing compared to the relatively large cost for upgrading the microbe. The fact that the model produced these estimates suggests that a significant proportion of the sample students are indeed choosing to play too often and to buy not often enough. The reward estimates, while within the range for what is rational, change the meaning of the MDP as a measurement model as they change which actions are optimal. If play is always the optimal action, then students who bought organelles would be considered to be showing poor judgment and thus receive lower ability estimates. Thus for models 3 and 4, β_j may no longer serve as a measurement of understanding cell biology.

Comparing the capability estimates given by the four MDP models, we can see from Figure 4.2 that models 3 and 4 appear to be behaving quite differently from models 1 and 2. Many of the students who were given very low estimates in the fixed reward models (1 and 2) are given high estimates in the corresponding estimated reward models (3 and 4). Other students seem to be distributed almost randomly by the estimated reward models. This fits with our interpretation of how the estimated values of R_{lose} and R_{buy} would alter the meaning of the MDP. While we would like to use the estimates of R_{buy} to measure the

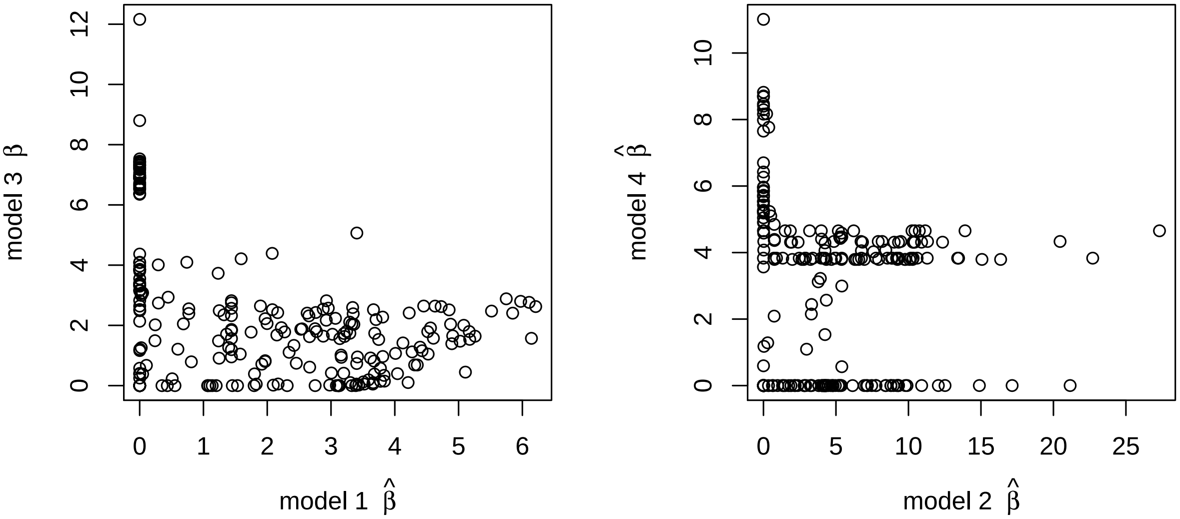


Figure 4.2: Comparison of fixed reward models (x-axes) and estimated reward models (y-axes) when estimated over the full sample. Correlations were -.43 for models 1 and 3 and -.14 for models 2 and 4. Note that 6 extreme values in the second graph are not shown for clarity of the remaining estimates.

general motivation of the students while also measuring their content understanding with β_j, this methodology cannot be used if the estimated R values fundamentally change the measurement construct.

Interestingly, for the fixed reward models, the distribution of the $log(\hat{\beta}_j)$ values appear bi-modal (Figure 4.3), though we would expect $log(\beta_j)$ to be normally distributed. Thus it seems probable that our population consists of two distinct subpopulations. Examination of the play records for the students whose estimates were in the lower clusters revealed that a large proportion of these students never bought any mitochondria or chloroplasts.

