Regularized Marginal Maximum Likelihood: The Use of Shrinkage and Selection Operators for Item Parameter Estimation in the Two-Parameter Logistic Model

A DISSERTATION SUBMITTED TO THE FACULTY OF UNIVERSITY OF MINNESOTA BY

Chris Hulme-Lowe

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Dr. Niels Waller, Adviser

April, 2016

Acknowledgments

I would like to thank my adviser, Dr. Niels Waller, for his help, support, guidance, and the occasional necessary kick in the rear over the past few years, my wife Danielle, my parents Susan and Alan, and my sister Carolyn for their unwavering support and willingness to listen to me talk about things that may or may not have been intelligible, Dr. Jeff Jones, without whom I would probably never have discovered Quantitative Psychology, Doctors Robert Krueger and Sarah Jahn of Concordia University, St. Paul for fostering my interest in mathematics and statistics, and Dr. Molly Isbell and Carlos Acevedo at Signature Science for their support in the final months of this undertaking.

Dedication

This thesis is dedicated to Danielle Hulme-Lowe, without whose love and support it would never have been completed.

Abstract

Regularized parameter estimation has attracted considerable interest in the statistical and machine learning communities as a powerful estimation method that is viable when classic maximum likelihood estimation is not. In this paper, we describe a method of applying regularized estimation to IRT item parameter estimation. The method proposed herein, Regularized Marginal Maximum Likelihood (RMML) is based on the well known Marginal Maximum Likelihood (MML) method, but penalizes on the discrimination parameter estimates with the aim of eliminating poorly performing items. A series of Monte Carlo simulation studies compare RMML estimates to estimates from both MML and Bayesian estimation under a variety of conditions. The results of these simulations demonstrate that RMML is useful when item parameters must be estimated from a small sample of examinees and provide insight into directions for further research in this area.

Contents

List of Figures

List of Tables

1 Introduction

Item response theory (IRT; Lord & Novick, 1968) is at the heart of modern psychometric methods (de Ayala, 2004). IRT is used by researchers in psychology, education, and other domains to characterize the effects of latent variables on observed behaviors or traits (de Ayala, 2004; Lord & Novick, 1968). It is also widely used to design, refine, and equate high-stakes achievement tests, such as the Scholastic Aptitude Test (SAT) and Graduate Record Exam (GRE), personality tests (Fraley, Waller, & Brennan, 2000; Waller & Reise, 2010), and instruments for measuring psychopathology (Shea, Tennant, & Pallant, 2009). IRT also underpins modern computerized adaptive testing (CAT) procedures (Chang, 2015; Weiss, 1982), which allow researchers to measure their subjects more accurately with fewer questions. Although a majority of IRT's development has taken place in psychology and education, the flexibility of IRT has recently garnered attention from other domains, such as genetics (Houseman, Karagas, Ryan, & Marsit, 2007) and criminal justice (Spergel & Curry, 2005).

In the context of IRT, a latent variable is an unobservable characteristic of an experimental unit (de Ayala, 2004). Generally speaking, the reason variables are latent in psychology and education is because they are traits or abilities that cannot be measured without some degree of error (de Ayala, 2004; Hambleton & Swaminathan, 1985). For example, depression is a trait of interest to clinical psychologists (Shea *et al.*, 2009), but it cannot be observed directly. Instead clinicians and researchers observe behaviors related to or indicative of depression and infer the severity of a client's condition from those behaviors (de Ayala, 2004; Shea *et al.*, 2009). This inference leads to measurement error; that is, error arising from the indirect nature of the data collection process (Crocker & Algina, 1986; Lord & Novick, 1968), as opposed to stochastic error arising from sample selection (Hays, 1981). In psychology and education, experimental units are usually individuals from a population of interest, such as people suffering from depression or middle school students.

IRT models describe traits or abilities and estimate measurement error by using latent characterizations of both the individuals being measured and the behaviors used to measure them (de Ayala, 2004; Hambleton & Swaminathan, 1985; Lord & Novick, 1968). This requires a complex statistical model and a procedure for estimating the model's parameters (de Ayala, 2004; Baker & Kim, 2004; Hambleton & Swaminathan, 1985; Lord & Novick, 1968). Several models and parameter estimation procedures have been proposed since IRT was formalized in the early to mid-20th century (e.g. Birnbaum, 1968; Bock & Lieberman, 1970; Richards, 1936; Swaminathan & Gifford, 1982; 1985; 1986). The drawback to many of these procedures is that they require large samples of examinees to yield accurate parameter estimates (Thissen & Wainer, 1982). Large samples have not been an issue in large-scale, high-stakes achievement testing, such as the aforementioned GRE and SAT Tests, where IRT has been widely applied (Houseman *et al.*, 2007). However, in clinical psychology samples are smaller, which has led to the under-application of a method that could give clinicians valuable insights into their patients' behavior (Shea, *et al.*, 2009). Likewise, IRT's reliance on large samples has made it difficult to apply to many problems in domains outside of psychology and education (Houseman *et al.*, 2007).

Our goal in this paper is to demonstrate a new procedure for estimating the IRT models' parameters and to compare its parameter estimates to those of other commonly used estimation procedures. The method we propose, Regularized Marginal Maximum Likelihood (RMML), is based on regularized parameter estimation, which has recently been applied to a wide variety of models in the statistical literature, such as linear, logistic, and multinomial regression (Hoerl & Kennard, 1970; Nyquist, 1991; Tibshirani, 1996; Zou & Hastie, 2005), linear discriminant analysis (Friedman, 1988), the Cox proportional hazards model (Tibshirani, 1997), and latent class analysis (Houseman, Coull, & Betensky, 2006). In particular, regularized estimation has shown considerable promise when the number of parameters exceeds the number of

observations in the data set (Friedman *et al.,* 2010; Zou & Hastie, 2005). Regularization of the estimates is achieved by subtracting a penalty function of the model parameters from the objective function as it is optimized. This shrinks the parameter estimates, helping to control over-fitting, and allowing parameter estimation even with otherwise insufficient data (Friedman *et al.*, 2010; Tibshirani, 1995; Hoerl & Kennard, 1970; Zou & Hastie, 2005). Several penalty functions, such as ridge regression (Hoerl & Kennard, 1970), the Least Absolute Shrinkage and Selection Operator (LASSO; Tibshirani, 1995), and the elastic net (Zou & Hastie, 2005) have been proposed in the statistical literature. These penalties differ in the effects they produce on the parameter estimates (Friedman *et al.*, 2010). For example, applying a ridge penalty to linear or generalized linear models tends to group highly correlated predictor variables together, whereas the LASSO tends to select one of variable from such a set and drop the rest from the model (Friedman *et al.*, 2010). The penalties and their effects are described in greater detail in Section 2.3.

To date, regularized estimation has been applied to IRT in a limited way (Houseman *et al.*, 2007; Paolino, 2013; Tutz & Schauberger, 2015), and the efficacy of it as a general purpose estimation procedure has not been well explored. Houseman *et al.* (2007) proposed using a modification of Hoerl and Kennard's (1970) ridge penalty with the two-parameter logistic IRT model (Birnbaum, 1968; Maxwell, 1959) in the context of exploring whether epigenetic markers could predict a patient's proclivity to develop bladder cancer. Tutz and Schauberger (2015) applied Tibshirani's (1996) LASSO penalty to detect differential item functioning (DIF) in the Rasch model (Rasch, 1960). These two studies share several important features. First, their focus was primarily substantive rather than methodological, and they apply regularized estimation to IRT models to address substantive questions, rather than to investigate the accuracy of the parameter estimates. Second, neither paper compares regularized estimation to other estimation procedures, such as Maximum Likelihood or Bayesian

estimation. Finally, both Houseman *et al.*, (2007) and Tutz and Schauberger (2015) include exogenous observed variables in their IRT models, what we refer to in the sequel as the augmented IRT model. Thus, although both studies serve as demonstrations of different regularized IRT models, neither provides us with data by which we can evaluate the regularized IRT parameter estimates.

The only methodological evaluation of regularized IRT to date has been that carried out by Paolino (2013). He applied penalties to both the discrimination and difficulty parameters of the two-parameter logistic model, and obtained promising results with small samples. However, the algorithm Paolino (2013) describes violates several important assumptions of the IRT model. It also uses Joint Maximum Likelihood (JML; Birnbaum, 1968) as a basis for regularized estimation. JML item parameter estimates are well-documented as not necessarily being statistically consistent (Andersen, 1970; Bock & Lieberman, 1970; see also Neyman & Scott, 1948), a problem not solved by regularizing them. We will examine Paolino's estimation algorithm in greater depth in a subsequent section.

The method proposed in this paper uses Marginal Maximum Likelihood (MML; Bock & Aitkin, 1981; Bock & Lieberman, 1970) as a jumping off point for regularizing the IRT parameter estimates. Unlike JML, MML yields consistent item parameter estimates and, as we shall show, the Expectation Maximization algorithm (Bock & Aitkin, 1981; see also Dempster, Rubin, & Laird, 1977) commonly used to compute estimates can be readily modified to produce penalized parameter estimates. Furthermore, such estimation can be achieved while keeping IRT's assumptions intact.

In the next section we review the pertinent literature, beginning with the development of IRT. We then describe regularized parameter estimation and show how it can be applied to logistic regression, a model closely related to certain IRT models. We end our literature review with a more detailed examination of the previous efforts to apply regularized parameter estimation to IRT (Houseman *et al.*, 2007; Paolino, 2013;

Tutz & Schauberger, 2015). Following our literature review, we describe the RMML approach to item parameter estimation. We then explore the efficacy of RMML as a parameter estimation method for IRT through two Monte Carlo simulations. The first study investigated the effects of different penalties on the method's estimation error and compared the precision of RMML to MML and to Bayesian estimation (Mislevy, 1986; Swaminathan & Gifford, 1982; 1985; 1986). The second study examined the performance of RMML, MML, and Bayesian estimation in the presence of non-discriminating items. Finally, we conclude with a few remarks about our results' implications for regularized IRT.

2 Literature Review

2.1 The Development of Item Response Theory

Although interest in IRT has surged in the last few decades (Chang, 2015), the ideas underlying it have been discussed in the psychological and educational testing literature since the beginning of the 20th century. Hambleton and Swaminathan (1985, p. 4) trace IRT's origins back to Binet and Simon's pioneering work measuring the intelligence of French school children. Binet and Simon (1916) describe how they used a plot of the proportion of their subjects that correctly answered a question against their subjects' ages as an aid to determining the appropriate age range for each question, or item, on their test. These plots produced an *s*-shaped ogival curve, since a continuous variable (age) was being plotted against a variable with asymptotes at zero and one (the proportion of examinees that answered the item correctly) that describes the probability of a correct response as a function of age.

The curves described by Binet and Simon (1916) are related to a concept at the heart of IRT, the Item Characteristic Curve (ICC; Tucker, 1946). Unlike the curves used by Binet and Simon, modern ICCs have the latent trait being studied

plotted on the x-axis, as shown in Figure 1. The y-axis of the ICC plot is the probability of answering the item correctly, or in the keyed direction, at every level of the latent variable (Tucker, 1946). The IRT model is a mathematical description of the ICC, with parameters describing location of the curve's center, the curve's slope at its center, and the curve's upper and lower asymptotes, depending on how flexible the model needs to be in order to fit the data. Obviously, different kinds of data require different models. Data gathered from multiple choice exams often require three item parameters: one describing the curve's location, one describing the curve's slope, and one describing the lower asymptote to account for guessing behavior (de Ayala, 2004). Psychopathology data often also requires a fourth parameter, governing the upper asymptote, to account for such things as social desirability bias (Waller & Reise, 2010), while two parameters are often sufficient for fitting the model to personality data (Reise & Waller, 1990). In the epigenetic application described by Houseman *et al.* (2007) the authors had no need to be concerned over guessing or social desirability, and accordingly fit a two-parameter model with only the location and slope parameters. This two-parameter model is also frequently used in psychology and education when there is no reason to believe the upper- and lower-asymptotes are other than 1 and 0, respectively (de Ayala, 2004; Hambleton & Swaminathan, 1985). The curve in Figure 1 is centered at zero, has a slope of one at its midpoint, and has upper- and lower-asymptotes of 1 and 0, respectively, shown by the dashed red lines.

Fundamentally, an IRT model is a mathematical description of the ICC, where the goal is to use an individual's response pattern to quantify the trait being measured. The development of IRT has taken place primarily in the contexts of educational testing and psychological research (de Ayala, 2004; Hambleton & Swaminathan, 1985), with the result that much of the terminology used to describe IRT models is colored by these applications. For example, the individual responding to the prescribed set of

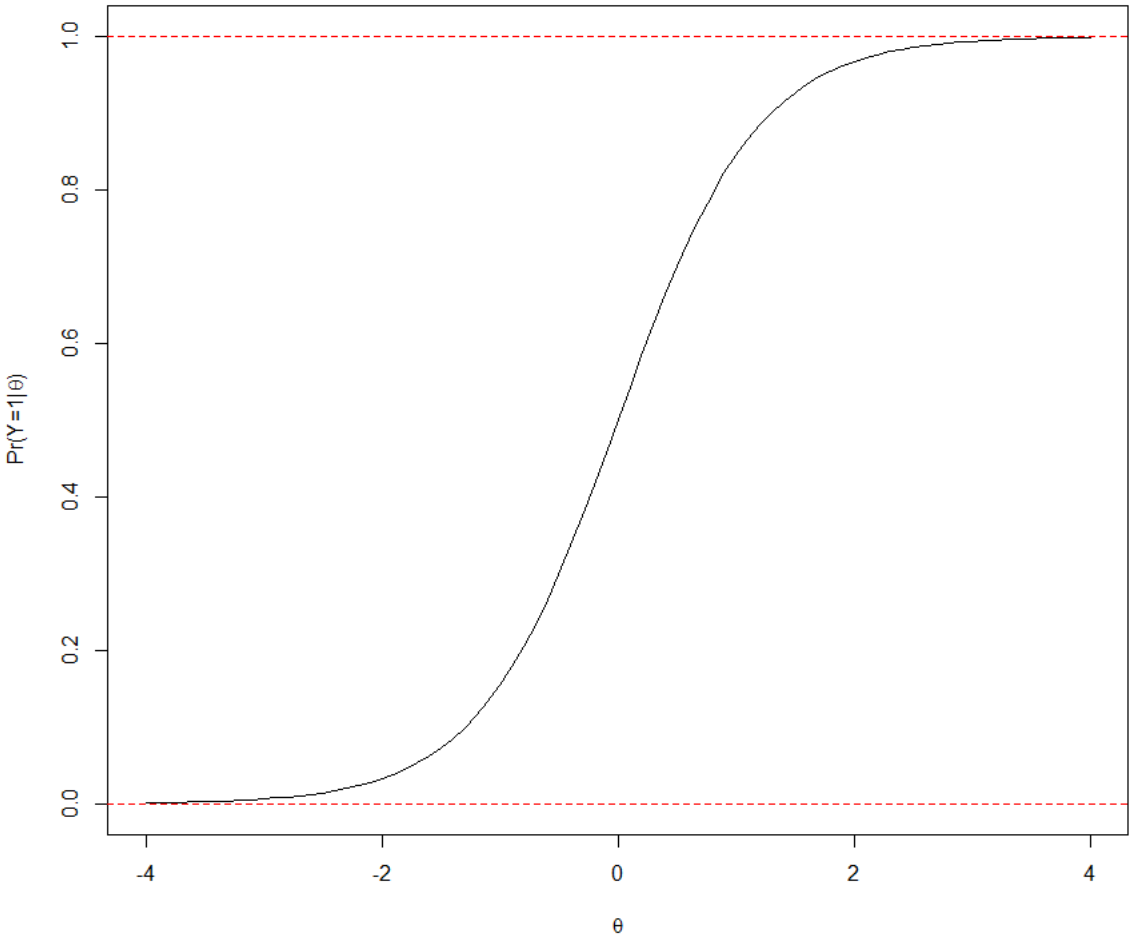

Figure 1: The Item Response Function of a Prototypical Binary Response Item. The latent trait is plotted along the x-axis and the probability of a correct response is plotted on the y-axis. The ogive gives the probability of a correct response at each latent trait value.

stimuli, such as a set of questions on a written test, is usually referred to as an examinee. Similarly, the stimuli an examinee responds to are referred to as items, and a set of items is called a test or exam. When an examinee responds to an item in a manner the indicates the presence of the latent trait being measured, they have answered the item correctly or in the keyed direction. When a examinee responds to an item in a manner that indicates the absence of the latent trait, they have answered the item incorrectly, or in an unkeyed direction. These terms are merely conventional shorthand – obviously on a personality test there are no correct or incorrect responses, hence the use of keyed and unkeyed.

IRT characterizes an examinee using one or more latent traits. As noted earlier, a latent trait is a facet of an individual's behavior or thought processes that is not directly observable, but which is hypothesized to influence behaviors that are observable. Conventional notation used in the IRT literature denotes a latent trait as θ, as on the x-axis of Figure 1. Although multidimensional IRT, in which an examinee is characterized by a composite of latent traits, is a growing area of research in IRT (Chang, 2015), the models examined in this paper are all unidimensional; that is, examinees are characterized by a single latent trait. A particular examinee's latent trait is denoted θ_i, $i = 1, \ldots, N$, where N is the number of examinees in the sample.

A test is composed of M items, each of which is described by between one and four item parameters. At the minimum, each item has a location or difficulty parameter, denoted b_j, $j = 1, \ldots, M$. Item difficulty defines the ICC's center and shares the same scale as the latent trait. If an examinee's ability is less than an item's difficulty, the probability of them correctly answering the item is less than .5. In addition to the difficulty parameter, an item might have parameters governing the slope of the ICC at its inflection point and the upper and lower asymptotes of the ICC. We will discuss these parameters in the context of the models in which they appear shortly.

Generally all of the items are held to have the same number of parameters, decided

a priori based on the characteristics of the data being analyzed or selected as part of a model fit study (Hambleton & Swaminathan, 1985, p. 151). Thus model selection is an important step in conducting an analysis using IRT. Selecting an inappropriate model can lead to a lack of fit between the model and data, which in turn gives rise to greater measurement error (Hambleton & Swaminathan, 1985, p. 151). Recently Houseman *et al.* (2007) proposed a technique for using regularized estimation of the discrimination parameters to partially automate model selection. We will discuss Houseman *et al.*'s approach in greater detail in Section 2.3.