Of the original 238 students, 50 of them never bought a single organelle. These were not students who merely quit early. The median number of levels played in this group was 5 out of the 6 possible, while the median number of play attempts was 17.5. Not surprisingly, these students made up a large proportion of those who received very low capability estimates in the fixed reward models. For model 1, for example, of the 71 students whose $log(\hat{\beta}_j)$ was less than -5, 46 were no-buy students. While some of these no-buy students are likely to have been truly confused about the value of the organelles, it seems odd that so many students would not have understood the simplest part of the content, that a microbe's organelles help it to survive. Indeed it seems plausible that instead many of them were playing based on a cognitive model that did not match our hypothesized model. It is possible, for example, that

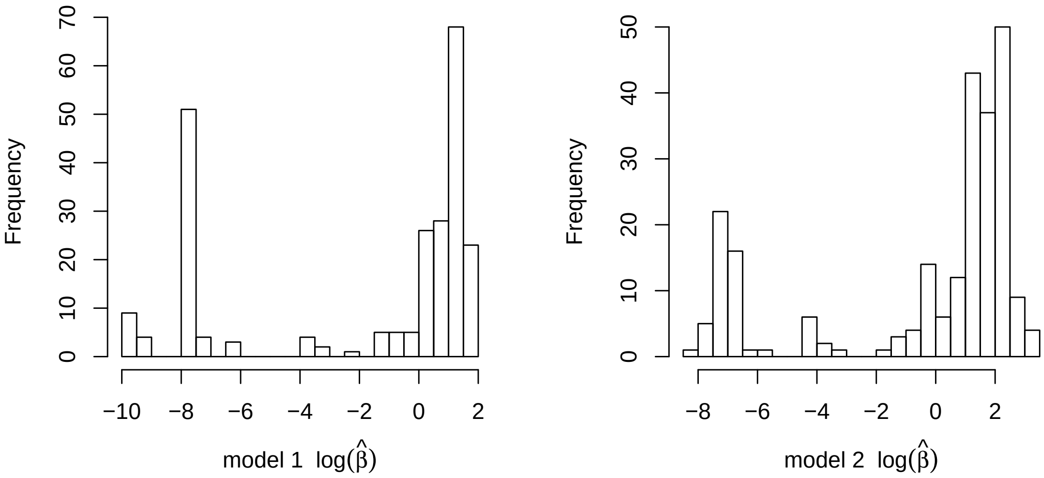


Figure 4.3: Distribution of $\log(\hat{\beta}_j)$ for the fixed reward models. We expect $\log(\beta_j)$ to be normally distributed but find it to be bimodal.

some of these students did not understand the game interface for purchasing organelles, and therefore never managed to buy any. This could explain the usual estimates for the reward parameters when we estimated them over the full population.

To test whether the estimation of the R parameters would be feasible in the absence of these possibly off-model students, models 3 and 4 were estimated again using only those students who had at least one organelle purchase. Those parameters estimates were then used to predict the individual β_j parameters for the full sample, including the no-buy students. These results are presented in Table 4.7 and referred to as models $3'$ and $4'$ to distinguish them from the original model estimates. Again we find the estimated penalty for losing to be much smaller than in our theoretical value, while the estimated cost for buying is larger than the fixed value. The buy cost this time is not prohibitive, however, and buying organelles remains a probable option in many game states. The predicted β_j values from these models correlate reasonably with models 1 and 2 with the results of model $3'$ correlating with model 1 as $\rho = .99$ and the results of model $4'$ correlating with model 2 as $\rho = .72$.

Interestingly, when we compare the person estimates for the models to the post-test capability estimates, we find the highest correlations are not given by the models with the best overall fit. Table 4.8 shows that model $3'$, using theoretical transition parameter values and estimated reward parameters, yields the highest correlations with the post-test capability estimates, $\rho = .52$. The IRT models correlate less well than any of the candidate MDP models with the best correlating model having $\rho = .38$, suggesting that the process data does in fact provide relevant information beyond what can be gained from the outcome data on which the IRT models rely. The correlation between model $3'$ and IRT PC was .76,

Table 4.7: Models 3 and 4 estimated only on records that included at least one buy action.

	model $3'$	model $4'$
Parameter Estimates		
μ	1.037	1.500
σ	1.041	4.844
R_{lose}	-0.035	-0.028
R_{buy}	-0.203	-0.328
Model Fit		
-2 log(L)	11235.18	10842.70
AIC	11243.18	10850.70

even higher than either's correlation to the post-test. This is not very surprising, as we would expect there to be a strong connection between the actions taken in the game and the final outcome. While the Pearson correlation statistics are encouraging, it should be noted that any inferences made from Pearson correlations assume two normal distributions, and that assumption is clearly violated here. Thus for comparison, the Spearman correlations, which do not assume normality, are also provided. The model rankings by Spearman correlation are the same as with the Pearson correlations, with model $3'$ doing the best overall and the partial credit model correlating highest of the two IRT models.