The basis of an IRT model is the mathematical function used to describe the ICC. Early IRT models described the ICC using the normal probit function, where the probability of a correct response to item j conditional on the examinee's latent trait score, $P_j(\theta_i)$, is

$$P_j(\theta_i) = \int_{-\infty}^{a_j(\theta_i - b_j)} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz \tag{1}$$

(Lord, 1952). The model given by (1) has two item parameters, a_j and b_j. As noted earlier, b_j is the center of ICC, the item difficulty. The second parameter, a_j, is the slope of the ICC at b_j. The slope parameter alters the rate at which the probability of an examinee answering correctly changes as one moves from left to right along the θ continuum. For this reason a_j is referred to as the *discrimination* parameter, since items with rapidly changing probabilities (and steep ICCs) are better able to discriminate between two examinees with very similar latent trait scores near the center of the curve (Hambleton & Swaminathan, 1985, p. 36). The term z is a deviate from a normal distribution with mean b_j and variance $\frac{1}{a_j^2}$ (Hambleton & Swaminathan, 1985, p. 36). This model assumes that the asymptotes of the curve are 0 and 1, as in Figure 1. Lord (1952) described a family of normal ogive models that differed in the number of item parameters they estimated. The simplest normal ogive model had a single parameter describing the the center of the ICC, often called the *difficulty* parameter, since the further to the right the curve's center is situated on

the θ continuum the higher an examinee's latent trait score needs to be to have a high probability of answering the item correctly (Lord, 1952). The most flexible normal ogive model proposed by Lord (1952) had three parameters, the two parameters in (1) and a third parameter governing the lower asymptote.

Although the normal probit model describes the ICC well, it is difficult to use in parameter estimation because the derivative of the natural logarithm of (1) has no closed form solution (Baker & Kim, 2004). As a result, the normal ogive model is rarely used in modern psychometric analyses. Instead, the logistic model of the ICC popularized by Birnbaum (1968, p. 399; see also Maxwell, 1959) is used as a substitute. The most widely used logistic IRT model is the three-parameter logistic model (3PL; Birnbaum, 1968),

$$P_j(\theta_i) = c_j + (1 - c_j)\frac{\exp[Da_j(\theta_i - b_j)]}{1 + \exp[Da_j(\theta_i - b_j)]}, \tag{2}$$

where a_j and b_j are the discrimination and difficulty parameters, as described above, c_j is a parameter allowing for adjustments to the lower asymptote, and D is a scaling constant. Birnbaum (1968) showed that by setting the scaling constant to 1.702 the parameters estimated from the logistic model were virtually identical to those estimated from the normal ogive model. However, the logistic model has become so ubiquitous in IRT that many modern software packages leave $D = 1$, which produces parameters in the logistic metric. Unless otherwise noted we will use $D = 1.702$ throughout. The lower asymptote parameter, c_j, allows the model to asymptote to values other than 0. Since what is being modeled is a probability, c_j generally is not allowed to reduce the lower asymptote. However, there are reasons that the lower asymptote of an IRT model might be greater than 0. In educational testing, c_j is thought of as the *guessing* parameter, since even an examinee with infinitely low ability could answer correctly by shear chance if the number of responses was

limited (e.g., on a multiple choice test). A logical extension of (2), in which the upper-asymptote is allowed to deviate from one, is the four-parameter logistic model (4PL; Barton & Lord, 1981; McDonald, 1967),

$$P_j(\theta_i) = c_i + (d_i - c_i)\frac{\exp[Da_j(\theta_i - b_j)]}{1 + \exp[Da_j(\theta_i - b_j)]}, \tag{3}$$

which includes a parameter d_j that is similar to c_j. Again, since the IRT model is modeling a probability, d_j can only adjust the upper asymptote downwards. Although not widely used, the 4PL is an area of active research in psychometric theory (e.g., Waller & Feuerstahler, In Press) with applications to psychopathology assessment (Loken & Rulison, 2010; Reise & Waller, 2003; Waller & Reise, 2010) and computerized adaptive testing (e.g. Rulison & Loken, 2009).

Birnbaum (1968) also described two simplifications of the 3PL. For many applications guessing is not a concern (e.g. Houseman et al., 2007). Otherwise, the difficulty of estimating the guessing parameter may be sufficient, because very few examinees have sufficiently low ability, that it is omitted. In these situations, c_j is set to zero, reducing the 3PL the two-parameter logistic model (2PL; Birnbaum, 1968; Maxwell, 1959),

$$P_j(\theta_i) = \frac{\exp[Da_j(\theta_i - b_j)]}{1 + \exp[Da_j(\theta_i - b_j)]}. \tag{4}$$

The 2PL is of interest to us because it is closely related to the logistic regression model (Baker & Kim, 2004, p. 38), as we shall see shortly. Regularized estimation has already been widely applied to logistic regression with observed variables (Friedman *et al.*, 2010; Zhu & Hastie, 2005), as recognized by Paolino (2013) and Houseman *et al.* (2007) in their IRT applications.

Finally, if we force $a_j = 1$ for all m items on a test, we are left with the one

parameter logistic model (1PL; Birnbaum, 1968),

$$P_j(\theta_i) = \frac{\exp[D(\theta_i - b_j)]}{1 + \exp[D(\theta_i - b_j)]}. \tag{5}$$

An alternative model, proposed by Rasch (1966), is to set $a_j = \bar{a}$ for all m items. The Rasch model is less restrictive than the 1PL, but more restrictive than the 2PL, since it requires that all m items must have identical discrimination parameters, and that there is no guessing (Hambleton & Swaminathan, 1985, p. 46). The Rasch model is the easiest IRT model from which to estimate parameters, both because it contains fewer parameters to begin with, and also because sufficient statistics exist for latent trait estimates (Hambleton & Swaminathan, 1985, p. 138). However, many researchers have shown that the Rasch model is too inflexible to be practical for many situations encountered in testing (Birnbaum, 1968; Hambleton & Traub, 1973; Lord, 1968; Ross, 1966).

The effects of varying the difficulty and discrimination parameters can be seen in Figure 2. The top panel shows three ICCs with difficulty $b_j = 0$ but with different discrimination parameters. The dashed curve has the highest discrimination with $a_j = 1.5$, resulting in a curve with a very steep slope during its exponential phase. ICCs with steep slopes are highly desirable for several reasons. As mentioned earlier, when the ICC's slope is steep, the probability of two examinees with very similar latent trait values answering the item correctly is markedly different (de Ayala, 2004). This can be seen in Figure 2, where the probability of correctly answering the item represented by the dashed curve ($a_j = 1.5$) changes by a not insignificant amount for small steps along the θ continuum compared to either the items represented by the dotted ($a_j = 1.0$) or solid ($a_j = 0.5$). Furthermore, items with higher discrimination parameters provide more information about an examinee (Birnbaum, 1968). Information is measured

using the item information function,

$$I(\theta, y_j) = \frac{P_j'(\theta)^2}{P_j(\theta) Q_j(\theta)} \tag{6}$$

where $P_j(\theta)$ is given by (2), (4), or (5); $P_j'(\theta)$ is the first derivative of $P_j(\theta)$ with respect to θ; and $Q_j(\theta) = 1 - P_j(\theta)$. For the two-parameter logistic model, which will be our focus in the sequel, the item information is maximized at

$$I(\theta, y_j)_{max} = \frac{1}{4} D^2 a_j^2 \tag{7}$$

which is achieved when $\theta_i = b_j$[1] (Hambleton & Swaminathan, 1985, p. 106). Thus, the maximum information is a function of the item's discrimination parameter. Item information is important because its square root is the inverse of the standard error of $\hat{\theta}$ (Birnbaum, 1968).

The lower panel of Figure 2 also shows three ICCs with $a_j = 1$ and different difficulty parameters. The solid curve on the left is the easiest of the three items ($b_j = -1.5$), the curve in the center represents an item of average difficulty ($b_j = 0$), and the curve on the right represents the most difficult of the three items ($b_j = 1.5$). The reason b_j is thought of as the "difficulty" parameter is that as b_j increases in value, the center of the ICC shifts to the right, requiring an examinee to have a higher latent trait score to have a .5 probability of answering the item correctly (Birnbaum, 1968). Conversely, an item that requires a lower latent trait value to answer correctly is considered easier.

Mathematical functions alone do not make a model. In addition to a mathematical description of the ICC, we must specify the assumptions made about both the behavior of the phenomena being modeled and the data used to model it. One as-

[1] This is true for the one- and two-parameter logistic models. The three-parameter logistic model attains maximum information at $\frac{D^2 a_j^2}{8(1-c_j^2)} \left[1 - 20c_j - 8c_j^2 + (1 + 8c_j)^{3/2}\right]$. See Hambleton and Swaminathan (1985, p. 105) for details.

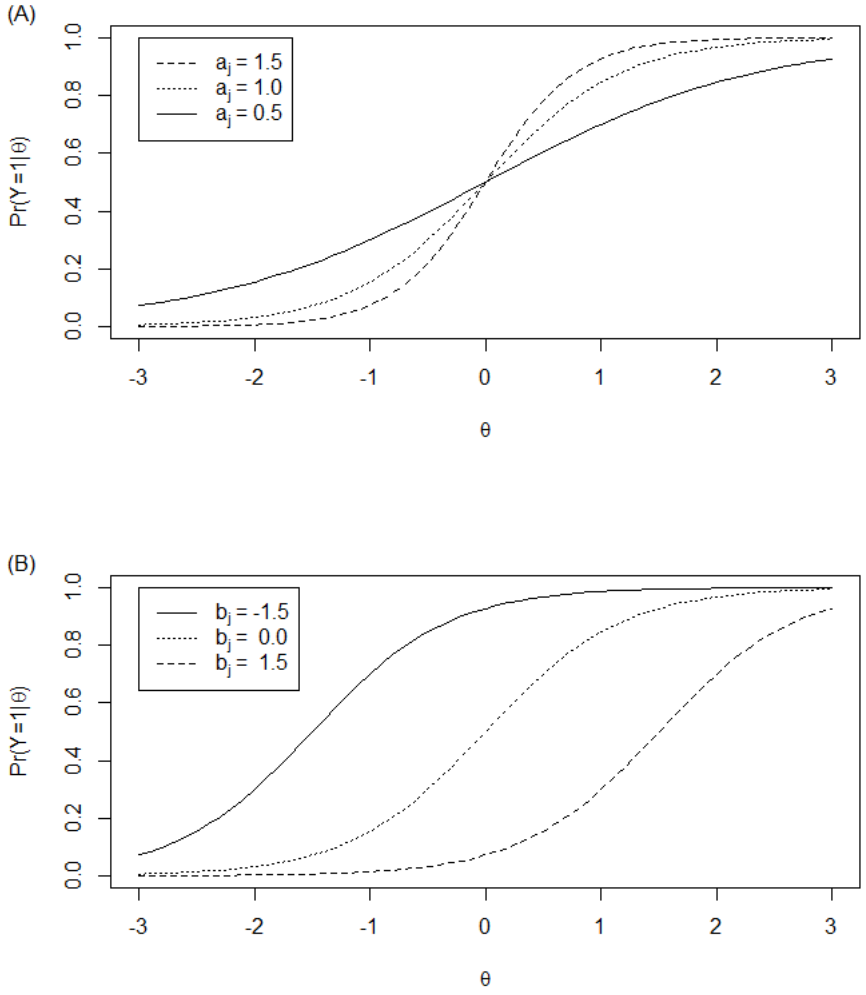

Figure 2: The Effects of Varying a_j and b_j in Birnbaum's 2PL.
Panel A (top) shows that as the discrimination parameter a_j is increased from 0.5 to 1.5, the slope of the ICC increases but the location remains unchanged. Panel B shows that as the difficulty parameter, b_j, is increased from -1.5 to 1.5 the center of the ICC moves from left to right down the θ continuum, but the slope remains unchanged.

sumption should already be clear from the preceding discussion: The probability of correctly answering an item is a monotonically increasing function of θ (Birnbaum, 1968, p. 405). In other words, examinees with higher θs have a higher probability of answering the item correctly across the entirety of the θ continuum and there is no point where a higher θ results in a lower probability of a correct answer. Note that this assumption is still true in the 4PL with the proviso that the probability need not every reach one.

In addition to assuming monotonicity, we assume that the latent trait space is completely specified in the model (Hambleton & Swaminathan, 1985, p. 17). That is, there is no latent trait that affects an examinee's responses that is not part of the model. Birnbaum (1968, p. 381) made the simpler assumption that there is only one latent trait affecting the response, or that the model is *unidimensional*. True unidimensionality is nearly always unobtainable in real testing situations (Hambleton & Swaminathan, 1985, p. 17). Instead, we require that a single latent trait is dominant across all of the items on a test (Hambleton & Swaminathan, 1985, p. 17). For example, unidimensionality is commonly violated when tests are timed; timing a test results in a second latent trait representing the examinee's ability to answer items quickly being introduced into the model (de Ayala, 2004, p. 21). Therefore, it is also common practice among test designers using IRT to assume the test is "unspeeded", or that all examinees have sufficient time to respond to all of the items (Birnbaum, 1968).

With respect to the observed data, we assume that an examinee's responses to the M items on a test are independent realizations of a Bernoulli random variable, Y_{ij}, with parameter $\pi = P_j(\theta_i)$(de Ayala, 2004, p. 21; Hambleton & Swaminathan, 1985. p. 23; Birnbaum, 1968, p. 399). In the IRT literature, this is known as Local Independence, and forms one of the building blocks of the likelihood function. Assuming local independence allows us to write the joint probability of observing all

m of an examinee's responses as,

$$\Pr\left(Y_{i1} = y_{i1}, \ldots, Y_{iM} = y_{iM} | \theta_i\right) = \prod_{j=1}^{M} \Pr\left(Y_{ij} = y_{ij} | \theta_i, a_j, b_j\right). \tag{8}$$

Stated in words, local independence assumes that the joint probability of observing an examinee's response pattern is equivalent to the product of each response's marginal probabilities conditional on θ (Hambleton & Swaminathan, 1985, p. 23). The IRT models popularized by Birnbaum (1968) allow only two possible responses: Correct (or keyed) and incorrect (or unkeyed). As a rule a correct response is coded such that $y_{ij} = 1$ and an incorrect response is coded such that $y_{ij} = 0$, where y_{ij} is the realization of Y_{ij} for the i^{th} examinee answering the j^{th}. This allows us to rewrite (8) as

$$\prod_{j=1}^{M} \Pr\left(Y_{ij} = y_{ij} | \theta_i, a_j, b_j\right) = \prod_{j=1}^{M} P_j\left(\theta_i\right)^{y_{ij}} Q_j\left(\theta_i\right)^{1-y_{ij}} \tag{9}$$

where $P_j\left(\theta_i\right) = \Pr\left(Y_{ij} = 1 | \theta_i\right)$ and $Q_j\left(\theta_i\right) = \Pr\left(Y_{ij} = 0 | \theta_i\right)$.

Some authors (e.g. Hambleton & Swaminathan, 1985, p. 24) have conflated local independence and unidimensionality. It is true that unidimensionality violations are also violations of local independence, since additional latent traits omitted from the model mean that create a dependency between responses not accounted for by (8), in essence invalidating the idea that the responses are independent. However, satisfying unidimensionality is not a sufficient condition for satisfying local independence (de Ayala, 2004, p. 21). For example, if the answer to one item is given by a subsequent or previous item, they may measure the same trait and yet not be independent.

In addition to monotonicity, unidimensionality, and local independence, we also assume that examinees respond to the items independently of one another (de Ayala, 2004, p. 21; Hambleton & Swaminathan, 1985, p. 24; Lord & Novick, 1968, p. 361). That is, we assume that one examinee's responses to the M items do not effect the responses of any other examinee. Given a sample of N examinees, this results in us

having $N \times M$ responses, all of which are independent conditional on θ. We organize the $N \times M$ responses into a response matrix, $\boldsymbol{Y} = \{y_{ij}\}$ (Lord & Novick, 1968, p. 362). When the assumptions listed above hold, the joint probability of observing the entire response matrix is

$$\Pr\left(\boldsymbol{Y} = \{y_{ij}\} \,|\boldsymbol{\theta}, \boldsymbol{a}, \boldsymbol{b}\right) = \prod_{i=1}^{N} \prod_{j=1}^{M} \Pr\left(Y_{ij} = y_{ij} | \theta_i, a_j, b_j\right) \tag{10}$$

where $\boldsymbol{\theta} = \{\theta_i\}$ is the N entry vector of latent trait parameters, $\boldsymbol{a} = \{a_j\}$ is the M entry vector of item discrimination parameters, and $\boldsymbol{b} = \{b_j\}$ is the M entry vector of item difficulty parameters (Birnbaum, 1968, p. 400). Using the more compact $P_j\left(\theta_i\right)$ and $Q_j\left(\theta_i\right)$ notation described above, we can the re-write (10) as

$$\Pr\left(\boldsymbol{Y} = \{y_{ij}\} \,|\boldsymbol{\theta}, \boldsymbol{a}, \boldsymbol{b}\right) = \prod_{i=1}^{N} \prod_{j=1}^{M} P_j\left(\theta_i\right)^{y_{ij}} Q_j\left(\theta_i\right)^{1-y_{ij}}, \tag{11}$$

(Birnbaum, 1968, p. 401).

Parameter estimation in IRT is usually conducted using either Maximum Likelihood or Bayesian techniques. In both approaches it is necessary to derive the likelihood function. The likelihood function is the joint probability function of the observed response matrix, $\boldsymbol{Y}$, expressed as a function of the unknown parameter vectors $\boldsymbol{\theta}$, $\boldsymbol{a}$, and $\boldsymbol{b}$ (DeGroot & Schervish, 2002, p. 355). The response matrix's joint probability when the assumptions hold, is given by (11),

$$L\left(\boldsymbol{\theta}, \boldsymbol{a}, \boldsymbol{b} | \boldsymbol{Y}\right) = \prod_{i=1}^{N} \prod_{j=1}^{M} P_j\left(\theta_i\right)^{y_{ij}} Q_j\left(\theta_i\right)^{1-y_{ij}} \tag{12}$$

(Birnbaum, 1968, p. 420; Hambleton & Swaminathan, 1985, p. 82). To estimate $\boldsymbol{\theta}$, $\boldsymbol{a}$, and $\boldsymbol{b}$ we find estimates $\hat{\boldsymbol{\theta}}$, $\hat{\boldsymbol{a}}$, and $\hat{\boldsymbol{b}}$ that maximize (12) for an observed response matrix (Birnbaum, 1968, p. 420). Several methods for estimating $\boldsymbol{\theta}$, $\boldsymbol{a}$, and $\boldsymbol{b}$ have been proposed in the literate (Baker & Kim, 2004). In the next few sections we review

some of the most prominent methods.

2.1.1 Joint Maximum Likelihood Estimation

Joint Maximum Likelihood (JML; Birnbaum, 1968) estimation was popularized by Birnbaum (1968) as a method for simultaneously estimating both the latent trait and item parameters in an IRT model. As with all maximum likelihood based estimation methods, the goal of JML is to estimate parameter values such that the likelihood function is maximized conditional on the observed data (Birnbaum, 1968; Bock & Lieberman, 1970; Hays, 1988). In the context of IRT the observed data is the response matrix $\boldsymbol{Y}$ and the likelihood function is given by (12)(Birnbaum, 1968, p. 420). From this point forward, we will assume that an IRT model with two item parameters, a_j and b_j such as the 2PL given by (4) is sufficient to describe the data (Birnbaum, 1968, p. 399; Maxwell, 1959). Our reasons for selecting this model are three-fold. First, the two-parameter logistic model is closely related to the logistic regression model (Baker & Kim, 2004; Houseman *et al.*, 2007; Paolino, 2013), as we will demonstrate momentarily. Regularized logistic regression is well-established for logistic regression in the statistical literature (Friedman, et al., 2010; Hastie, et al., 2009; Zhou & Hastie, 2005).. Second, the 2PL has a wide applicability both in psychology and education and in other fields (Houseman, *et al.*, 2007). Finally, the guessing parameter of the three-parameter model, c_j, is itself difficult to estimate accurately because estimating c_j relies on there being a low sample of low-ability examinees whose probability of answering the items correctly is predominantly a function of guessing (Hambleton & Swaminathan, 1985). Our goal is to better understand the effects of regularized estimation on the IRT model's discrimination parameter, therefore it seems prudent not to complicate the results by estimate the guessing parameter.