Table 4.8: Correlations between capability estimates from MDP models, IRT models and the post-test IRT model.

	Correlation with Post-Test	
	Pearson	Spearman
MDP mod 1	.507	.474
MDP mod 2	.439	.445
MDP mod $3'$	.516	.492
MDP mod $4'$	.405	.450
IRT FT	.317	.311
IRT PC	.379	.388

Correlations can be deceptive, however, so we examined the relationship between the highest correlating MDP model, the highest correlating IRT model and the post-test estimates in Figure 4.4. There are plausible linear trends in all three graphs, confirming the interpretation of the correlation statistics. Of the students who performed poorly on the MDP assessment ($\hat{\beta}_j < 0.1$), 66% of them also performed poorly on the post-test ($\hat{\beta}_j < 0$). Of the 12 low performing MDP students who achieved an above zero estimate on the post-test, 8 were students who had not bought any organelles. This might be considered further evidence that the game performance of some students was more reflective of problems with the technology rather than understanding of the content.

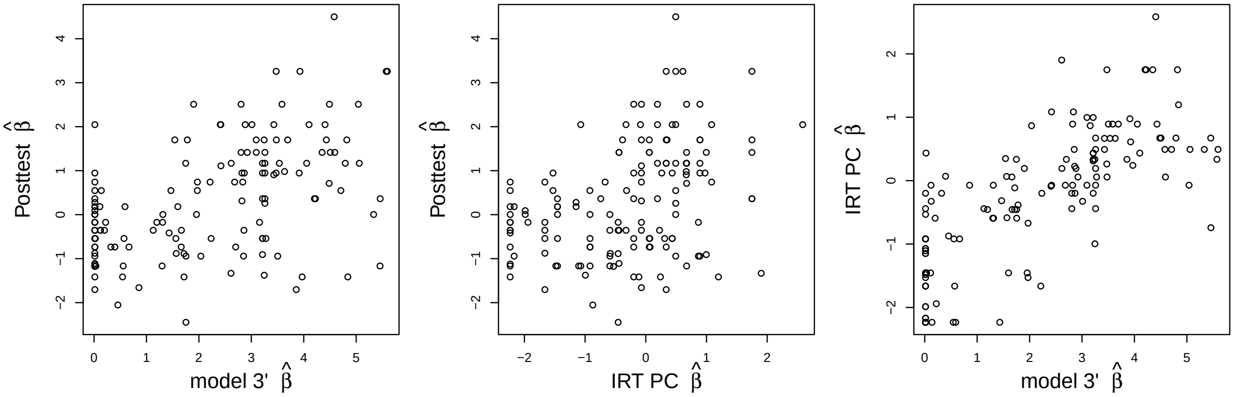


Figure 4.4: The relationships between capability estimates based on the MDP model 3′, the IRT PC model and the post-test IRT model.

4.7 Discussion

The application of the MDP measurement model to the Microbe game data was generally successful. The positive correlations with post-test capability estimates provide reasonable validity evidence that the model is measuring a construct similar to the construct measured in the post-test. Further, the fact that the MDP capability estimates correlated more highly than the best IRT model's estimates suggests that the MDP model, and the process data on which it relies, can yield more information about student competency than outcome data alone.

It should be noted, however, that with the highest correlation at .52, all of these model estimates correlate less strongly than we would expect from an alternate form assessment. While the game data does contain information about student capability, it does not appear to provide the same amount of information that a dedicated assessment would. This is not surprising, as the game in question was not designed to be an assessment at all. As Mislevy and colleagues point out (2012), while games can share similar design goals to assessments, they do not necessarily do so. To act as good assessments, games need to be designed with assessment principles in mind.

An interesting issue that this study has uncovered is the relationship between model fit and measurement utility when using a cognitive model for measurement. The MDP models that used the empirical transition parameters proved to be a much better fit for the data, implying that those models were closer to the cognitive model students used while playing the game. On the other hand, those models did not correlate as well with the post-test results. This seeming contradiction may be caused by the differing purposes of cognitive modeling and measurement. Educational measurement involves comparing a performance against a standard, much like a yardstick. In traditional psychometrics, the instrument is calibrated by estimating assessment-specific model parameters. While these estimates rely upon student data, they are assumed to be estimates of true parameters that do not. For the MDP as a cognitive model, on the other hand, almost all of the parameters can be considered person-specific or at least population specific. In this study we identify only one of those parameters as the latent variable which we are measuring. If we then estimate the non-measurement parameters from the data, we may be doing something akin to customizing the yardstick to the population before measuring with it. Thus it is not surprising that the theoretical values would produce a better measurement as they are, in effect, acting as a standard, ensuring that the measurement still means what we intend it to mean.