In order to simplify estimation and clarify the link between the 2PL and logistic regression, we re-express (4) in the slope-threshold parameterization (Baker & Kim,

2004). Though less intuitively appealing than the difficulty-discrimination form of the model given by (4), the slope-threshold parameterization is simpler to differentiate (Baker & Kim, 2004, p. 157) and results in a negative log-likelihood function that is convex in the parameter estimates (Baker & Kim, 2004, p. 157; Baker, 1988). To change the parameterization, we distribute a_j over $(\theta_i - b_j)$. This results in two terms inside the exponential function, $a_j\theta_i$ and $-a_j{b_j}^2$. We want to retain the latent trait score in the equation, but we rename $a_j = \beta_j$ and combine $-a_j b_j = \gamma_j$ so that the probability of the examinee correctly answering the item is written as

$$P_j^\star(\theta_i) = \frac{\exp(\gamma_j + \beta_j\theta_i)}{1 + \exp(\gamma_j + \beta_j\theta_i)} \tag{13}$$

and the probability of the examinee incorrectly answering the item is written as

$$\begin{aligned} Q_j^\star(\theta_i) &= 1 - P_j^\star(\theta_i), \\ &= \frac{1}{1 + \exp(\gamma_j + \beta_j\theta_i)} \end{aligned} \tag{14}$$

(Baker & Kim, 2004, p. 157). The relationship between the 2PL and the logistic regression model is also apparent from (13) and (14). The probability of a correct response is a linear function of the examinee's ability, the slope parameter β_j and the threshold parameter γ_j. The important difference between IRT and logistic regression is that the regressor variable in logistic regression is observed, whereas in IRT the regressor is the latent trait or ability parameter, which we must estimate. This fact significantly complicates IRT model estimation (Baker & Kim, 2004).

To account for the fact that the ability parameter is unknown at the beginning of estimation, Birnbaum (1968) proposed an iterative estimation algorithm in which the item parameter estimates are based on the θ estimates from the previous iteration of the algorithm, and then updates the θ estimate based on the current item parameter

[2]To keep our notation compact, we assume $D = 1$

estimates. At each iteration Birnbaum (1968, p. 405) estimated the parameters using the Newton-Raphson algorithm (Boyd & Vandenberge, 2004, p. 484; Hambleton & Swaminathan, 1985, p. 130).

The Newton-Raphson algorithm is itself an iterative algorithm for finding the minima or maxima of a twice differentiable function $f(\boldsymbol{x})$ that is a function of the p-dimensional vector $\boldsymbol{x}$. Let the vector $\boldsymbol{x}^{\star}$ be the vector of values that maximizes (or minimizes) the function f, and let $\boldsymbol{x}^{(t)}$ denote the value of $\boldsymbol{x}$ at the t^{th} iteration. A better approximation to $\boldsymbol{x}^{\star}$ can always be found by

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \left[f''\left(\boldsymbol{x}^{(t)}\right)\right]^{-1} f'\left(\boldsymbol{x}^{(t)}\right) \tag{15}$$

where f'' denotes the $p \times p$ matrix of second partial derivatives of f with respect to each element of $\boldsymbol{x}$ (i.e. the Hessian) and f' denotes the p-dimensional vector of first order partial derivatives of f with respect to each element of $\boldsymbol{x}$ (i.e. the gradient). The starting values of $\boldsymbol{x}^{(0)}$ may be any "convenient" value (Hambleton & Swaminathan, 1985, p. 130). The algorithm continues until some convergence criteria is met. This criteria may be on the estimates of $\boldsymbol{x}$ themselves, such as requiring that $\left|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right| < \alpha$ for some small $\alpha > 0$, on the difference between values of f at successive iterations, or that $f' < \alpha$ (i.e., the gradient is sufficiently close to zero).

In maximum likelihood the function f is the likelihood function (12). Taking the partial derivatives of the likelihood function can lead to round off errors, as the value of the function approaches zero as the number of terms increases. To avoid this problem, we instead take derivatives of the log-likelihood function

$$\begin{aligned} \ell(\theta, \beta, \gamma, |\boldsymbol{Y}) &= \ln\left[L(\theta, \beta, \gamma|\boldsymbol{Y})\right] \\ &= \sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} \ln\left[P_j^{\star}(\theta_i)\right] + (1 - y_{ij}) \ln\left[Q_j^{\star}(\theta_i)\right] \end{aligned} \tag{16}$$

where the function ln denotes the natural logarithm. Since the natural logarithm is

Table 1: First Partial Derivatives of the Two-Parameter Logistic Model

Derivative	Expression
$\frac{\partial}{\partial\theta_i}P_j^\star(\theta_i)$	$\beta_j P_j^\star(\theta_i)\,Q_j^\star(\theta_i)$
$\frac{\partial}{\partial\beta_j}P_j^\star(\theta_i)$	$\theta_i P_j^\star(\theta_i)\,Q_j^\star(\theta_i)$
$\frac{\partial}{\partial\gamma_j}P_j^\star(\theta_i)$	$P_j^\star(\theta_i)\,Q_j^\star(\theta_i)$
$\frac{\partial}{\partial\theta_i}Q_j^\star(\theta_i)$	$-\beta_j P_j^\star(\theta_i)\,Q_j^\star(\theta_i)$
$\frac{\partial}{\partial\beta_j}Q_j^\star(\theta_i)$	$-\theta_i P_j^\star(\theta_i)\,Q_j^\star(\theta_i)$
$\frac{\partial}{\partial\gamma_j}Q_j^\star(\theta_i)$	$-P_j^\star(\theta_i)\,Q_j^\star(\theta_i)$

a monotonic function, the estimates of θ, β, and γ that maximize (16) also maximize (12).

In order to use JML, we must take the first and second partial derivatives of (16) with respect to each of the parameters. These can be easily constructed using the partial derivatives of $P_j^\star(\theta_i)$ and $Q_j^\star(\theta_i)$ given in Table 1. Detailed derivations of these expressions can be found in Appendix A. As stated earlier, JML is a two-stage estimation algorithm. During the first stage, β and γ are treated as known, and θ is estimated for each examinee in the sample. To do this we have to take the first- and second partial derivatives of (16) with respect to θ. The first partial derivative of (16) with respect to θ_i is

$$\begin{aligned}\frac{\partial}{\partial\theta_i}\ell(\boldsymbol{\theta},\boldsymbol{\beta},\boldsymbol{\gamma}|\boldsymbol{Y}) &= \frac{\partial}{\partial\theta_i}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\ln\left[P_j^\star(\theta_i)\right]+(1-y_{ij})\ln\left[Q_j^\star(\theta_i)\right],\\ &= \sum_{j=1}^{M}\beta_j\left[y_{ij}-P_j^\star(\theta_i)\right]. \qquad (17)\end{aligned}$$

Since we have assumed that the N examinees respond independently, the form of the first partial derivative is the same for all N trait parameters. Taking all N partial derivatives produces the gradient vector $\boldsymbol{\ell'}$ (the equivalent of f' in (15)). The second partial derivative of ℓ is the first partial derivative of ℓ' with respect to θ_i. Note

that since we assume that the examinees respond independently to the M items, the derivative of (17) with respect to θ_i and then θ_k for $k \neq i$ is zero. Thus, rather than having to compute all $N \times N$ possible second partial derivatives, we only need to compute the N second partial derivatives with respect to θ_i twice. These are terms are given by

$$\begin{aligned} \frac{\partial^2}{\partial \theta_i^2} \ell\left(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\gamma} | \boldsymbol{Y}\right) &= \frac{\partial}{\partial \theta_i} \ell_i'\left(\theta_i, \boldsymbol{\beta}, \boldsymbol{\gamma} | \boldsymbol{Y}\right), \\ &= \frac{\partial}{\partial \theta_i} \sum_{j=1}^{M} \beta_j \left[y_{ij} - P_j^\star\left(\theta_i\right)\right], \\ &= -\sum_{j=1}^{M} \beta_j^2 P_j^\star\left(\theta_i\right) Q_j^\star\left(\theta_i\right). \end{aligned} \tag{18}$$

The results of this are organized into a Hessian matrix, $\boldsymbol{\ell}''$(equivalent to f'' in (15)). Using the Newton-Raphson step described by (15), the new θ estimates are

$$\hat{\boldsymbol{\theta}}^{(t+1)} = \hat{\boldsymbol{\theta}}^{(t)} - \left[\boldsymbol{\ell}''\left(\hat{\boldsymbol{\theta}}^{(t)}\right)\right]^{-1} \boldsymbol{\ell}'\left(\hat{\boldsymbol{\theta}}^{(t)}\right) \tag{19}$$

where $\hat{\boldsymbol{\theta}}^{(t)}$ is the vector of previous θ estimates and all other terms are as previously defined. Note that the domain of β is the positive real numbers and both $P_j^\star\left(\theta_i\right)$ and $Q_j^\star\left(\theta_i\right)$ are between 0 and 1, (18) is negative, indicating that the log-likelihood has at least a local maximum for $\hat{\boldsymbol{\theta}}^{(t)}$.

Having estimated the θs, we move onto the JML algorithm's second stage in which the θ estimates obtained when the Newton-Raphson algorithm converged in the first step are treated as known values and the item parameters are estimated (Baker &

Kim, 2004, p. 90). The first derivative of (16) with respect to β is

$$\begin{aligned}
\frac{\partial}{\partial\beta_j}\ell\left(\boldsymbol{\theta},\boldsymbol{\beta},\boldsymbol{\gamma}|\boldsymbol{Y}\right) &= \frac{\partial}{\partial\beta_j}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\ln\left[P_j^\star\left(\theta_i\right)\right] + \left(1-y_{ij}\right)\ln\left[Q_j^\star\left(\theta_i\right)\right], \\
&= \sum_{i=1}^{N}\frac{\partial}{\partial\beta_j}y_{ij}\ln\left[P_j^\star\left(\theta_i\right)\right] + \frac{\partial}{\partial\beta_j}\left(1-y_{ij}\right)\ln\left[Q_j^\star\left(\theta_i\right)\right], \\
&= \sum_{i=1}^{N} y_{ij}\frac{1}{P_j^\star\left(\theta_i\right)}\frac{\partial}{\partial\beta_j}P_j^\star\left(\theta_i\right) + \left(1-y_{ij}\right)\frac{1}{Q_j^\star\left(\theta_i\right)}\frac{\partial}{\partial\beta_j}Q_j^\star\left(\theta_i\right), \quad (20) \\
&= \sum_{i=1}^{N} y_{ij}\theta_i Q_j^\star\left(\theta_i\right) - \left(1-y_{ij}\right)\theta_i P_j^\star\left(\theta_i\right), \\
&= \sum_{i=1}^{N}\theta_i\left[y_{ij}Q_j^\star\left(\theta_i\right) - P_j^\star\left(\theta_i\right) + y_{ij}P_j^\star\left(\theta_i\right)\right], \\
&= \sum_{i=1}^{N}\theta_i\left[y_{ij} - P_j^\star\left(\theta_i\right)\right].
\end{aligned}$$

Similarly, the first partial derivative of (16) with respect to γ is

$$\begin{aligned}
\frac{\partial}{\partial\gamma_j}\ell\left(\boldsymbol{\theta},\boldsymbol{\beta},\boldsymbol{\gamma}|\boldsymbol{Y}\right) &= \frac{\partial}{\partial\gamma_j}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\ln\left[P_j^\star\left(\theta_i\right)\right] + \left(1-y_{ij}\right)\ln\left[Q_j^\star\left(\theta_i\right)\right], \\
&= \sum_{i=1}^{N}\frac{\partial}{\partial\gamma_j}y_{ij}\ln\left[P_j^\star\left(\theta_i\right)\right] + \frac{\partial}{\partial\gamma_j}\left(1-y_{ij}\right)\ln\left[Q_j^\star\left(\theta_i\right)\right], \\
&= \sum_{i=1}^{N} y_{ij}\frac{1}{P_j^\star\left(\theta_i\right)}\frac{\partial}{\partial\gamma_j}P_j^\star\left(\theta_i\right) + \left(1-y_{ij}\right)\frac{1}{Q_j^\star\left(\theta_i\right)}\frac{\partial}{\partial\gamma_j}Q_j^\star\left(\theta_i\right), \quad (21) \\
&= \sum_{i=1}^{N} y_{ij}Q_j^\star\left(\theta_i\right) - \left(1-y_{ij}\right)P_j^\star\left(\theta_i\right), \\
&= \sum_{i=1}^{N} y_{ij}Q_j^\star\left(\theta_i\right) - P_j^\star\left(\theta_i\right) + y_{ij}P_j^\star\left(\theta_i\right), \\
&= \sum_{i=1}^{N} y_{ij} - P_j^\star\left(\theta_i\right).
\end{aligned}$$

This results in a $2M$ long gradient vector,

$$\boldsymbol{\ell}' = \begin{bmatrix} \sum_{i=1}^{N} \theta_i \left[y_{i1} - P_1^\star \left(\theta_i \right) \right] \\ \vdots \\ \sum_{i=1}^{N} \theta_i \left[y_{iM} - P_M^\star \left(\theta_i \right) \right] \\ \sum_{i=1}^{N} y_{i1} - P_1^\star \left(\theta_i \right) \\ \vdots \\ \sum_{i=1}^{N} y_{iM} - P_M^\star \left(\theta_i \right) \end{bmatrix} \tag{22}$$

because each item has a slope and location parameter. Unlike the Hessian in the JML algorithm's first stage, the Hessian used to estimate the item parameters is block-diagonal, where each block consists of a 2×2 sub-matrix containing the second partial derivatives of (16) with respect the β twice, γ twice, β and then γ, and γ and then β. In actuality, it does not matter whether we differentiate (16) with respect to γ and then β or β and then γ. Both derivatives yield the same function. Thus there are three second-derivatives, $\frac{\partial^2}{\partial \beta_j^2} \ell$, $\frac{\partial^2}{\partial \gamma_j^2} \ell$, and $\frac{\partial^2}{\partial \beta_j \partial \gamma_j} \ell$. Other than the diagonal blocks, all of the elements of the Hessian matrix are zero because the items are assumed to be independent. For the second partial derivative of ℓ with respect to β twice we get,

$$\begin{aligned} \frac{\partial^2}{\partial \beta_j^2} \ell &= \frac{\partial}{\partial \beta_j} \sum_{i=1}^{N} \theta_i \left[y_{ij} - P_j^\star \left(\theta_i \right) \right], \\ &= \frac{\partial}{\partial \beta_j} \sum_{i=1}^{N} \theta_i y_{ij} - \theta_i P_j^\star \left(\theta_i \right), \\ &= \sum_{i=1}^{N} \frac{\partial}{\partial \beta_j} \theta_i y_{ij} - \frac{\partial}{\partial \beta_j} \theta_i P_j^\star \left(\theta_i \right), \\ &= -\sum_{i=1}^{N} \theta_i \frac{\partial}{\partial \beta_j} P_j^\star \left(\theta_i \right), \\ &= -\sum_{i=1}^{N} \theta_i^2 P_j^\star \left(\theta_i \right) Q_j^\star \left(\theta_i \right). \end{aligned} \tag{23}$$

The second partial derivative of ℓ with respect to γ twice is

$$\begin{aligned}
\frac{\partial^2}{\partial \gamma_j^2}\ell &= \frac{\partial}{\partial \gamma_j}\sum_{i=1}^{N} y_{ij} - P_j^{\star}\left(\theta_i\right), \\
&= \sum_{i=1}^{N}\frac{\partial}{\partial \gamma_j}y_{ij} - \frac{\partial}{\partial \gamma_j}P_j^{\star}\left(\theta_j\right), \\
&= -\sum_{i=1}^{N} P_j^{\star}\left(\theta_i\right)Q_j^{\star}\left(\theta_i\right).
\end{aligned} \tag{24}$$

Finally, the second partial derivative of ℓ with respect to β and γ is

$$\begin{aligned}
\frac{\partial^2}{\partial \beta_j \partial \gamma_j}\ell &= \frac{\partial}{\partial \gamma_j}\sum_{i=1}^{N}\theta_i\left[y_{ij} - P_j^{\star}\left(\theta_i\right)\right], \\
&= \frac{\partial}{\partial \gamma_j}\sum_{i=1}^{N}\theta_i y_{ij} - \theta_i P_j^{\star}\left(\theta_i\right), \\
&= \sum_{i=1}^{N}\frac{\partial}{\partial \gamma_j}\theta_i y_{ij} - \frac{\partial}{\partial \gamma_j}\theta_i P_j^{\star}\left(\theta_i\right), \\
&= -\sum_{i=1}^{N}\theta_i\frac{\partial}{\partial \gamma_j}P_j^{\star}\left(\theta_i\right) \\
&= -\sum_{i=1}^{N}\theta_i P_j^{\star}\left(\theta_i\right)Q_j^{\star}\left(\theta_i\right).
\end{aligned} \tag{25}$$

The fact that the Hessian is block-diagonal makes makes JML computationally efficient: Rather than inverting the entire $2M \times 2M$ Hessian matrix, we can invert each block independently, allowing us to optimize each item separately (Baker & Kim, 2004). As before, estimation is achieved by using the Newton-Raphson algorithm. After the Newton-Raphson algorithm converges, the new item parameter estimates are used in the first stage of the next iteration to obtain new estimates of the θs (Birnbaum, 1968). The JML algorithm iterates between estimating the θs and estimating the item parameters until a convergence criteria is reached. This may be that the estimates themselves change by less than some pre-specified amount or that the value of the log-likelihood function changes by less than some small value $\alpha > 0$. This

back and forth process led Harwell, Baker, and Zwarts (1988) to describe Birnbaum's JML algorithm as "the ping-pong method" for parameter estimation.

Although Birnbaum's method is intuitively appealing and computationally straight forward there are several problems with the algorithm that are not immediately obvious (Hambleton & Swaminathan, 1985, p. 133). The most serious of these is that the item parameter estimates produced by the JML are not statistically consistent. That is, as the sample size increases, the JML item parameter estimates do not converge to the true parameter values (Andersen, 1972; 1973; Neyman & Scott, 1948). This problem results from estimating the item parameters, the number of which is static as we increase the sample size, simultaneously with the θs, the number of which increases as we increase the sample size (Neyman & Scott, 1948). Neyman and Scott (1948) first recognized this problem in another context. They termed the set of parameters with constant ordinality *structural* parameters, and the set of parameters whose ordinality depends on the sample size *incidental* parameters. They showed that when both structural and incidental parameters are estimated simultaneously, the estimates of the structural parameters need not be consistent. In the context of JML, β and γ are structural parameters because their ordinality is invariant to sample size. The incidental parameters are the θs since increasing the sample size also increases the number of θs needing to be estimated.