If on the other hand, the conceptions reflected in the transition or reward parameters are of interest, a latent class model could be used to diagnose the student conceptions (Rafferty et al., in press). A direction of future research would be to combine the latent class model and the continuous measurement model as a mixture model, similar to the DSDMM described in Chapter 2. Such a model might, for example, be able to distinguish the students who did not seem to understand the organelle purchasing interface while at the same time providing valid capability measures for those that did. This would be analogous to our model $3'$ in which we estimated the model parameters based on a subset of the population, though in the present study we identified the subpopulations by hand.

One of the goals of this study was to examine the feasibility of applying the MDP measurement model to real life data. While the MDP approach holds the promise of a generalized model that can be fit to many different complex tasks, the current state of the art requires considerable customization to the task of interest. The first challenge is specifying the MDP model for the task. It was not always clear which variables to include in the state space, which actions were important and how to best parameterize the transition function and the reward structure. Using the MDP as a generating model can be a great help in this phase of development to determine if the model is producing the behavior that would be expected. Even with the MDP properly specified, custom C++ code was required to implement the transition and reward modules. If these models are to be used by non-specialists, future development should include both a wider range of out-of-the-box MDP solutions and tools to aid in the formulation and testing of new type of MDP models.

Chapter 5

Concluding Thoughts

One of the reoccurring themes of this work has been the interplay between student conception and student ability. While high ability students, holding the accepted correct concept of the domain, are easy to recognize, what does it mean to be "low ability?" Is ignorance the same as misconception? I would argue that complete ignorance, what I refer to as "concept-free" in Chapter 2, results in choosing responses uniformly at random. These are students for whom the terms in the prompt may have little meaning. If, for example, a kindergardener was presented with a multiple-choice calculus test, the best he could do is to randomly choose responses. Thus when we observe students performing worse than random chance, they are demonstrating something beyond ignorance; they are demonstrating an incorrect conception. This could be a misconception about the content knowledge, about the goal of the task, about the procedure needed to solve the problem, or about the problem space in general.

Chapter 2 explicitly addressed this issue by defining a concept space and classifying students within that space by task conception. In Chapters 3 and 4, however, the same issue lurked under the surface. The Markov decision process model assumes that students understand the system dynamics as specified in the transition model and have goals that are consistent with the defined reward structure. It also assumes that students know what is possible, both in terms of their available actions and in terms of the scope of the state space. From this stance, we can identify students who hold the correct conception and are good at optimizing their actions, but we cannot distinguish a poor planner from a confused student. The obvious extension is to create an MDP mixture model, one in which we can classify students according to their beliefs, while also measuring their strategic thinking within their belief system. This combines the conceptual framework of the DSDMM with the context-sensitive decision modeling of the MDP. It is a direction in which I hope to extend my work.

In the case of formative assessment, diagnosing misconceptions may be more valuable than labeling students as "low ability," as instruction can be customized based on the diagnoses. Even in the case of summative or high-stakes assessments, however, the distinction is important. If a misconception about a non-construct-relevant element of the task is confounding measurement of the construct of interest, it would be better to know that we cannot estimate this student's ability, rather than to believe we have accurately estimated him as low-ability. Traditional assessments have attempted to avoid this problem by making each item clear and simple enough that construct irrelevant variance was minimized. With complex tasks, we can no longer avoid such complications.

As the NGSS prods psychometricians to look more seriously at the implications of using complex tasks, I hope that these thoughts will become part of a wider discussion about the cognitive models needed to produce efficient, reliable and valid estimates about relevant student traits, based on the data available from these tasks. Such a discussion will need to include cognitive scientists, data scientists, and learning specialists as well as psychometricians and science educators. I look forward to being part of the conversation.

References

Baker, C., Saxe, R., & Tenenbaum, J. (2009). Action understanding as inverse planning. *Cognition*, *113*(3), 329–349.

Baker, C., Saxe, R., & Tenenbaum, J. (2011). Bayesian theory of mind: Modeling joint belief-desire attribution. In *Proceedings of the thirty-third annual conference of the cognitive science society* (p. 2469–2474).