To illustrate this, consider the following example given by Kendall and Stuart (1973, p. 61; see also Zellner 1971, p. 114-115; Hambleton & Swaminathan, 1985, p. 127-128). Suppose that X is a random variable measured on n populations. In each population, X has a different mean, but X has the same variance in all n populations. Let the means be denoted by $\mu_1, \mu_2, \ldots, \mu_n$ and the common variance be denoted σ^2. From each population we make k observations denoted as x_{ij}, $i = 1, \ldots, n$,

$j = 1, \ldots, k$ so that,

$$x_{ij} \sim N\left(\mu_i, \sigma^2\right) \; i = 1, \ldots, n; \; j = 1, \ldots, k \tag{26}$$

where $N\left(\mu, \sigma^2\right)$ represents the density function of the normal (Gaussian) distribution with mean μ and variance σ^2,

$$f\left(x_{ij} | \mu_i, \sigma^2\right) = \left(2\pi\sigma^2\right)^{-\frac{1}{2}} \exp\left[-(x_{ij}-\mu_i)^2/2\sigma^2\right]. \tag{27}$$

For notational convenience, we define $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{ik}]^T$. The likelihood of observing $[\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$ is

$$\begin{aligned} L\left(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n | \boldsymbol{\mu}, \sigma^2\right) &= \prod_{j=1}^{k} \left(2\pi\sigma^2\right)^{-\frac{n}{2}} \exp\left[-1/2 \sum_{i=1}^{n} -(x_{ij}-\mu_i)^2/2\sigma^2\right], \\ &= \left(2\pi\sigma^2\right)^{-\frac{nk}{2}} \exp\left[-1/2 \sum_{j=1}^{k} \sum_{i=1}^{n} -(x_{ij}-\mu_i)^2/2\sigma^2\right]. \end{aligned} \tag{28}$$

Taking the natural logarithm and necessary derivatives of this function shows us that the estimate of μ_i is,

$$\hat{\mu}_i = \frac{1}{k} \sum_{j=1}^{k} x_{ij}, \tag{29}$$

and the estimate of σ^2 is,

$$\hat{\sigma}^2 = \frac{1}{nk} \sum_{i=1}^{n} \sum_{j=1}^{k} \left(x_{ij} - \mu_i\right)^2. \tag{30}$$

The expectation of $\hat{\mu}_i$ is,

$$E\left(\hat{\mu}_i\right) = \mu_i, \tag{31}$$

so the estimates of the means are consistent. However, the expectation of $\hat{\sigma}^2$ is,

$$\begin{aligned} E\left(\hat{\sigma}^2\right) &= \sigma^2 \frac{kn-n}{kn}, \\ &= \sigma^2 \left(1 - \frac{1}{k}\right). \end{aligned} \tag{32}$$

For a fixed sample size (k), the bias in this last equation does not vanish as $n \to \infty$. The number of unknown parameters is always $n+1$, the n population means and the single common variance. The count obviously increases as n increases. Furthermore, as $n \to \infty$,

$$\frac{(n+1)}{nk} \to \frac{1}{k}. \tag{33}$$

Therefore, the estimate of the structural σ^2 parameter is not consistent as it does not converge to σ^2 as the sample size increases. Anderson (1972) demonstrates that something similar holds for the item parameter estimates in JML.

The solution to the Neyman-Scott problem is to estimate the structural parameters using their marginal log-likelihood; that is, to remove the incidental parameters from the estimation process. For the 1PL and Rasch models there is a simple solution because there are sufficient statistics for θ which can be substituted into the algorithm to get item parameter estimates (Andersen, 1972; 1973). Hambleton and Swaminathan (1985, p. 138) describe Andersen's approach, but since it cannot be used with the two-parameter model, we will not dwell on it. A second approach is to integrate over the distribution of the incidental parameters. This is the approach taken by Bock and Lieberman (1980) in their Marginal Maximum Likelihood (MML) algorithm, which we describe in the next section.

2.1.2 Marginal Maximum Likelihood Estimation

Several authors have proposed alternatives to Birnbaum's (1968) JML algorithm that yield consistent estimates of the item parameters (Andersen, 1973; Bock & Lieberman,

1970; Mislevy, 1986; Swaminathan & Gifford 1982; 1985; 1986). Of these methods, the most widely used algorithm for estimating item parameters for the 2PL is MML, proposed by Bock and Lieberman (1970; see also Bock & Aitkin, 1981). Like JML, MML is a maximum likelihood based algorithm, as opposed to a Bayesian algorithm such as those proposed by Swaminathan and Gifford (1982; 1985; 1986) and Mislevy (1986). Parameter estimates are the values that maximize the likelihood function, conditional on the observed data. Unlike JML, the likelihood function used in the MML algorithm is the marginal likelihood of the item parameters after integrating out θ (Bock & Lieberman, 1970; Harwell, Baker, & Zwarts, 1988). As a result, the item parameters are estimated without reference to θ, solving the problem described at the end of the previous section (Baker & Kim, 2004, p. 158; Bock & Lieberman, 1970). It should also be noted that MML estimation is, in a sense, more consistent with the theory of IRT than JML is. Earlier we stated that the items and examinees are independent by assumption. That means that the item parameter estimates should not rely on the sample of examinees taking the test. This is, of course, impossible in a strict sense. However, but integrating θ out of the estimation algorithm, we make the item parameter estimates conditional on the θ distribution, rather than the specific sample of θs (Baker & Kim, 2004).

The difference between JML and MML can be illustrated by analogy to the Analysis of Variance (ANOVA) model (Bock & Aitkin, 1981; Bock & Lieberman, 1970). Consider a situation in which we have two independent variables, each with two or more levels. If a variable's levels are the only levels of interest, then the variable is considered to have a fixed effect on the response (Hays, 1988). On the other hand, if a variable's levels are a random sample from a population of levels (e.g., experimental units), and our goal is to make inferences back to the population, then the variable is considered to have a random effect (Hays, 1988). For example, if our goal was to measure how the effectiveness of a new drug treatment differs for males and females,

our two variables are fixed effects: treatment level (drug or placebo) and gender (male or female). But what if we wanted to make an inference from a sample of participants back to the population we drew them from? Then the levels (participants) are a random effect. In IRT, JML treats both the items and examinees as fixed effects. Our interest is in estimating parameters for these particular items and for these particular examinees. In contrast, MML treats the items as a fixed effect and the examinees as a random effect (Bock & Aitkin, 1981; Bock & Lieberman, 1970). Our interest is in estimating parameters for these specific items for the entire examinee population, of which the sample we have data for is a random subset.

To estimate the item parameters, Bock and Lieberman (1970) first found the probability density of the latent trait conditional on the response vector, current item parameter estimates, and the marginal density of θ. To do this, we first denote the marginal density of θ as $g\left(\theta|\boldsymbol{\tau}\right)$, where $\boldsymbol{\tau}$ is a vector containing the parameters defining the θ density (Baker & Kim, 2004, p. 159). Let $\boldsymbol{u}_k$, $k = 1, \ldots, 2^M$ denote the k^{th} possible response vector. Note that $\boldsymbol{u}_k$ is not an *observed* response vector, as was the case for $\boldsymbol{y}_i$ when estimating the parameters using JML. Rather, $\boldsymbol{u}_k$ is one of the 2^M *possible* response vectors among M binary items (Hambleton & Swaminathan, 1985, p. 141). Assuming that θ is continuous, Bayes' Theorem (Hayes, 1988, p. 45) tells us that the posterior probability distribution of θ conditional on the item parameters, $\boldsymbol{u}_k$, and $g(\theta|\boldsymbol{\tau})$ is

$$\Pr\left(\theta|\boldsymbol{u}_k, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\tau}\right) = \frac{\Pr\left(\boldsymbol{u}_k|\theta, \boldsymbol{\beta}, \boldsymbol{\gamma}\right) g\left(\theta|\boldsymbol{\tau}\right)}{\int_\theta \Pr\left(\boldsymbol{u}_k|\theta, \boldsymbol{\beta}, \boldsymbol{\gamma}\right) g\left(\theta|\boldsymbol{\tau}\right) d\theta} \tag{34}$$

where $\int_\theta$ indicates that we must integrate with respect to θ across its entire domain. The denominator of (34) is the marginal joint probability of the response vector given

the item parameters and $g(\theta|\boldsymbol{\tau})$. Solving for this term yields,

$$\int_{\theta} \Pr\left(\boldsymbol{u}_k|\theta,\boldsymbol{\beta},\boldsymbol{\gamma}\right) g\left(\theta|\boldsymbol{\tau}\right) d\theta = \frac{\Pr\left(\boldsymbol{u}_k|\theta,\boldsymbol{\beta},\boldsymbol{\gamma}\right) g\left(\theta|\boldsymbol{\tau}\right)}{\Pr\left(\theta|\boldsymbol{u}_k,\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\tau}\right)}. \tag{35}$$

Assuming that the M items are locally independent, the first term in the integration on the left-hand side of (35) is the familiar expression,

$$\Pr\left(\boldsymbol{u}_k|\theta,\boldsymbol{\beta},\boldsymbol{\gamma}\right) = \prod_{j=1}^{M} P_j^{\star}\left(\theta\right)^{u_{kj}} Q_j^{\star}\left(\theta\right)^{1-u_{kj}}. \tag{36}$$

Assuming that the 2^M possible response vectors are independent, the marginal likelihood is proportional to

$$L \propto \prod_{k=1}^{2^M} \int_{\theta} \prod_{j=1}^{M} P_j^{\star}\left(\theta\right)^{u_{kj}} Q_j^{\star}\left(\theta\right)^{1-u_{kj}}. \tag{37}$$

Bock and Lieberman (1970) estimate the item parameters by finding vectors $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ that maximize (37). To do this they numerically integrated across the domain of θ. They then maximized the resulting equations using the Newton-Raphson algorithm, similar to Birnbaum's (1968) JML algorithm.

Although the MML algorithm described by Bock and Lieberman (1970) solves the Neyman-Scott problem and yields statistically consistent estimates of the item parameters (de Ayala, 2004), it proved too computationally expensive to be practical for test analysis (Baker & Kim, 2004, p. 157; Hambleton & Swaminathan, 1985). This is because the numerical integration over the θ probability distribution is difficult to accomplish on a computer. As a result, between its introduction in 1970 and its revision in 1981 (Bock & Aitkin, 1981), the MML algorithm was primarily of interest to methodological researchers, while JML continued to be widely used by practitioners (Baker & Kim, 2004; Hambleton & Swaminathan, 1985).

The revision of the MML algorithm proposed by Bock and Aitkin (1981) is more

computationally feasible than the algorithm originally described by Bock and Lieberman (1970). Their algorithm is based on the Expectation-Maximization (EM) algorithm developed by Dempster, Laird, and Rubin (1977) for estimating parameters from incomplete data. Bock and Aitkin's (1981) algorithm treats the unknown θ parameters as missing data, and applies the EM algorithm to calculate expected values based on the response patterns and current item parameter estimates. These expected values are then used to re-estimate the item parameters, and the process is repeated until either the item parameter estimates or the log-likelihood function value ceases to change between iterations (Baker & Kim, 2004; Harwell, Baker, & Zwarts, 1988).

To further simplify the estimation process, Bock and Aitkin (1981) replaced the difficulty-to-compute integral with an approximation by Gauss-Hermite quadrature. Gauss-Hermite quadrature is a method for approximating the integral of a continuous function with finite moments. The principle is illustrated in Figure 3. First, a finite number of equally spaced nodes are selected from the domain of the function to be approximated. Then the function value at each node is calculated. Finally, for each node, the area of a rectangle with length equal to the function's value at the node and width equal to the interval between nodes is computed. For any node, the integral between the left end of the function's domain and the node is approximately the sum of the areas of the rectangles of the nodes less than it. Another way of thinking about this process is that approximation by Gauss-Hermite quadrature is a Riemann sum in reverse (Harwell *et al.*, 1988). Finer approximations of the function's integral can be calculated by increasing the number of nodes.

Let x_k be the k^{th} node selected from the domain of $g(\theta|\boldsymbol{\tau})$ for $k = 1, \ldots, K$, and $A(x_k) = g(\theta = x_k|\boldsymbol{\tau})$ be the k^{th} quadrature weight (Baker & Kim, 2004; Bock & Aitkin, 1981). To calculate the posterior probability of the observed response vector given that $\theta_i = x_k$ and the current item parameter estimates, we substitute x_k and

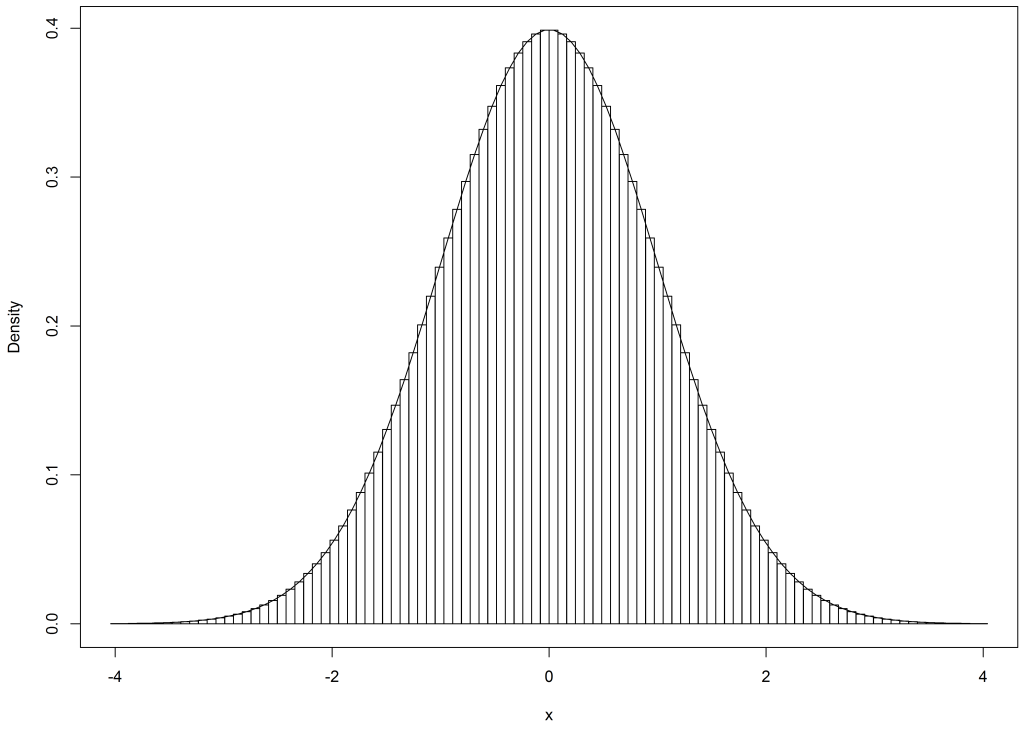

Figure 3: Gauss-Hermite Quadrature

$A(x_k)$ into the denominator of (34) resulting in

$$\Pr(\theta_i = x_k|\boldsymbol{y}_i, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\tau}) = \frac{\Pr(\boldsymbol{y}_i|x_k, \boldsymbol{\beta}, \boldsymbol{\gamma}) A(x_k)}{\sum_{k=1}^{K} \Pr(\boldsymbol{y}_i|x_k, \boldsymbol{\beta}, \boldsymbol{\gamma}) A(x_k)}. \tag{38}$$

Solving for the denominator yields,

$$\sum_{k=1}^{K} \Pr(\boldsymbol{y}_i|x_k, \boldsymbol{\beta}, \boldsymbol{\gamma}) A(x_k) = \frac{\Pr(\boldsymbol{y}_i|x_k, \boldsymbol{\beta}, \boldsymbol{\gamma}) A(x_k)}{\Pr(\theta_i = x_k|\boldsymbol{y}_i, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\tau})}, \tag{39}$$

which replaces the intractable integration on the left-hand side with a more tractable summation. Assuming local independence the probability of the observed response vector given $\theta_i = x_k$ and the item parameters is,

$$\Pr(\boldsymbol{y}_i|x_k, \boldsymbol{\beta}, \boldsymbol{\gamma}) = \prod_{j=1}^{m} P_j^{\star}(x_k)^{y_{ij}} Q_j^{\star}(x_k)^{1-y_{ij}}, \tag{40}$$

which means that the left hand side of 39 is the weighted sum of the probabilities of each examinee having a latent trait score equal to the k^{th} quadrature node, where the weights are the $A(x_k)$ (i.e., the density of $g(\theta|\boldsymbol{\tau})$ at x_k). If we substitute (40) into (39), we get

$$\sum_{k=1}^{K} \prod_{j=1}^{m} P_j^{\star}(x_k)^{y_{ij}} Q_j^{\star}(x_k)^{1-y_{ij}} A(x_k) = \frac{\prod_{j=1}^{m} P_j^{\star}(x_k)^{y_{ij}} Q_j^{\star}(x_k)^{1-y_{ij}} A(x_k)}{\Pr(\theta_i = x_k|\boldsymbol{y}_i, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\tau})}. \tag{41}$$

The EM algorithm described by Bock and Aitkin (1981) for estimating the item parameters proceeds in two steps.

E-Step The goal in the E-Step is to compute the expected number of examinees with $\theta_i = x_k$ and expected number of correct responses to each item from the examinees with $\theta_i = x_k$. Baker and Kim (2004, p. 171) refer to these two quantities as the "artificial data", although it would be more accurate to say they are the expected data. Given an observed response matrix and provisional item parameter estimates,

$\hat{\boldsymbol{\beta}} = \left\{\hat{\beta}_j\right\}$ and $\hat{\boldsymbol{\gamma}} = \{\hat{\gamma}_j\}$, the expected number of examinees with $\theta_i = x_k$ is,

$$\bar{n}_k = \sum_{i=1}^{N} \left[\frac{\prod_{j=1}^{M} P_j^{\star}(x_k)^{y_{ij}} Q_j^{\star}(x_k)^{1-y_{ij}} A(x_k)}{\sum_{k=1}^{K} \prod_{j=1}^{M} P_j(x_k)^{y_{ij}} Q_j(x_k)^{1-y_{ij}} A(x_k)} \right], \tag{42}$$

where the term in brackets is the posterior probability of the i^{th} examinee having a latent trait score equal to x_k given by (38). The expected number of correct answers to the j^{th} item given by examinees $\theta_i = x_k$ in a group of N examinees is

$$\bar{r}_{jk} = \sum_{i=1}^{n} \left[\frac{\prod_{j=1}^{m} y_{ij} P_j^{\star}(x_k)^{y_{ij}} Q_j^{\star}(x_k)^{1-y_{ij}} A(x_k)}{\sum_{k=1}^{K} \prod_{j=1}^{m} P_j^{\star}(x_k)^{y_{ij}} Q_j^{\star}(x_k)^{1-y_{ij}} A(x_k)} \right]. \tag{43}$$

Equation (42) arises directly from (38). To compute the expected number of examinees with $\theta_i = x_k$ we sum the probability for each examinee across examinees. Equation (43) arises in a similar manner. The difference between (42) and (43) is that in the latter we multiply the probabilities by the elements of the response matrix removing terms that represent items an examinee answered incorrectly. Therefore, the result of (43) is the expected number of correct responses to the j^{th} item among examinees with $\theta_i = x_k$. Also note that $\bar{r}_{jk} \leq \bar{n}_k$, since we cannot expect more correct answers from examinees with $\theta_i = x_k$ than we have examinees with $\theta_i = x_k$.