Behrens, J. T., Mislevy, R. J., DiCerbo, K. E., & Levy, R. (2012). Evidence centered design for learning and assessment in the digital world. In M. C. Mayrath, J. Clark-Midura, D. H. Robinson, & G. Schraw (Eds.), *Technology-based assessments for 21st century skills: Theoretical and practical implications from modern research* (p. 13–53). Charlotte, NC: Information Age Publishing.

Bock, D. R. (1972). Estimating item parameters and latent ability when responses are scored in two or more nominal categories. *Psychometrika*, *37*(1), 29–51.

Bradshaw, L., & Templin, J. (2013). Combining item response theory and diagnostic classification models: A psychometric model for scaling ability and diagnosing misconceptions. *Psychometrika*, 1–23.

Bransford, J. D., Brown, A. L., & Cocking, R. R. (2000). *How people learn.* Washington, DC: National Academy Press.

Brennan, R. L., & Johnson, E. G. (1995). Generalizability of performance assessments. *Educational Measurement: Issues and Practice*, *14*(4), 9–12.

Buckley, B. C., Gobert, J. D., Horwitz, P., & O'Dwyer, L. M. (2010). Looking inside the black box: assessing model-based learning and inquiry in BioLogica$^{\text{TM}}$. *International Journal of Learning Technology*, *5*(2), 166–190.

Corbett, A., Anderson, J., & O'Brien, A. (1995). Student modeling in the ACT programming tutor. In P. D. Nichols, S. F. Chipman, & R. L. Brennan (Eds.), *Cognitively diagnostic assessment* (pp. 19–41). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

DiCerbo, K., & Behrens. (2012). Implications of the digital ocean on current and future assessment. In R. W. Lissitz & H. Jiao (Eds.), *Computers and their impact on state assessments: Recent history and predictions for the future* (p. 273–306). Charlotte, NC: Information Age Publishing.

Draney, K. L., Pirolli, P., & Wilson, M. (1995). A measurement model for a complex cognitive skill. In P. D. Nichols, S. F. Chipman, & R. L. Brennan (Eds.), *Cognitively diagnostic assessment* (p. 103–125). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Fischer, G. H. (1973). The linear logistic test model as an instrument in educational research. *Acta psychologica*, *37*(6), 359–374.

Fu, A. C., Raizen, S. A., & Shavelson, R. J. (2009). The nation's report card: A vision of large-scale science assessment. *Science*, *326*(5960), 1637–1638.

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical science*, 457–472.

Gelman, A., & Shirley, K. (2011). Inference and monitoring convergence. In S. Brooks, A. Gelman, G. Jones, & X.-L. Meng (Eds.), *Handbook of markov chain monte carlo* (pp. 163–175). Boca Raton, FL: Chapman & Hall/CRC.

Howard, R. A. (1960). *Dynamic programming and markov processes* (1st ed.). Cambridge, Mass.: The MIT Press.

Limpert, E., Stahel, W. A., & Abbt, M. (2001). Log-normal distributions across the sciences: Keys and clues. *BioScience*, *51*(5), 341. doi: 10.1641/0006-3568(2001)051[0341:LNDATS]2.0.CO;2

Linn, R. L., Baker, E. L., & Dunbar, S. B. (1991). Complex, performance-based assessment: Expectations and validation criteria. *Educational Researcher*, *20*(8), 15.

Lunn, D. J., Thomas, A., Best, N., & Spiegelhalter, D. (2000). WinBUGS - a bayesian modelling framework: concepts, structure, and extensibility. *Statistics and computing*, *10*(4), 325–337.

McElhaney, K. W., & Linn, M. C. (2011). Investigations of a complex, realistic task: Intentional, unsystematic, and exhaustive experimenters. *Journal of Research in Science Teaching*, *48*(7), 745–770. doi: 10.1002/tea.20423

Mislevy, R. J., Almond, R., Dibello, L., Jenkins, F., Steinberg, L., Yan, D., & Senturk, D. (2002, November). *Modeling conditional probabilities in complex educational assessments.* (CSE Tech. Rep.) Los Angeles, CA: The National Center for Research on Evaluation, Standards, Student Testing, Cen- ter for Studies in Education, University of California, Los Angeles.

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

Mislevy, R. J., Oranje, A., Bauer, M. I., Von Davier, A., Hao, J., Corrigan, S., ... John, M. (2014). *Psychometric considerations in game-based assessment* (White Paper). Glass Lab.

Mislevy, R. J., & Verhelst, N. (1990). Modeling item responses when different subjects employ different solution strategies. *Psychometrika*, *55*(2), 195–215.

National Research Council. (2001). *Knowing what students know: The science and design of education assessment* (J. W. Pellegrino, N. Chudowsky, & R. Glaser, Eds.). Washington, DC: National Academies Press.

National Research Council. (2014). *Developing assessments for the next generation science standards.* Washington, DC: The National Academies Press.

Ng, A. Y., & Russell, S. (2000). Algorithms for inverse reinforcement learning. In *Proceedings* *of the seventeenth international conference on machine learning (2000)* (p. 663–670). doi: 10.1.1.41.7513

NGSS Lead States. (2013). *The next generation science standards: Executive summary.* Achieve, Inc. on behalf of the twenty-six states and partners that collaborated on the NGSS. Retrieved 2014-07-10, from `http://www.nextgenscience.org/next-generation-science-standards`

Pellegrino, J. W., & Quellmalz, E. S. (2010). Perspectives on the integration of technology and assessment. *Journal of Research on Technology in Education*, *43*(2), 119–134.

Puterman, M. L. (1994). *Markov decision processes: Discrete stochastic dynamic programming.* New York: John Wiley & Sons, Inc.

Quellmalz, E. S., Timms, M. J., Buckley, B. C., Davenport, J., Loveland, M., & Silberglitt, M. D. (2012). 21st century dynamic assessment. In M. C. Mayrath, J. Clark-Midura, D. H. Robinson, & G. Schraw (Eds.), *Technology-based assessments for 21st century skills: Theoretical and practical implications from modern research* (pp. 55–90). Charlotte, NC: Information Age Publishing.

Quellmalz, E. S., Timms, M. J., Silberglitt, M. D., & Buckley, B. C. (2012). Science assessments for all: Integrating science simulations into balanced state science assessment systems. *Journal of Research in Science Teaching*, *49*(3), 363–393.

Rafferty, A., LaMar, M., & Griffiths, T. (in press). Inferring learners' knowledge from their actions. *Cognitive Science*.

Resnick, L. B., & Resnick, D. P. (1992). Assessing the thinking curriculum: New tools for educational reform. In B. R. Gifford & M. C. O'Connor (Eds.), *Changing assessments: alternative views of aptitude, achievement, and instruction* (p. 37–75). New York: Springer.

Rost, J. (1990). Rasch models in latent classes: An integration of two approaches to item analysis. *Applied Psychological Measurement*, *14*(3), 271.

Rupp, A. A., Templin, J., & Henson, R. A. (2012). *Diagnostic measurement: Theory, methods, and applications.* New York: Guilford Press.

Samejima, F. (1979). *A new family of models for the multiple-choice item.* (Tech. Rep.). DTIC Document.

San Martín, E., del Pino, G., & De Boeck, P. (2006). IRT models for ability-based guessing. *Applied Psychological Measurement*, *30*(3), 183–203.

San Martín, E., Rolin, J.-M., & Castro, L. M. (2013). Identification of the 1pl model with guessing parameter: parametric and semi-parametric results. *Psychometrika*, *78*(2), 341–379.

Shaffer, D. W., & Gee, J. P. (2012). The right kind of GATE. In M. C. Mayrath, J. Clark-Midura, D. H. Robinson, & G. Schraw (Eds.), *Technology-based assessments for 21st century skills: Theoretical and practical implications from modern research* (pp. 211–228). Charlotte, NC: Information Age Publishing.

Shavelson, R. J., Baxter, G. P., & Gao, X. (1993). Sampling variability of performance assessments. *Journal of Educational Measurement*, *30*(3), 215–232.

Shute, V. (2011). Stealth assessment in computer-based games to support learning. In S. Tobias & J. Fletcher (Eds.), *Computer games and instruction* (pp. 503–523). Charlotte, NC: Information Age Publishers.

Shute, V. J., Masduki, I., Donmez, O., Dennen, V. P., Kim, Y.-J., Jeong, A. C., & Wang, C.-Y. (2010, January). Modeling, assessing, and supporting key competencies within game environments. In D. Ifenthaler, P. Pirnay-Dummer, & N. M. Seel (Eds.), *Computer-based diagnostics and systematic analysis of knowledge* (pp. 281–309). Springer US.