M-Step Using the artificial data and the numeric quadrature, the item parameters are estimated by maximizing the grouped likelihood function shown in (44). Using the artificial data, numeric quadrature, and assuming local independence, the likelihood functions is

$$L\left(\boldsymbol{\beta}, \boldsymbol{\gamma} | \boldsymbol{x}, \bar{\boldsymbol{n}}, \bar{\boldsymbol{R}}\right) = \prod_{k=1}^{K} \prod_{j=1}^{M} P_j^{\star}(x_k)^{\bar{r}_{jk}} Q_j^{\star}(x_k)^{\bar{n}_k - \bar{r}_{jk}} \tag{44}$$

where $\boldsymbol{x} = \{x_k\}$ is the $K \times 1$ vector of quadrature nodes, $\bar{\boldsymbol{n}}_{K\times 1} = \{\bar{n}_k\}$, and $\bar{\boldsymbol{R}}_{M\times K} = \{\bar{r}_{jk}\}$. Note that $\bar{n}_k$ and $\bar{r}_{jk}$ *do not* depend on the item parameters as they were computed using $\hat{\beta}_j$ and $\hat{\gamma}_j$. For the purposes of the M-Step $\bar{n}_k$ and $\bar{r}_{jk}$ are

constants. As in JML, it is more accurate to maximize the log-likelihood function

$$\ell\left(\boldsymbol{\beta},\boldsymbol{\gamma}|\boldsymbol{x},\bar{\boldsymbol{n}},\bar{\boldsymbol{R}}\right)=\sum_{k=1}^{K}\sum_{j=1}^{M}\left\{\bar{r}_{jk}\ln\left[P_j^{\star}\left(x_k\right)\right]+\left(\bar{n}_k-\bar{r}_{jk}\right)\ln\left[Q_j^{\star}\left(x_k\right)\right]\right\}, \tag{45}$$

than it is maximize (44) directly. Our assumption that the responses are independent allows us to perform this optimization separately for each item using the Newton-Raphson algorithm (Baker & Kim, 2004, p. 87; Bock & Aitkin, 1981; Harwell *et al.*, 1988). This requires us to take the first and second partial derivative of (45) with respect to β and γ.

The first derivative of (45) with respect to β_j is

$$\begin{aligned}
\frac{\partial}{\partial\beta_j}\ell\left(\boldsymbol{\beta},\boldsymbol{\gamma}|\boldsymbol{x},\bar{\boldsymbol{n}},\bar{\boldsymbol{R}}\right) &= \frac{\partial}{\partial\beta_j}\sum_{k=1}^{K}\sum_{j=1}^{M}\bar{r}_{jk}\ln\left[P_j^{\star}\left(x_k\right)\right]+\left(\bar{n}_k-\bar{r}_{jk}\right)\ln\left[Q_j^{\star}\left(x_k\right)\right], \\
&= \sum_{k=1}^{K}\frac{\partial}{\partial\beta_j}\bar{r}_{jk}\ln\left[P_j^{\star}\left(x_k\right)\right]+\frac{\partial}{\partial\beta_j}\left(\bar{n}_k-\bar{r}_{jk}\right)\ln\left[Q_j^{\star}\left(x_k\right)\right], \\
&= \sum_{k=1}^{K}\bar{r}_{jk}\frac{P_j^{\star}\left(x_k\right)Q_j^{\star}\left(x_k\right)}{P_j^{\star}\left(x_k\right)}x_k-\left(\bar{n}_k-\bar{r}_{jk}\right)\frac{P_j^{\star}\left(x_k\right)Q_j^{\star}\left(x_k\right)}{Q_j^{\star}\left(x_k\right)}x_k, \\
&= \sum_{k=1}^{K}\bar{r}_{jk}x_kQ_j^{\star}\left(x_k\right)-\left(\bar{n}_k-\bar{r}_{jk}\right)x_kP_j^{\star}\left(x_k\right), \\
&= \sum_{k=1}^{K}x_k\left[\bar{r}_{jk}-\bar{n}_kP_j^{\star}\left(x_k\right)\right].
\end{aligned} \tag{46}$$

The first derivative of (45) with respect to γ_j is

$$\begin{aligned}
\frac{\partial}{\partial\gamma_j}\ell\left(\boldsymbol{\beta},\boldsymbol{\gamma}|\boldsymbol{x},\bar{\boldsymbol{n}},\bar{\boldsymbol{R}}\right) &= \frac{\partial}{\partial\gamma_j}\sum_{k=1}^{K}\sum_{j=1}^{M}\bar{r}_{jk}\ln\left[P_j^\star\left(x_k\right)\right]+\left(\bar{n}_k-\bar{r}_{jk}\right)\ln\left[Q_j^\star\left(x_k\right)\right], \\
&= \sum_{k=1}^{K}\frac{\partial}{\partial\gamma_j}\bar{r}_{jk}\ln\left[P_j^\star\left(x_k\right)\right]+\frac{\partial}{\partial\gamma_j}\left(\bar{n}_k-\bar{r}_{jk}\right)\ln\left[Q_j^\star\left(x_k\right)\right], \\
&= \sum_{k=1}^{K}\bar{r}_{jk}\frac{P_j^\star\left(x_k\right)Q_j^\star\left(x_k\right)}{P_j^\star\left(x_k\right)}-\left(\bar{n}_k-\bar{r}_{jk}\right)\frac{P_j^\star\left(x_k\right)Q_j^\star\left(x_k\right)}{Q_j^\star\left(x_k\right)}, \quad (47) \\
&= \sum_{k=1}^{K}\bar{r}_{jk}Q_j^\star\left(x_k\right)-\left(\bar{n}_k-\bar{r}_{jk}\right)P_j^\star\left(x_k\right), \\
&= \sum_{k=1}^{K}\bar{r}_{jk}-\bar{n}_kP_j^\star\left(x_k\right).
\end{aligned}$$

In addition to the two first partial derivatives needed to form the gradient vector, we also need the three partial second derivatives, $\frac{\partial^2}{\partial\beta_j^2}\ell$, $\frac{\partial^2}{\partial\gamma_j^2}\ell$, and $\frac{\partial^2}{\partial\beta_j\partial\gamma_j}\ell$, that comprise the Hessian matrix. As before, assuming local independence means that the Hessian is block-diagonal (Bock & Aitkin, 1981). Each block is a 2×2 matrix of second partial derivatives of ℓ of the form

$$\boldsymbol{H}_j=\begin{bmatrix}\frac{\partial^2\ell}{\partial\beta_j^2} & \frac{\partial^2\ell}{\partial\beta_j\partial\gamma_j}\\ \frac{\partial^2\ell}{\partial\beta_j\partial\gamma_j} & \frac{\partial^2\ell}{\partial\gamma_j^2}\end{bmatrix}. \quad (48)$$

The partial second derivative of ℓ with respect to β_j twice is

$$\begin{aligned}
\frac{\partial^2}{\partial\beta_j^2}\ell &= \frac{\partial}{\partial\beta_j}\sum_{k=1}^{K}x_k\left[\bar{r}_{jk}-\bar{n}_kP_j^\star\left(x_k\right)\right], \\
&= \sum_{k=1}^{K}\frac{\partial}{\partial\beta_j}x_k\bar{r}_{jk}-\frac{\partial}{\partial\beta_j}x_k\bar{n}_kP_j^\star\left(x_k\right), \quad (49) \\
&= -\sum_{k=1}^{K}x_k\bar{n}_k\frac{\partial}{\partial\beta_j}P_j^\star\left(x_k\right), \\
&= -\sum_{k=1}^{K}x_k^2\bar{n}_kP_j^\star\left(x_k\right)Q_j^\star\left(x_k\right).
\end{aligned}$$

With respect to γ_j twice, the second partial derivative of ℓ is

$$\begin{aligned}
\frac{\partial^2}{\partial \gamma_j^2}\ell &= \frac{\partial}{\partial \gamma_j}\sum_{k=1}^{K}\bar{r}_{jk} - \bar{n}_k P_j^\star\left(x_k\right), \\
&= \sum_{k=1}^{K}\frac{\partial}{\partial \gamma_j}\bar{r}_{jk} - \frac{\partial}{\partial \gamma_j}\bar{n}_k P_j^\star\left(x_k\right), \\
&= -\sum_{k=1}^{K}\bar{n}_k\frac{\partial}{\partial \gamma_j}P_j^\star\left(x_k\right), \\
&= -\sum_{k=1}^{K}\bar{n}_k P_j^\star\left(x_k\right)Q_j^\star\left(x_k\right).
\end{aligned} \tag{50}$$

Finally, the second partial derivative of ℓ with respect to β_j and γ_j is

$$\begin{aligned}
\frac{\partial^2}{\partial \beta_j \partial \gamma_j}\ell &= \frac{\partial}{\partial \gamma_j}\sum_{k=1}^{K}x_k\left[\bar{r}_{jk} - \bar{n}_k P_j^\star\left(x_k\right)\right], \\
&= \sum_{k=1}^{K}\frac{\partial}{\partial \gamma_j}x_k\bar{r}_{jk} - \frac{\partial}{\partial \gamma_j}x_k\bar{n}_k P_j^\star\left(x_k\right), \\
&= -\sum_{k=1}^{K}x_k\bar{n}_k\frac{\partial}{\partial \gamma_j}P_j^\star\left(x_k\right), \\
&= -\sum_{k=1}^{K}x_k\bar{n}_k P_j^\star\left(x_k\right)Q_j^\star\left(x_k\right).
\end{aligned} \tag{51}$$

With these equations in hand, the parameters are estimated iteratively using the Newton-Raphson step

$$\begin{bmatrix}\hat{\beta}_j \\ \hat{\gamma}_j\end{bmatrix}^{(t+1)} = \begin{bmatrix}\hat{\beta}_j \\ \hat{\gamma}_j\end{bmatrix}^{(t)} - \boldsymbol{H}_j^{-1}\begin{bmatrix}\frac{\partial}{\partial \beta_j}\ell \\ \frac{\partial}{\partial \gamma_j}\ell\end{bmatrix} \tag{52}$$

where t indexes the Newton-Raphson iterations. As in the JML algorithm, the Newton-Raphson algorithm continues to update the item parameter estimates until either the estimates remain unchanged between iterations or the gradient vector decreases to zero. Once the Newton-Raphson algorithm has converged, the EM al-

gorithm returns the item parameter estimates to the E-Step and re-calculates the expected data, $\bar{n}_k$ and $\bar{r}_{jk}$. This back and forth process continues until the parameter estimates converge. After the item parameters have been estimated, the latent trait parameters can be estimated by finding the $\hat{\boldsymbol{\theta}}$ values that maximize (12) using the Newton-Raphson algorithm or a similar method.

Bock and Aitkin's EM-algorithm for MML is the most widely used IRT parameter estimation, however, it is not the only method of solving the Neyman-Scott problem. Another approach is to use Bayesian estimation (Mislevy, 1987; Swaminathan & Gifford, 1982; 1985; 1986). From a mechanical standpoint, Bayesian estimation is an extension of MML in which we give all of the parameters a prior distribution, rather than only giving such a distribution to θ. However, from the theoretical vantage point, Bayesian estimation and Maximum Likelihood estimation make very different assumptions about the model.

In addition to addressing the Neyman-Scott problem, Bayesian estimation has been widely used to estimate IRT models from small samples (Gao & Chen, 2005). This is because the use of prior distributions solves many of the problems associated with small sample estimation. Since these algorithms represent the current state-of-the-art for small sample IRT estimation, in the next section we review them.

2.1.3 Bayesian Estimation

In Sections 2.1.1 and 2.1.2 we described the two most common maximum likelihood algorithms for estimating the IRT model. Although these methods are still commonly used in educational and psychological measurement, the accuracy of both methods suffer when the sample of examinees is small (Gao & Chen, 2005; Paolino, 2013). When estimating IRT parameters from a small sample is necessary, psychometricians often use Bayesian methods (Mislevy, 1986; Swaminathan & Gifford, 1985; Gao & Chen, 2005). The chief difference between Maximum Likelihood and Bayesian estima-

tion and maximum likelihood estimation is what we consider a random variable. In maximum likelihood estimation, the observed data are considered realizations of one or more random variables, and the parameters are considered fixed, albeit unknown, constants (Hays, 1988). In the specific case of IRT, we consider the responses to be realizations of Bernoulli random variables with success probabilities given by (13), and consider θ, γ, and β to be fixed constants.

Under the Bayesian paradigm, the opposite is true. Once the data have bee observed, they are considered known, fixed quantities. The parameters are considered random variables, each following a probability distribution (DeGroot & Schervish, 2002, p. 346). The probability distributions that the parameters follow are called prior distributions, since they serve to quantify the prior information about the parameters. By observing the data, we learn more about the parameters. The combination of the prior distribution and the new information gleaned from the observed data is called the posterior distribution (DeGroot & Schervish, 2002, p. 66). This posterior distribution can, in turn, become the prior distribution for our next estimation, using a new sample of data. For this reason, the process of Bayesian estimation has often been likened to the scientific method: when new data about a phenomena are gathered, prior understanding is updated to accommodate it.

The effect of using a prior distribution is to create a bias-variance trade-off (DeGroot & Schervish, 2002). That is, the prior leads to biased parameter estimates that are less variable than the unbiased Maximum Likelihood estimates (DeGroot & Schervish, 2002). The bias in Bayesian estimation is always towards the prior distribution's center, hence care is needed when selecting a prior. A prior distribution ought to be selected based on previous knowledge of the parameter , and when a dearth of such knowledge exists a non-informative prior should be used (Baker & Kim, 2004; Hambleton & Swaminathan, 1985). Indeed, one criticism of Bayesian estimation is that the prior distribution is always at least partially subjective. With sufficient data,

this subjectivity is irrelevant, since the likelihood function overwhelms the influence of the prior (DeGroot & Schervish, 2002, p.346). However, with relatively little data, the prior distribution's influence can bias the estimates. A prior distribution that reflects a strong belief about the parameter values is said to be informative, whereas one that reflects greater uncertainty is said to be non-informative. When Bayesian estimation uses a non-informative prior, the Bayesian and maximum likelihood estimations will be very similar, and the Bayesian estimates can be unbiased. That is not, however, to say they are the same, because the fundamental difference described above still exists. Furthermore, using an uniformative prior yields little benefit in the case of a small sample (Gao & Chen, 2005; Swaminathan & Gifford, 1985).

As noted earlier, MML (Bock & Aitkin, 1981; Bock & Lieberman, 1970) is a pseudo-Bayesian procedure because it uses a prior distribution on θ to integrate the latent trait out of the model (de Ayala, 2004; Baker & Kim, 2004; Hambleton & Swaminathan, 1985). However, unlike a true Bayesian estimation, MML does not use the prior to compute the posterior distribution of θ and get point estimates, and no priors are specified for the item parameters (Bock & Aitkin, 1981; Bock & Lieberman, 1970).

Swaminathan and Gifford (1985) described a fully Bayesian approach to estimating IRT parameters. Like MML, Marginal Bayesian estimation does not estimate the θ parameters. Instead it uses the θ prior the same way as MML, to integrate θ out of the model. It then estimates the item parameters with reference to the θ prior, rather than the θ estimates for the observed sample. Swaminathan and Gifford (1985) described this process for the 2PL. Under the same assumptions made for the Maximum Likelihood algorithms described in the two previous sections, the posterior distribution of the parameters $\boldsymbol{\theta}$, $\boldsymbol{a}$, and $\boldsymbol{b}$ conditional on the observed responses $\boldsymbol{Y}$ is expressed as,

$$\Pr\left(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\gamma} | \boldsymbol{Y}\right) \propto L\left(\boldsymbol{Y} | \boldsymbol{\theta}, \boldsymbol{a}, \boldsymbol{b}\right) \Pr\left(\boldsymbol{\theta}, \boldsymbol{a}, \boldsymbol{b}\right), \tag{53}$$

where $L(\boldsymbol{Y}|\boldsymbol{\theta}, \boldsymbol{a}, \boldsymbol{b})$ is the likelihood function given by (12) and $\Pr(\boldsymbol{\theta}, \boldsymbol{a}, \boldsymbol{b})$ is the joint prior distribution of the parameters. Swaminathan and Gifford (1985) assumed that the parameters $\boldsymbol{\theta}$, $\boldsymbol{a}$, and $\boldsymbol{b}$ are independent such that,

$$\Pr(\boldsymbol{\theta}, \boldsymbol{a}, \boldsymbol{b}) = \Pr(\boldsymbol{\theta}) \Pr(\boldsymbol{a}) \Pr(\boldsymbol{b}). \tag{54}$$

They further assumed that the examinees (and thus the ability parameters) are a random sample from the population and, that for the purposes of estimating the item parameters, we are interested in the population of examinees not in this sample. We have already encountered this assumption once in MML, where the random effect of the ability parameters in the context of item parameter estimation was given as a justification for integrating them out of the estimation process. In the language of Bayesian estimation, we are assuming that examinees are exchangeable (Novick, Lewis, & Jackson, 1973). Similarly to Bock and Lieberman (1970), Swaminathan and Gifford (1985) suggest using a standard normal distribution as the prior for the θ parameters.

Swaminathan and Gifford (1985) also suggest specifying a normal distribution for the b prior. However, unlike θ, Swaminathan and Gifford allow the mean (μ_b) and the standard deviation (σ_b) of the b prior to vary. Note that this is only permissible if μ_θ and σ_θ have been specified, otherwise the model is unidentified (Swaminathan & Gifford, 1985, p. 352). An alternative approach to identifying the model would be to specify μ_b and σ_b and allow μ_θ and σ_θ to vary. Although Swaminathan and Gifford do not specify the hyper-parameters of the b prior in their model, they do make the following three assumptions. First, they assume that μ_b and σ_b are independent; second, they assume that μ_b follows a uniform distribution; and third, they assume that σ_b follows an inverse chi-squared distribution (Novick & Jackson, 1970, p. 109). Finally, Swaminathan and Gifford suggest using a chi-squared prior for a. They based

this suggestion on the observation that a has the form of the reciprocal of the variance when (1) is used as a model for the probability of correctly answering a item. Since an inverse chi-square distribution is often used as the prior for the variance in a normal model, Swaminathan and Gifford reasoned that a chi-square distribution would work well for a.