Siegler, R. S. (1976). Three aspects of cognitive development. *Cognitive Psychology*, *8*(4), 481–520.

Skrondal, A., & Rabe-Hesketh, S. (2007). Latent variable modelling: A survey. *Scandinavian Journal of Statistics*, *34*(4), 712–745.

Tatsuoka, K. K. (1985). A probabilistic model for diagnosing misconceptions by the pattern classification approach. *Journal of Educational and Behavioral Statistics*, *10*(1), 55.

Thissen, D., & Steinberg, L. (1986). A taxonomy of item response models. *Psychometrika*, *51*(4), 567–577.

Vendlinksi, T., & Stevens, R. (2002). Assessing student problem-solving skills with complex computer-based tasks. *The Journal of Technology, Learning and Assessment*, *1*(3).

Wilson, M. (1989). Saltus: A psychometric model of discontinuity in cognitive development. *Psychological Bulletin*, *105*(2), 276.

Wilson, M. (2005). *Constructing measures.* New York: Psychology Press.

Wu, M. L., Adams, R. J., & Wilson, M. R. (1998). *ConQuest* [Computer software and manual]. Camberwell, Victoria, Australia: Australian Council for Educational Research.

Appendix A

DSDMM BUGS Code

Full Model

```
model
{
  # Item difficulties are constrained to sum to zero.  We start by
  # drawing all but the last item from a zero mean normal distribution.
  # The variance is a hyper-parameter, but not group specific.
  for(i in 1:(I-1)) {
    for (g in 1:G) {
      delta[i,g] ~ dnorm(0, d_pre)
    }
  }

  # Constrain the item parameters to sum to 0 by setting
  # the last item delta to the negative sum of the others.
  for (g in 1:G) {
    delta[I,g] <- -sum(delta[1:(I-1),g])
  }

  # Group membership has a categorical prior.
  for(s in 1:N) {
    group[s] ~ dcat(pi[1:G])
  }

  # The priors for the mixing proportion parameters (pi) are dirichlet(1,...,1),
  # which is equivalent to gamma(1,1,), normalized.
  for (g in 1:G) {
    pi[g] <- dg[g]/sum(dg[1:G])
    dg[g] ~ dgamma(1,1)
  }

  # The guessing parameters.  Also using dirichlet priors.
  for(i in 1:I) {
        for(r in 1:R) {
        c[i,r] <- ci[i,r]/sum(ci[i,1:R])
        ci[i,r] ~ dgamma(1,1)
        }
  }

  for(s in 1:N) {
    # Student abilities, with normal priors:
```

```
    theta[s] ~ dnorm(th_mu[group[s]], th_pre[group[s]])

    # The model of the data:
    for(i in 1:I) {
      for(r in 1:R) {
         p[s,i,r] <- c[i,r] +
            (equals(r, key[group[s],i]) - c[i,r])* (exp(theta[s] - delta[i,group[s]]) /
            (1+exp(theta[s] - delta[i,group[s]])))

      }
       x[s,i] ~ dcat(p[s,i,1:R])
    }
  }

  # Hyper-priors:
  d_pre ~ dgamma(0.5, 0.5)

  for (g in 1:G) {
    th_mu[g] ~ dnorm(0, 0.01)
    th_pre[g] ~ dgamma(0.2, 0.2)
  }
}
```

Appendix B

Transition Parameter HLM Regression

```
Generalized linear mixed model fit by the adaptive Gaussian Hermite approximation
Formula: success ~ (1 | studID) + food + food_mito + sun_chloro + sun_mcRatio
   Data: playResDat_sub
  AIC  BIC logLik deviance
 2919 2956  -1453     2907
Random effects:
 Groups Name        Variance Std.Dev.
 studID (Intercept) 1.0464   1.023
Number of obs: 3393, groups: studID, 235

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.69638    0.14171 -19.028  < 2e-16 ***
food         0.67497    0.09475   7.124 1.05e-12 ***
food_mito    0.12202    0.01739   7.016 2.28e-12 ***
sun_chloro   0.23440    0.03468   6.758 1.40e-11 ***
sun_mcRatio  0.96557    0.13853   6.970 3.17e-12 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Correlation of Fixed Effects:
            (Intr) food   fod_mt sn_chl
food        -0.591
food_mito   -0.141 -0.439
sun_chloro  -0.320  0.249  0.029
sun_mcRatio -0.283  0.191  0.108 -0.628
```