It should be noted that Swaminathan and Gifford's algorithm is similar to JML (Birnbaum, 1968) in that the θ parameters are estimated simultaneously with a and b. Mislevy (1986) proposed a Bayesian equivalent of MML, in which the θ prior distribution is integrated over, removing θ from the estimation. The item parameters are then estimated conditional on the θ prior distribution, rather than on the θ values of the observed sample. Although this is not mathematically necessary, as it is in Maximum Likelihood, Mislevy's approach is more consistent with the theory of IRT laid out earlier in this section (i.e., that the items are independent of the examinees).

The premise of Mislevy's (1986) algorithm is to use the EM algorithm (Dempster *et al.*, 1977) to remove θ from the estimation by treating the θ parameters as missing data and integrating across the prior distribution. Once θ has been removed, a and b are estimated using the marginal log-likelihood function and their prior distributions. As before, we assume that θ is a random variable that follows a probability density $p(\theta|\boldsymbol{\tau})$, where $\boldsymbol{\tau}$ is a vector containing the examinee population parameters (e.g., μ_θ and σ_θ for the normal distribution described earlier). Unlike Swaminathan and Gifford (1985), who fixed the examinee population parameters, Mislevy (1986) allows the examinee population parameters to vary as a density function $p(\boldsymbol{\tau})$. Similarly, we assume that the vector of item parameters $\boldsymbol{\xi}_j = (a_j, b_j)^T$ follows a probability density function $p\left(\boldsymbol{\xi}_j|\boldsymbol{\eta}\right)$, where $\boldsymbol{\eta}$ is a vector of item population parameters that follow a density $p(\boldsymbol{\eta})$. If we assume that the items and examinees are independent, the joint

probability function of all of the item and ability parameters is

$$p\left(\boldsymbol{\theta}, \boldsymbol{\Xi}, \boldsymbol{\tau}, \boldsymbol{\eta}\right) = \prod_i p\left(\theta_i | \boldsymbol{\tau}\right) \prod_j p\left(\boldsymbol{\xi}_j | \boldsymbol{\eta}\right) p\left(\boldsymbol{\tau}\right) p\left(\boldsymbol{\eta}\right) \tag{55}$$

where $\boldsymbol{\Xi}$ is an $M \times 2$ matrix of item parameters. By Bayes' Theorem (DeGroot & Schervish, 2002, p. 66), the posterior density of $\boldsymbol{\theta}$, $\boldsymbol{\Xi}$, $\boldsymbol{\tau}$, and $\boldsymbol{\eta}$ is

$$p\left(\boldsymbol{\theta}, \boldsymbol{\Xi}, \boldsymbol{\tau}, \boldsymbol{\eta} | \boldsymbol{Y}\right) \propto L\left(\boldsymbol{Y} | \boldsymbol{\theta}, \boldsymbol{\Xi}\right) \cdot p\left(\boldsymbol{\theta}, \boldsymbol{\Xi}, \boldsymbol{\tau}, \boldsymbol{\eta}\right), \tag{56}$$

where $\boldsymbol{Y}$ is the $N \times M$ response matrix described earlier and $L\left(\boldsymbol{Y} | \boldsymbol{\theta}, \boldsymbol{\Xi}\right)$ is the likelihood function given by (12). Once the forms of the likelihood function and the various prior densities have been established, and the data have been observed, (56) contains all of the salient information about the parameters (Mislevy, 1986, p. 178).

The method proposed by Swaminathan and Gifford (1985) estimates the parameters from (56) by computing the joint posterior distribution of all of the unknown parameters and then using the mode of the distribution as point estimates. This results in a simulataneous estimation of both item and ability parameters analagous to JML (Birnbaum, 1968). To do this, Swaminathan and Gifford assume that the elements of $\boldsymbol{\xi}_j$ are independent so that the joint prior distribution $p\left(\boldsymbol{\xi}_j | \boldsymbol{\eta}\right)$ is simply the product of the marginal prior distributions of the elements of $\boldsymbol{\xi}_j$. After taking the natural logarithm of (56), Swaminathan and Gifford estimate the parameters by approximating solutions to

$$\frac{\partial \left\{\ell\left(\boldsymbol{Y} | \boldsymbol{\theta}, \boldsymbol{a}, \boldsymbol{b}\right) + \log\left[p\left(\theta_i | \boldsymbol{\tau}\right)\right]\right\}}{\partial \theta_i} = 0, \tag{57}$$

$$\frac{\partial \left\{\ell\left(\boldsymbol{Y} | \boldsymbol{\theta}, \boldsymbol{a}, \boldsymbol{b}\right) + \log\left[p\left(\boldsymbol{\xi}_j | \boldsymbol{\eta}\right)\right]\right\}}{\partial \boldsymbol{\xi}_j} = 0, \tag{58}$$

using the Newton-Raphson algorithm described earlier.

Mislevy (1986, p. 179) notes that the posterior mean would better serve as an estimator for the parameters since "its value for any subset of the parameters is invariant with respect to marginalization of (56) ver any subset of the remaining variables". However, in many cases there is no closed form solution for the posterior mean, leading to the use of the posterior mode, which can be approximated numerically. The approximation of the posterior mean by the posterior mode can be improved by marginalizing (56) over a set of "nuisance" parameters, such as the (incidental) ability parameters (Mislevy, 1986). This observation led Mislevy to propose an alternative algorithm, similar to MML (Bock & Aitkin, 1981), in which the item parameters are estimated from

$$\begin{aligned} p\left(\boldsymbol{\Xi}, \boldsymbol{\tau}, \boldsymbol{\eta} | \boldsymbol{Y}\right) &= \int p\left(\boldsymbol{\theta}, \boldsymbol{\Xi}, \boldsymbol{\tau}, \boldsymbol{\eta} | \boldsymbol{Y}\right) d\theta \\ &\propto L\left(\boldsymbol{Y} | \boldsymbol{\Xi}, \boldsymbol{\tau}\right) \cdot p\left(\boldsymbol{\tau}\right) \cdot p\left(\boldsymbol{\Xi} | \boldsymbol{\eta}\right) \cdot p\left(\boldsymbol{\eta}\right) \end{aligned} \tag{59}$$

where $p\left(\boldsymbol{\Xi}|\boldsymbol{\eta}\right) = \prod_j p\left(\boldsymbol{\xi}_j|\boldsymbol{\eta}\right)$. Note that (56) can be separated into two parts such that the terms that depend on $\boldsymbol{\theta}$ are separated from those that do not,

$$p\left(\boldsymbol{\Xi}, \boldsymbol{\tau}, \boldsymbol{\eta} | \boldsymbol{Y}\right) = \left\{ \int L\left(\boldsymbol{Y} | \boldsymbol{\theta}, \boldsymbol{\Xi}\right) \cdot p\left(\boldsymbol{\theta} | \boldsymbol{\tau}\right) d\boldsymbol{\theta} \right\} \cdot \left\{ p\left(\boldsymbol{\tau}\right) \cdot p\left(\boldsymbol{\Xi} | \boldsymbol{\eta}\right) \cdot p\left(\boldsymbol{\eta}\right) \right\} \tag{60}$$

where $p\left(\boldsymbol{\theta}|\boldsymbol{\tau}\right)$ is defined in an analagous fashion to $p\left(\boldsymbol{\Xi}|\boldsymbol{\eta}\right)$. The first bracketed term is approximated using Gauss-Hermite quadrature as described in Section 2.1.2. The approximation is then used to estimate parameter values using the EM algorithm (Dempster *et al.*, 1977) in a manner analagous to the algorithm described by Bock and Aitkin (1981).

In this section and the two previous sections, we have described a number of algorithms for estimating item parameters for IRT models. Having done this, the next question to address is how large a sample of examinees is necessary to obtain accurate parameter estimates? This question has been investigated in the literature

on numerous occasions, and as would be expected, the answer depends on which algorithm is used to estimate the parameters. In the next section we briefly review the literature on the necessary sample size for IRT model estimation.

2.1.4 Necessary Sample Size to Fit IRT Models

In the previous three sections, we described several algorithms for estimating the parameters of an IRT model, as well as some of the problems attendant to this process. One central question left unaddressed in these sections was how many examinees are necessary for accurate parameter estimation? This question has been asked since Birnbaum (1968) formalized JML, leading to a considerable body of research. An early example of this issue arising in practical research is the difficulty Lord (1968) had in estimating the parameters of the 2PL for the Verbal SAT as a means of demonstrating JML as a practical estimation method. In the appendix to his study, Lord (1968, p. 1016) notes that "unless $[M] > 50$, perhaps, and $N > 1,000$" both $\hat{a}$ and $\hat{b}$ had very high standard errors. Lord did not set out to investigate the sample size question and does not make any recommendations about sample size based on his results. However, subsequent authors (e.g., Baker & Kim, 2004; Hulin, Lissak, & Drasgow, 1982) have dubbed his observations the "Lord Heuristic": To estimate the parameters for a test of approximately 50 items requires a sample of approximately 1,000 examinees. This heuristic is still widely cited today (Gao & Chen, 2005; Harwell, Baker, & Zwarts, 1988; Hulin, Lissak & Drasgow, 1982).

Lord's observations spurred further research into the effects of sample size on IRT parameter estimates. Hulin *et al.* (1982) investigated the sample size question using a Monte Carlo simulation (Harwell, Stone, Hsu, & Kirisci, 1996) in which they varied sample size and test length in order to study their effects on JML parameter estimates. Like Lord (1969), Hulin *et al.* estimated the parameters for the 2PL using the computer program LOGIST (Wood & Lord, 1976; Wood, Wingersky, & Lord,

1976). They simulated samples of 200, 500, 1,000 and 2,000 examinees taking tests of 15, 30 and 60 items. To simulate the responses, they randomly sampled true values of the ability parameters from the standard normal distribution. The item parameters were both sampled from uniform distributions. The item difficulty parameters were randomly sampled from a uniform distribution on the interval $[-3, +3]$, and the item discrimination parameters were randomly sampled from a uniform distribution on the interval $[.3, 1.4]$, and then applying a 1.4 power transformation (i.e., $a_j = x_j^{1.4}$ where $x_j \sim U(.3, 1.4)$).

To assess the accuracy of their parameter estimates, Hulin *et al.* (1982) calculated the root mean square error (RMSE) of the estimated ICC and the true ICC at 31 points along the θ continuum. For the condition most comparable to the testing situation Lord (1968) described, Hulin *et al.* reported average RMSEs of 0.07 for the 200-examinee sample, 0.04 for the 500-examinee sample, 0.03 for the 1,000-examinee sample, and 0.02 for the 2,000-examinee sample. Their results show that the RMSE of the estimated ICC decreases with sample size. Furthermore, their results show that the average RMSE in all four sample size conditions decreased as the test length increases. For instance, for the 1,000 examinee sample described by Lord (1968), Hulin *et al.* found average RMSEs of 0.05, 0.04, and 0.03 for tests of 15, 30, and 60 items, respectively. It should be noted that both Lord (1968) and Hulin *et al.* (1982) estimated the parameters using JML. Consequently, their results can serve only as a rough guide to the effects of sample size on MML estimation.

More directly relevant results can be found in Swaminathan and Gifford's (1985) work on Bayesian IRT estimation. In order to demonstrate the Bayesian method described in the previous section, Swaminathan and Gifford (1985) conducted a series of Monte Carlo simulations in which they compared the performance of MML and Bayesian parameter estimates for a variety of test lengths and sample sizes. They made their comparison of the MML and Bayesian parameter estimates using a sim-

ilar procedure to Hulin *et al.* (1982). First they generated the true parameters by randomly sampling from pre-specified distributions. Specifically, they sampled θ_i and b_j from the standard normal distribution and sampling a_j from a uniform distribution on the interval $[0.6, 1.9]$. Using this approach Swaminathan and Gifford (1985) simulated 15, 25, and 35 item tests taken by 50, 100, 200, and 500 examinees. The item parameters for all 12 conditions were estimated using both MML and Bayesian estimation methods. In order to not overly favor the Bayesian estimation method, Swaminathan and Gifford used uniform priors for both θ and b, and a χ^2-distribution as the prior for a. They then computed the Mean Square Error (MSE) of the parameter estimates and the Pearson correlation between the parameter estimates and the true parameter values.

Swaminathan and Gifford's simulation results give some interesting insight into the interplay between sample size, test length, and estimation method. Like Hulin *et al.*, Swaminathan and Gifford's results show that the estimation error for both MML and Bayesian estimates decrease as both the test length and the sample size increased. However, their focus on the error of the individual estimates, rather than a global measure such as the RMSE of the ICC, gives us particular insights into which item parameter estimates are performing well. Broadly speaking, Swaminathan and Gifford's results show that the Bayesian estimates out perform the MML estimates in all conditions, consistently yielding smaller MSEs and higher Pearson correlations with the true parameter values. They also show that the b and θ estimates are more accurate than the a estimates. Finally, the difference between the accuracy of the MML and Bayesian estimates is especially pronounced for small groups of examinees taking short tests. For example, a sample of 50 examinees taking a 15 item test, had Bayesian estimate MSEs of b, a, and θ were 0.04, 0.04, and 0.13, respectively. The MML estimates had MSEs of 7.98, 14.52, and 1.26 for b, a, and θ, respectively, suggesting that using Bayesian estimation in testing situations with limited sample

sizes can dramatically increase the accuracy of the parameter estimates.

Yen (1987) performed a similar comparison of JML, as operationalized by BILOG (Mislevy & Bock, 1984), and MML, as operationalized by LOGIST (Wingersky & Lord, 1973). Based on the previous works of Lord (1968) and Hulin *et al.* (1982), Yen simulated responses for 1,000 examinees to 10-, 20-, and 40-item tests. She then estimated the item parameters from the 3PL model given by (2) using both BILOG and LOGIST. Her results showed that for the 10-item test JML yielded more accurate parameter estimates than MML, and for the 20- and 40-item tests the accuracy of the two methods were very similar. However, it should be noted that Yen (1987, p. 289) posits that BILOG may have had an advantage due the trait distributions used in the simulation. Specifically, some of her data sets had constant a and c parameter values, giving BILOG an artifactual advantage over LOGIST since the former program uses an empirical prior distribution to aid estimation (see Mislevy & Bock, 1984 for details).

Similarly, Harwell and Janosky (1991) compared the performance of Marginal Bayes Modal (MBM; Mislevy, 1986) estimation and MML estimation (Bock & Aitkin, 1981) in small samples. To do this Harwell and Janosky generated responses for 75, 100, 150, 250, 500, or 1000 examinees to either 15 or 25 items using the same distributions for generating θ, a, and b as Swaminathan and Gifford (1985). Their results demonstrated two points of interest. Firstly, like Swaminathan and Gifford (1985), Harwell and Janosky's results show that the Bayesian estimates out performed the MML estimates, yielding smaller RMSEs and higher correlations with the true parameters for all test lengths and sample sizes. However, unlike previous comparisons, Harwell and Janosky fit several Bayesian models that differed in the variance of the a prior distribution. The second interesting result of their study was that the accuracy of the parameter estimates improved as the variance of the prior drew closer to the true variance of the distribution that the discrimination parameters were sam-

pled from, and then decreased as the prior variance dropped below the true variance. Their prior variance conditions consisted of prior distributions with variances of $.75^2$, $.5^2$, $.25^2$, and $.1^2$. The RMSEs of the discrimination parameters were smallest when the prior variance was $.5^2$, which is closest to the true variance of ${}^{(1.9-.6)^2}/_{12} \approx .14$.

Gao and Chen (2005) compared the accuracy of MML (Bock & Aitkin, 1981) and MBM (Mislevy, 1987) parameter estimates by simulating responses for 100, 500, and 2000 examinees to tests of 10, 30, and 60 items using the 3PL (Birnbaum, 1968; Maxwell, 1959). They generated true values of a, b, and c by sampling either from a uniform distribution or from a 4-parameter beta distribution, depending on which condition they were simulating. In their first condition, all three parameters were drawn from uniform distributions, while in conditions 2-4 all of the parameters were drawn from 4-parameter beta distribution,

$$g(x) = \frac{1}{(u-l)^{\alpha+\beta-1} B(\alpha,\beta)} (x-l)^{\alpha-1} (u-x)^{\beta-1}, \; l < x < u \tag{61}$$

where u and l are the upper- and lower-bounds of the distribution, α and β are the two shape parameters of the beta distribution, and $B(\alpha,\beta)$ is the beta function. Their θ parameters were sampled from the standard normal distribution in all four conditions. Gao and Chen (2005) estimated parameters with the Estimation Toolkit for Item Response Models (ETRIM; Hanson, 2000) program modified to allow MML estimation. Their results, similar to those presented by Harwell and Janosky (1991), showed that across all conditions the MBM estimates performed better than the MML estimates in terms of the correlation between the estimate and the generating parameter and the average RMSE of the estimates. However, they also showed that the accuracy gains for using Bayesian estimation were more modest when the prior and the generating distributions were miss-matched.

The general thrust of these results is that MML item parameter estimates for

a 50 item test are accurate with approximately 500 examinees (Gao & Chen, 2005, Yen, 1987). Swaminathan and Gifford (1985), Harwell and Janosky (1991), and Gao and Chen (2005) have also found that Bayes Modal estimation, whether marginal or full, can accurately estimate IRT parameters with approximately half that number of examinees, if the prior is correctly specified. On the topic of specifying the prior, Baker and Kim (2004, p. 202) note that "[f]or the unwary, however, there are traps within the Bayesian approach. The specification of values of the hyper-parameters of the prior distributions requires an intimate understanding of both the test instrument and group of examinees." Clearly a test instrument under development or being used with a novel population presents difficulties for Bayesian estimation techniques. Baker and Kim (2004, p. 201) go on to suggest that, when appropriate, analysts can get preliminary parameter estimates on which to base a prior distribution using MML. However, they give no guidance on how to approach hyper-parameter or prior distribution selection when the prior is not known and MML is not appropriate.

What would be more useful in such situations is a method for estimating the parameters with the attractive qualities of Bayesian estimation, but without having to directly specify the prior distribution. One approach that has recently gained considerable attention in the statistical literature is penalized estimation (Hoerl & Kennard, 1970; Tibshirani, 1996; Zou & Hastie, 2005). Penalized estimation has already been applied to a number of models similar to the IRT model. We review the literature on this model in the next section, followed by an investigation of the few studies that have attempted to apply this method to IRT.

2.2 Regularized Logistic Regression

The problem of accurately estimating parameters from small samples of data is not unique to Psychometrics. One approach to small sample parameter estimation that has recently garnered considerable attention in statistics and computer science is

regularized estimation (Chen, Donoho, & Saunders, 1998; Hoerl & Kennard, 1970; Tibshirani, 1996; Zou & Hastie, 2005). Regularized estimation is similar to Bayesian estimation in that the objective function (i.e., the likelihood function) is augmented to provide additional information about the parameter values (Tibshirani, 1996). However, instead of specifying a Bayesian prior distribution, regularized estimation augments the objective function with a penalty function that constrains the parameter estimate to be close to a predetermined value, usually zero. The penalty prevents the parameter estimates from increasing or decreasing without bound (Hoerl & Kennard, 1970; Tibshirani, 1996; Zou & Hastie, 2005). Like the Bayesian prior distribution, the penalty function must be selected before fitting the model. However, regularized estimation has a mechanism for allowing the observed data to guide the penalty, reducing the subjectivity of its selection (Friedman, Hastie, & Tibshirani, 2010; Hastie, Tibshirani, & Friedman, 2009; Tibshirani, 1996; Zou & Hastie, 2005).

Regularized estimation has been applied to a wide variety of statistical models including linear regression (Hoerl & Kennard, 1970; Tibshirani, 1996; Zou & Hastie, 2005), generalized linear models (Tibshirani, 1996; Zou & Hastie, 2005), the Cox proportional hazards model (Tibshirani, 1997), support vector machines (Hastie, Rosset, Tibshirani, & Zhu, 2004), latent class analysis (Houseman, Coull, & Betensky, 2004), and linear discriminant analysis (Friedman, 1988). Here we focus on the logistic regression model (Tibshirani, 1996; Zou & Hastie, 2005) because of the close relationship between it and the 2PL IRT model (Baker & Kim, 2004; Paolino, 2013).

Logistic regression is a form of the generalized linear model (GLM) used when the response variable, Y , is binary indicating the "success" or "failure" of an event (Agresti, 2003, p. 165). For example, the response variable being observed might be whether a student answers a math problem correctly. If the student solves the problem and obtains the correct answer then the event is a success. If they do not solve the problem the event is a failure. We assume that each observation of the

response follows a Binomial distribution, $B(n_i, \pi_i)$, where n_i is the number of trials and π_i is the probability that a trial results in a successful outcome. If, as is the case in our example, only one trial is undertaken (i.e. $n_i = 1$) then the Binomial distribution simplifies to a Bernoulli distribution.

As with other regression models, the goal of logistic regression is to predict or explain the response by constructing a statistical model based on one or more predictor variables (Cohen, 1968). In logistic regression we do not model the response directly. Instead, we model the probability of success by using the logistic link function to connect a linear function of the predictor(s), X_j , to the success probability,

$$\pi(x) = \frac{\exp\left(\delta + \sum_{j=1}^{n} \beta_j x_j\right)}{1 + \exp\left(\delta + \sum_{j=1}^{n} \beta_j x_j\right)}, \tag{62}$$

where δ describes the location of the resulting logistic curve and β_j for $j = 1, \ldots, n$ describes the curve's instantaneous slope at δ . An equivalent form of this model is that the log-odds (the logit) of success is

$$\frac{\log(\pi(x))}{\log(1 - \pi(x))} = \delta + \sum_{j=1}^{n} \beta_j x_j. \tag{63}$$

In order to predict the probability of success, we must estimate the values of δ and β_j . The most common approach to this is maximum likelihood estimation in which the values $\hat{\delta}$ and $\hat{\beta}_j$ that maximize the likelihood function,

$$L(y_i|\alpha, \boldsymbol{\beta}, \boldsymbol{x}_i) = \prod_{i=1}^{N} \pi(\boldsymbol{x}_i)^{y_i} \left[1 - \pi(\boldsymbol{x}_i)\right]^{1-y_i}, \tag{64}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n)$ is a vector of slope parameters and $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{in})$ is the vector of observed predictors of the i^{th} observation (Agresti, 2003, p. 192). Finding $\hat{\delta}$ and $\hat{\beta}_j$ that maximize (64) by taking its first and second derivatives is computationally challenging (Agresti, 2003, p. 192). As a result, it is more common to find $\hat{\delta}$ and $\hat{\beta}_j$

that maximize the natural logarithm of (64),

$$\ell\left(y_i|\alpha,\boldsymbol{\beta},\boldsymbol{x}_i\right)=\sum_{i=1}^{N}y_i\log\left[\pi\left(\boldsymbol{x}_i\right)\right]+\left(1-y_i\right)\log\left[1-\pi\left(\boldsymbol{x}_i\right)\right], \tag{65}$$

which also avoids the round-off problem described earlier. Since the natural logarithm is a monotonic function of its argument the values of $\hat{\delta}$ and $\hat{\beta}_j$ that maximize (65) also maximize (64).

There are a number of methods for actually finding values of $\hat{\delta}$ and $\hat{\beta}_j$ that maximize (65). The most direct method is to set the first partial derivatives of (65) with respect to each of the $n+1$ model parameters equal to zero and to solve the resulting system of equations (Agresti, 2003, p. 193). In order to ensure that such a solution is indeed a maximum, it is also necessary to show that the matrix second partial derivatives of (65) (i.e., the Hessian) is negative definite. That is, for any non-zero vector $\boldsymbol{z}$, $\boldsymbol{z}^T\boldsymbol{H}\boldsymbol{z}$, where $\boldsymbol{H}$ is the Hessian matrix, is negative. Alternatively, numerical methods, such as the Newton-Raphson algorithm described earlier, may also be used to find parameter estimates that maximize Equation (65). Such methods are often advantageous because the system of equations that results from setting the first partial derivatives of Equation (65) equal to zero is inherently non-linear, and therefore may prove difficult to solve (Agresti, 2003, p. 194).

In order to regularize the parameter estimates, we subtract a penalty function from the function being maximized, in this case the log-likelihood function given by Equation (65). This results in the penalized log-likelihood function,

$$\ell_p\left(y_i|\boldsymbol{\beta},\boldsymbol{x}_i\right)=\sum_{i=1}^{N}y_i\log\left[\pi\left(\boldsymbol{x}_i\right)\right]+\left(1-y_i\right)\log\left[1-\pi\left(\boldsymbol{x}_i\right)\right]-P\left(\boldsymbol{\beta}|\lambda\right), \tag{66}$$

where $P\left(\boldsymbol{\beta}|\lambda\right)$ is the penalty function. Typically, the penalty function is a function of the coefficients associated with each variable and a tuning parameter, λ . Most authors who have contributed to this topic suggest scaling the predictor variables such

that $\delta = 0$, hence its omission from Equation (66) (Friedman, Hastie, & Tibshirani, 2010; Tibshirani, 1996; Zou & Hastie, 2005).

The mathematical form of the penalty function depends on the desired effect (Friedman, Hastie, & Tibshirani, 2010). The statistical literature has largely focused on three related penalty functions, the ridge penalty (Hoerl & Kennard, 1970; Nyquist, 1991), the Least Absolute Shrinkage and Selection Operator (LASSO; Tibshirani, 1996), and the elastic net penalty (Zou & Hastie, 2005). The ridge penalty (Hoerl & Kennard, 1970) penalizes the parameter estimates by a weighted sum of their squares[3],

$$P_R\left(\boldsymbol{\beta}|\lambda\right) = \lambda \sum_{j=1}^{n} \beta_j^2. \tag{67}$$

The effect of penalizing the log-likelihood with the ridge penalty is that, if two or more of the predictor variables are highly correlated, the associated parameter estimates are shrunken towards zero. However, it is unusual for the ridge penalty to shrink any parameters to zero (Tibshirani, 1996), so a ridge penalized model will retain all of the predictors. Instead, the ridge penalty averages the parameter estimates associated with high correlated groups of predictors, in essence allowing them to "borrow strength" from one another (Friedman, *et al.*, 2010, p. 3). In an effort to both shrink the parameter estimates and to select predictor variables for the model, Tibshirani (1996) introduced the LASSO penalty. The LASSO penalty is a weighted sum of the parameter estimates' absolute values[4],

$$P_L\left(\boldsymbol{\beta}|\lambda\right) = \lambda \sum_{j=1}^{n} \left|\beta_j\right|. \tag{68}$$

Unlike the ridge penalty, the LASSO penalty selects one of a group of highly correlated predictors, and shrinks the parameters associated with the other predictors to

[3]Alternatively, the ridge penalty is the weighted L_2 norm of $\boldsymbol{\beta}$ (Tibshirani, 1996).
[4]Alternatively the LASSO penalty is the weighted L_1 norm of $\boldsymbol{\beta}$ (Tibshirani, 1996).

zero, dropping them out of the model. This process is known as feature selection in the computer science literature (Chen *et al.*, 1998). With N observations the LASSO penalty admits at most N predictor variables into the model, even if $n > N$ (Friedman, Hastie, & Tibshirani, 2010, p. 5). This behavior of the LASSO penalty has been of great interest to researchers both because it allows to fit models with more parameters than observations, and because it allows for a form of variable selection.

In a sense, the ridge and LASSO penalties represent two ends of a spectrum of penalty functions that average the parameters of highly correlated predictors at one end and omit all but one of such predictors at the other (Friedman, *et al.*, 2010). In order to bridge the space between these two extremes Zou and Hastie (2005) proposed the elastic net penalty[5]. The elastic net penalty is a weighted combination of the ridge and LASSO penalties,

$$P_{EN}\left(\boldsymbol{\beta}|\lambda,\alpha\right) = \lambda\left[(1-\alpha)\sum_{j=1}^{n}\beta_j^2 + \alpha\sum_{j=1}^{n}|\beta_j|\right], \tag{69}$$

where $\alpha \in [0,1]$ is a second tuning parameter governing the relative influence of the ridge and LASSO penalty functions on the objective function (Friedman *et al.*, 2010; Zou & Hastie, 2005). When α is zero, the elastic net penalty simplifies to the ridge penalty, and averages the parameter estimates associated with highly correlated predictors. When α is one, the elastic net penalty simplifies to the LASSO penalty, dropping predictor variables from the model by setting their associated parameters to zero (Friedman, Hastie, & Tibshirani, 2010, p. 5).

Fitting the penalized model using either the LASSO or elastic net penalties (with $\alpha > 0$) is more difficult than fitting the un-penalized model (Friedman, Hastie, & Tibshirani, 2010). This is because the first partial derivatives of Equation (66) do

[5] Zou and Hastie (2005) introduce several variants of the elastic net penalty. The variant described here is the one most frequently discussed in the literature (e.g. Friedman, Hastie, & Tibshirani, 2010), the naive elastic net.

not exist when $\beta_j = 0$. As a result, Equation (66) cannot be maximized by solving the system of likelihood equations or by numerically approximating such a solution. However, several methods of maximizing the penalized log-likelihood function have been suggested in the literature (e.g. Efron, Hastie, Johnstone, & Tibshirani, 2004; Friedman, Hastie, Hoefling, & Tibshirani, 2007; Friedman, Hastie, & Tibshirani, 2010). Broadly speaking, the optimization methods used in penalized estimation problems fall into one of three categories: methods that exploit the piece-wise linearity of the parameter profiles (e.g. Efron *et al.*, 2004), coordinate descent methods (e.g. Friedman, *et al.*, 2010), and quasi-Newton methods (e.g. Broyden, 1970). Of these three methods, the most computationally efficient, and most widely used, are the coordinate descent methods (Friedman, Hastie, & Tibshirani, 2010). Coordinate descent methods for penalized estimation have been implemented in the R package `glmnet` (Friedman, Hastie, & Tibshirani, 2010). The piece-wise linear methods described by Efron *et al.* (2004) have been implemented in the R package `penalized` (Goeman, 2012). A detailed review of these methods is beyond the scope of our work, and interested readers are recommended to Friedman, Hastie, and Tibshirani (2010) for a recent overview of these methods.

However, in order to optimize the objective function, we must first select the tuning parameters λ and α. Generally speaking, α is selected subjectively based on the effect desired (Friedman, *et al.*, 2010). Since α is bounded between zero and one, its interpretation is relatively clear and, if in doubt, several values of α might be tried and their results assessed. Like α, the λ tuning parameter is bounded below by zero, which yields the same solution as optimizing the un-penalized log-likelihood function, (Friedman *et al.*, 2010; Tibshirani, 1996; Zou & Hastie, 2005). Unlike α, λ has no natural upper bound. Conceptually, a higher λ results in the penalty having a greater influence and, consequently, a greater degree of parameter shrinkage. With a sufficiently high λ, all of the parameter estimates will be shrunk to zero, resulting

in an intercept-only model (Friedman *et al.*, 2009). Since any λ higher than this will result in the same solution, this can be thought of as the upper bound of λ. However, the actual value of this $\lambda_{\max}$ depends on the model being fit (Friedman *et al.*, 2010).

Ideally the value of λ should not be chosen subjectively, otherwise regularized estimation has little philosophical advantage over Bayesian estimation. Rather, there should be a data-driven process for selecting λ based on the resulting model having met some criteria. Several approaches to selecting λ based on the data have been proposed (Friedman *et al.*, 2010; Houseman *et al.*, 2007; Tibshirani, 1996; Tutz & Schauberger, 2015; Zou, Hastie, & Tibshirani, 2007). Of these approaches, the one most widely implemented is the K-fold cross-validation approach proposed by Tibshirani (1996) and expanded on by Friedman *et al.* (2010). This method selects λ to minimize the model's cross-validated prediction error, resulting in a model that yields good out-of-sample predictions. To select λ by K-fold cross-validation, we split the data set into K subsets, or folds, in such a way that no observation appears in more than one fold. We then select a set of candidate values for λ for which to fit the model. Friedman, *et al.* (2010, p. 7) show that the smallest value of λ for which all of the parameter estimates are zero in a GLM model is

$$\lambda_{\max} = \frac{1}{N\alpha} \max_{k \subset n} |\langle \boldsymbol{x}_k, \boldsymbol{y} \rangle| \tag{70}$$

where $\langle \boldsymbol{x}_k, \boldsymbol{y} \rangle$ is the inner product of $\boldsymbol{x}_k$ and $\boldsymbol{y}$, and k denotes the sub-set of the predictors without the predictor whose parameter is being estimated. For each candidate λ between 0 and $\lambda_{\max}$ the penalized log-likelihood function is maximized using $K-1$ of the folds. The estimation error is then assessed using the K^{th} fold by comparing the predicted responses to the observed responses. In the context of logistic regression,

prediction error is calculated as the binomial deviance (Friedman, *et al.*, 2010),

$$D = 2\sum_{i=1}^{n} y_i \log\left(\frac{y_i}{\hat{y}_i}\right) + (1 - y_i)\log\left(\frac{1 - y_i}{1 - \hat{y}_i}\right), \tag{71}$$

where $\hat{y}_i = N\hat{\pi}_i$ for $\hat{\pi}_i$ given by Equation (62) with $\hat{\delta}$ and $\hat{\beta}_j$ substituted for δ and β_j. Typically, λ is selected to minimizes the binomial deviance (Tibshirani, 1996). Recently Friedman *et al.* (2010, p. 7) have suggested selecting the largest λ with a binomial deviance within one standard deviation of the λ that minimizes the binomial deviance. This results in a model with fewer parameters than the one resulting from following Tibshirani's (1996) suggestion, but still provides robust out-of-sample predictions (Friedman *et al.*, 2010).

The other general class of λ selection methods chooses λ to maximize the fit of the model (Houseman *et al.*, 2007; Tutz & Schauberger, 2015; Zou *et al.*, 2007). Broadly the procedure is the same as described above. A set of candidate λ values is selected and the regularized model is fit for each λ. The model fit is then assessed (rather than the estimation error or binomial deviance), and the λ that results in the best model fit is chosen. Previous literature has assessed model fit either by the Akaike Information Criterion (AIC; Akaike, 1974),

$$\text{AIC} = 2p - 2\ell, \tag{72}$$

or the Bayesian Information Criterion (BIC; Schwarz, 1978),

$$\text{BIC} = d.f. - 2\ell. \tag{73}$$

In the AIC, p is the number of parameters in the model and ℓ is the model's log-likelihood. The idea behind including this term is to favor more parsimonious models. In the BIC this term is replaced by an estimate of the model's degrees of freedom

($d.f.$). Zou *et al.* (2007) showed that the number of non-zero parameter estimates is an unbiased estimator of the BIC degrees of freedom for the LASSO model. Similarly, Hoerl and Kennard (1970) showed that the degrees of freedom for the ridge penalty is

$$d.f.(\lambda) = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda}, \tag{74}$$

where λ is the tuning parameter and d_j is the j^{th} singular value of the predictor correlation matrix. To date, no unbiased estimator for the elastic net penalty has been derived (Zou *et al.*, 2007, p. 2191).

It is important to note that there is a close relationship between regularized estimation and Bayesian estimation. Results identical to the ridge and LASSO penalties can be achieved using Bayesian estimation (Friedman, *et al.*, 2010; Tibshirani, 1996). Specifically, a result identical to that of the ridge penalty can be achieved by using a Gaussian prior distribution with mean zero and a standard deviation of $1/\lambda$ on the parameters. Likewise, an effect equivalent to the LASSO penalty can be achieved by imposing a Laplace prior, also with mean zero and standard deviation $1/\lambda$ on the parameters (Friedman, *et al.*, 2010, p. 3). This further demonstrates why a larger λ tuning parameter results in a more penalized model. Since λ is inversely proportional to the Bayesian prior's variance, increasing λ results in a stronger prior (i.e. a prior with lower variance). The advantage regularized estimation has is that one need not specify λ a priori, as is necessary for the variance of the Bayesian prior, but can allow the data to guide the selection of λ.

In addition to the statistical models already mentioned, regularized estimation has been applied to IRT in a limited way (Houseman, *et al.*, 2007; Paolino, 2013; Tutz & Schauberger, 2015). In the next section, we examine these studies and discuss their implications for the current work.

2.3 Regularized Estimation of IRT Models

Given regularized estimation's ability to reduce the parameter estimates' variance and to yield more reliable estimates from small samples, it is unsurprising that several researchers have already applied it to IRT (Houseman, Marsit, Karagas, & Ryan, 2007; Paolino, 2013; Tutz & Schauberger, 2015). The goals of and models implemented by these researchers are varied, and do not all correspond to the our goals. For example, Houseman *et al.* (2007) used regularized IRT to identify epigenetic predictors of bladder cancer in male patients. Their goal was to apply regularized IRT to answer a substantive question in their field. Consequently, Houseman *et al.* (2007) devote little energy to examining the methodological implications of their model. Likewise, Tutz and Schauberger (2015) applied regularized estimation to identifying differential item functioning (DIF; Zumbo, 1999; 2007) using the Rasch model (Rasch, 1960). Indeed, Tutz and Schauberger do not regularize the item parameters at all. They regularize the parameters associated with a set of group membership variables added to the Rasch model for the purpose of identifying DIF. To date, only Paolino (2013) has conducted a simulation study investigating the methodological implication of regularizing the item parameters in the 2PL. However, there are good reasons to be circumspect about his results. In this section we review these studies in detail, outlining both the similarities and differences with our proposed method, and describing what implications their results have.

Although Houseman *et al.*'s (2007) primary focus is substantive, they make two substantial methodological contributions to regularized IRT. The first is a modified version of the ridge penalty that acts as a data-drive model selection device. Item selection, rather than model selection, is our primary goal in regularizing the IRT model, and our approach is more akin to the approach taken by Paolino (2013) described below. However, Houseman *et al.*'s application of regularized IRT is an interesting alternative use of this technology. Recall that the ridge penalty described by Hoerl and

Kennard (1970) is the sum of the L_2 norms of the parameter vector. Subtracting this penalty function from the log-likelihood function effectively constrains the parameter estimates, preventing them from increasing or decreasing without bound (Friedman *et al.*, 2010; Hoerl & Kennard, 1970; Tibshirani, 1996). Another way of looking at this is that we are shrinking the parameter estimates towards zero (Friedman *et al.*, 2010). For the purpose of model selection in IRT, we do not want to shrink the discrimination parameter estimates ($\hat{\beta}_j$) towards zero, which would imply that the item does not measure θ. Instead, our aim would be to shrink $\hat{\beta}_j$ towards a common slope parameter, $\hat{\beta}_0$. This was the approach taken by Houseman *et al.* (2007) as a data-driven method of selecting either the 2PL or 1PL.

Houseman *et al.* (2007) demonstrated this model by analyzing a data set consisting of epigenetic, demographic, and medical data for 1,200 men from New Hampshire who had been screened for bladder cancer as part of an earlier study (Karagas, Tosteson, Morris, Demidenko, Mott, Heany, & Schned, 2004; Marsit, Karagas, Andrew, Liu, Danaee, Schned, Nelso, & Kelsey, 2005). As a point of comparison for their results, Houseman *et al.* (2007) also estimated their model using Bayesian estimation with a normal prior on the intercepts and a gamma prior on the slopes.Their results showed that the Bayesian and regularized models yielded very similar parameter estimates, though it should be noted that Houseman *et al.*'s priors are diffuse and their sample is relatively large, so this is unsurprising. Additionally, since the data used to demonstrate the method is from a real sample there is no way to know how accurate their parameter estimates are.

The other methodological contribution made by Houseman *et al.* (2007) is a method of selecting λ using the Akaike Information Criteria (AIC; Akaike, 1974). This is similar to the cross-validation λ selection method proposed by Tibshirani (1996; see also Friedman *et al.*, 2010). Prior to fitting the model, a sequence of candidate λ values is chosen. The model is then fit for each candidate λ value and the AIC for each of the

resulting model fits is calculated. Finally, λ is selected to provide the best model fit. To demonstrate their method, Houseman *et al.* (2007) conducted a simulation study comparing AIC-based λ selection to Mean Square Error (MSE)-based λ selection. The generated data for samples of 250 and 500 examinees responding to 25 items. For each simulation they set the intercepts of the 25 items to a sequence of values from -1.38 to 1.5 in intervals of 0.12 and generated the slopes of the 25 items from a normal distribution with mean 1.0 and variance $\sigma^2 \in \{0.25, 0.5, 1.0\}$[6]. For each of the six resulting conditions (2 sample sizes by three slope variances) Houseman *et al.* generated 1000 sets of binary responses using a Monte Carlo simulation (Harwell *et al.*, 1996). They then fit their model to the simulated data using both the AIC-base λ selection method and λselection method that sought to minimize the MSE of the slope parameter estimates,

$$\sum_{j=1}^{M} \left(\beta_j - \hat{\beta}_j\right)^2. \tag{75}$$

They fit each model for $\lambda \in \{0.001, 0.1, 0.5, 1, 5, 10\}$.

The result of Houseman *et al.*'s simulation showed that the MSE of the slope estimates tended to be minimized for higher values of λ when the variance of the slope parameters was low, at moderate λ values when the variance of the slope parameters was moderate, and at low λ values when the slope parameter variance was high. This finding is not surprising. Recall that in the previous section we noted that the effects of using the ridge penalty to regularize the parameter estimates could be achieved by using a normal Bayesian prior with mean zero and variance $1/\lambda$, which is, in effect, what Houseman *et al.*'s first simulation amounts to. Far more interesting is the fact that both AIC and MSE were minimized by using the same value of λ for the majority of the simulation runs. Houseman *et al.* (2007, p. 1273) interpret this finding as support for their hypothesis that selecting λ based on model fit is equivalent to selecting λ to minimize the estimation error, such as is the goal in the

[6]Houseman *et al.* (2007) do not describe how they generate θ in this simulation.

cross-validation approach (Friedman *et al.*, 2010; Tibshirani, 1996).

More recently, Tutz and Schauberger (2015) used regularized IRT to identify DIF (Zumbo, 1999; 2007) using the Rasch model (Rasch, 1960). DIF occurs when the probability of correctly answering an item for equally able examinees differs due to membership in some sub-set of the population, such as gender or ethnic groups (Zumbo, 1999). Typically DIF is seen as undesirable by test designers, but detecting differentially functioning items can be difficult in practice (Tutz & Schauberger, 2015, p. 21). Tutz and Schauberger's approach used an augmented IRT model that included a vector of observed group-membership indicators. When fitting this model, they used the LASSO penalty (Tibshirani, 1996) to shrink the parameter estimates associated with the indicators so that these parameters tended to zero if membership in the associated group did not change the probability of correctly answering the item. Like Houseman *et al.* (2007), Tutz and Schauberger's application of regularized IRT is to achieve a fundamentally different outcome than ours, where the aim is to identify items that do not measure θ well. However, like Houseman *et al.* (2007), Tutz and Schauberger propose a method of selecting λ based on model-data fit. In their case, model fit is quantified by the BIC (Schwartz, 1978), but otherwise the approach is similar to the one proposed by Houseman *et al.* (2007): the model is fit for a set of candidate λ values and the candidate λ value producing the smallest BIC (best fit) is chosen.

To test their model, Tutz and Schauberger simulated data from five different conditions that varied by sample size, test length, the number of differentially functioning items, and the strength of the DIF. In their first condition, they simulated responses for 250 examinees to a 20-item test, four of which functioned differentially on five of their sub-groups. Their second condition was the same, except that the sample size was 500 examinees. Their third and fourth conditions also consisted of 500 examinee samples taking tests of 20 and 40 items, respectively, with 8 differentially functioning

items. Finally, in their fifth condition, Tutz and Schauberger simulated the examinees' latent traits so that they correlated with the first observed variable (i.e., ability differed by sub-group) in order to test how their model fared when the items were functioning correctly, but true differences existed between sub-groups. Although the DIF related results of these simulations are not relevant in the current context, Tutz and Schauberger's simulations showed that the regularized IRT item parameter estimates had significantly less error (as measured by the MSE of the estimates) than MML item parameter estimates using the same data set. However, the errors of the latent trait parameter estimates were not effected by the penalty function.

Overall, Houseman *et al.* (2007) and Tutz and Schauberger's (2015) results are very encouraging for regularized IRT. Houseman *et al.* (2007) showed that regularized IRT can be applied to real-world data sets and that the regularized IRT item parameter estimates out-performed Bayesian estimates in terms of their error. Tutz and Schauberger (2015) showed that even when the item parameters themselves are not penalized, penalizing other parameters in the model can lead to more accurate estimates. However, as noted earlier, both studies differ from the current study in both objectives and methodology. Both Houseman *et al.* (2007) and Tutz and Schauberger (2015) used IRT models augmented with observable data, Houseman *et al.* to identify the latent variable model and Tutz and Schauberger to assess DIF. Both also used restricted versions of the elastic net penalty, which would allow for greater flexibility in how the parameter estimates are penalized. Finally, neither Houseman *et al.* nor Tutz and Schauberger's models were fit with the aim of selecting high performing items, as is the goal of our work. Houseman *et al.*'s model was closer in methodology to the model we present in the next section than that of Tutz and Schauberger because the penalty was applied across items, rather than across observable variables within an item. However, the penalty's purpose in Houseman *et al.*'s model was substantively different from its purpose in our model.

The aim of penalizing the item parameters in RMML is to remove poorly performing items (i.e., items with low discrimination parameters) from the estimation process, leaving more data available to estimate the parameters of the discriminating items. This idea has previously been investigated by Paolino (2013) using a slightly different approach. Paolino applied regularized estimation to the 2PL IRT model without any of the modifications used by Houseman *et al.* (2007) and Tutz and Schauberger (2015) to investigate whether regularized estimation could reduce the sample size needed to estimate the item parameters without compromising the estimates' accuracy. In order to do this, Paolino (2013) developed a variant of Birnbaum's (1968) JML algorithm in which the difficulty, discrimination, and latent trait parameter estimates are penalized, which he dubs Penalized JML (PJML).

Like the JML algorithm (Birnbaum, 1968), Paolino's (2013) PJML algorithm consists of two stages. In the first stage, Paolino estimates the discrimination parameters using the latent trait estimates obtained from the previous cycle as a faux data set. To constrain the estimates, Paolino applies the LASSO penalty (Tibshirani, 1996) to all of the estimates during the first stage. The purpose behind using the LASSO penalty with the discrimination parameter estimates is so that items with low discrimination parameters will be estimated with $\hat{a}_j = 0$. In the second stage, Paolino estimates the difficulty parameters and latent trait parameters with the ridge penalty (Hoerl & Kennard, 1970), using the discrimination parameter estimates obtained in the first stage as a faux data set. In both stages, Paolino used cyclical coordinate descent (van der Kooij, 2007) as operationalized in the `penalized` package (Friedman *et al.*, 2010) in R to estimate the parameters. Cyclical coordinate descent is a method for estimating parameters by optimizing non-differentiable multivariate objective functions (van der Kooij, 2007), such as a log-likelihood function penalized with the LASSO penalty function, that has been shown to be computationally efficient for generalized linear models (Friedman *et al.*, 2010). Without going into detail, cyclical coordinate

descent works by using the co-variance structure of the predictor variables to calculate coordinate-wise updates of the estimates.

There are several reasons to be circumspect regarding Paolino's algorithm. First, it is important to note that the stages Paolino described are quite different from those described by Birnbaum (1968). The JML algorithm's first stage estimated the latent trait parameters while treating the item parameter estimates as known quantities (Birnbaum, 1968). Then, in its second stage, the JML algorithm did the same to the latent trait estimates in order to estimate the item parameters (Birnbaum, 1968). This division of labor preserves the independence of the examinees' latent traits and the parameters describing the items used to measure them that we have assumed in order to write (12). The only parameters allowed to covary in the IRT model are the discrimination and difficulty parameters within a single item (Hambleton & Swaminathan, 1985, p. 129). By estimating the discrimination parameters and difficulty parameters separately, Paolino (2013) is ignoring this covariation. Furthermore, in estimating the ability and difficulty parameters in the manner he describes, Paolino violates the assumption that the examinees' abilities exist independently of the items. Indeed, his method appears to rely on there being a relationship between the latent trait and difficulty parameters in order to estimate them. This renders (12) invalid as the model's likelihood function.

The reason Paolino takes this strange approach to estimating the parameters because he relies on the cyclical coordinate descent algorithm to estimate the parameters with the non-differentiable LASSO penalty. Earlier, we noted that cyclical coordinate descent was intended for use with multivariate models (Friedman *et al.*, 2010; van der Kooij, 2007). Although IRT is a multivariate model in the sense that each item can be considered a variable, at the item-level the model is univariate. Specifically, we are regressing the vector of responses to an item onto the latent trait (de Ayala, 2002; Hambleton & Swaminathan, 1985). This makes the kind of coordinate-wise

updates described by Friedman *et al.* (2010) impossible without altering the model. Tutz and Schauberger (2015) also used cyclical coordinate descent to estimate their DIF detection model. However, they altered the IRT model by including observed covariates, and so did not face the same difficulty as Paolino (2013). Houseman *et al.* (2007) applied the ridge penalty in estimating the discrimination parameters. Since the ridge penalty is twice differentiable, they were able to estimate the parameters using a variation on the Newton-Raphson algorithm described previously.

Paolino (2013) does not include observed covariates, so the model he is attempting to estimate is univariate at the item level. In order to use cyclical coordinate descent as operationalized in the `penalized` package (Friedman *et al.*, 2010), Paolino stacks the columns of the response matrix, $\boldsymbol{Y}$, to form a vector of length $N \times M$. He then creates two "design" matrices, one for each step of his algorithm. The first design matrix is a block-diagonal matrix with blocks of size $N \times 1$ containing the current estimates of the latent trait parameters, as shown in Figure 4. This matrix is used to estimate the discrimination parameters. The second design matrix is an $NM \times (N + M)$ matrix as shown in Figure 5. Each row of this matrix contains two non-zero entries. The current discrimination parameter estimate appears in the column corresponding to the examinee whose latent trait parameter is being estimated (one of the first N columns), and a 1 appears in the column corresponding to the item whose difficulty parameter is being estimated (one of the last M columns).

There are several problems caused by Paolino's algorithm. The most concerning is the one discussed earlier: the algorithm appears to ignore the assumption of independence between items and examinees that is central to IRT. Additionally, by stacking the columns of the response matrix, Paolino violates the assumptions of logistic regression. As noted earlier, in logistic regression, we assume that the response vector consists of N independent realizations of a Bernoulli random variable with success probability π. However, the elements of the stacked columns of the response

$$
\begin{array}{cccccc}
y_{11} & \theta_1 & 0 & \cdot & \cdot & 0 \\
y_{21} & \theta_2 & 0 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
y_{N1} & \theta_N & 0 & \cdot & \cdot & \cdot \\
y_{12} & 0 & \theta_1 & \cdot & \cdot & \cdot \\
y_{22} & 0 & \theta_2 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
y_{N2} & 0 & \theta_N & \cdot & \cdot & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
y_{1n} & 0 & 0 & \cdot & \cdot & \theta_1 \\
y_{2n} & 0 & 0 & \cdot & \cdot & \theta_2 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
y_{Nn} & 0 & 0 & \cdot & \cdot & \theta_N
\end{array}
$$

Figure 4: Paolino's (2013) Stage One Design Matrix.

$$
\begin{array}{cccccccccc}
y_{11} & a_1 & 0 & \cdot & \cdot & 0 & 1 & 0 & \cdot & \cdot & 0 \\
y_{21} & 0 & a_1 & 0 & \cdot & \cdot & 1 & 0 & \cdot & \cdot & 0 \\
\cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
y_{N1} & 0 & 0 & \cdot & \cdot & a_1 & 1 & 0 & \cdot & \cdot & 0 \\
y_{12} & a_2 & 0 & \cdot & \cdot & 0 & 0 & 1 & \cdot & \cdot & 0 \\
y_{22} & 0 & a_2 & 0 & \cdot & \cdot & 0 & 1 & \cdot & \cdot & 0 \\
\cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\
y_{N2} & 0 & 0 & 0 & \cdot & a_2 & 0 & 1 & 0 & \cdot & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
y_{1M} & a_M & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 1 \\
y_{2M} & 0 & a_M & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 1 \\
\cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
y_{NM} & 0 & 0 & \cdot & \cdot & a_M & 0 & \cdot & \cdot & 0 & 1
\end{array}
$$

Figure 5: Paolino's (2013) Stage Two Design Matrix.

matrix are not independent, since every $N+1$ observation is originates from the same examinee. Finally, it is difficult to see the justification for penalizing the difficulty and ability parameters. As noted in an earlier section, the difficulty parameter is related to the threshold parameter in the linear form of the IRT model by the equation $\beta_j = -a_j b_j$ (Baker & Kim, 2004, p. 38). Thus, penalizing the difficulty parameter at least implies penalizing the location parameter, a practice that runs counter to the recommendations found in most of the literature (Friedman *et al.*, 2010; Hastie *et al.*, 2009, p. 125; Tibshirani, 1996). Likewise, in IRT, we assume that the examinees are independent (Hambleton & Swaminathan, 1985, p. 17). If different examinees' latent traits are independent, there is no co-variance structure between them inflating their coefficient (i.e., the slope parameters), and we gain nothing by penalizing them.

Our objections notwithstanding, Paolino's (2013) results are of interest. He compares PJML to MML (Bock & Aitkin, 1981; Bock & Lieberman, 1970) as operationalized in the R package `ltm` (Rizopoulos, 2006) and to Bayes Modal estimation (Swaminathan & Gifford, 1985) as operationalized in the `irtoys` (Partchev, 2012) and `ICL` (Hanson, 2002) packages for R (R Core Team, 2015). The comparison was made using simulated data generated from the 2PL model (Birnbaum, 1968; Maxwell, 1958). Paolino examined six combinations of test length and sample size: 200 examinees responding to 20 items, 300 examinees responding to 50 items, 400 examinees responding to 100 items, 20 examinees responding to 20 items, 50 examinees responding to 20 items, and 50 examinees responding to 100 items. In the first three conditions, PJML yielded a lower Root Mean Square Error (RMSE) than MML for the discrimination and difficulty parameter estimates, but Bayes Modal estimation had a lower RMSE than either of the other methods. In the other three conditions, the MML and Bayes Modal algorithms did not converge. However, PJML was able to produce parameter estimates. Although Paolino (2013, p. 40 - 41) reports RMSEs for these conditions for PJML, with nothing to compare them to it is difficult to use

them to assess performance.

Although previous research on penalized IRT estimation has shown some promise, it is clear that further work is needed in this area. In the next section, we present a new method for regularizing IRT parameter estimation, Regularized MML (RMML). RMML is an item selection approach, similar in concept to PJML. Unlike PJML, RMML is based on the MML algorithm described by Bock and Aitkin (1981) and presented in Section 2.1.2, and penalizes only the slope parameters in (13), since these are most directly related to an item's ability to measure the latent trait. In addition, we estimate parameters using a numerical approximation to the Newton-Raphson algorithm to avoid the problems incumbent with using cyclical coordinate descent with a univariate model. The details of this algorithm are given in the next section.

3 Regularized Marginal Maximum Likelihood Estimation

Although the results described in the previous section are promising, none are entirely satisfactory. The method proposed by Tutz and Schauberger (2015), while an interesting application of regularized estimation, does not address our central question, namely whether regularized estimation can compensate for small samples. Houseman *et al.* (2007) and Paolino (2013) both proposed methods that attempt to address this question. However, the method proposed by Houseman *et al.* (2007) cannot be applied to the normal ability testing situation without considerable modification, and the method proposed by Paolino (2013) violates several of the IRT model's assumptions, as detailed in the last section. Furthermore, neither study presents a clear picture of how regularized estimates compare to estimates from MML (Bock & Aitkin, 1981; Bock & Lieberman, 1970) or Bayesian estimation (Mislevy, 1986; Swaminathan & Gifford, 1985).

In this section, we propose a new regularized estimation algorithm for IRT based on the EM algorithm (Bock & Aitkin, 1981; Dempster *et al.*, 1977). Since our algorithm is founded on Bock and Aitkin's (1981) MML algorithm, we have dubbed it Regularized Marginal Maximum Likelihood (RMML). The aim of this algorithm is similar to Paolino's (2013) PJML algorithm. In order to estimate item parameters from small samples, we use penalty functions to identify poorly performing items by constraining their slopes to be close to zero. Items with zero slopes are dropped from the model, freeing up information to estimate the other, better performing items. The concept behind this approach will be familiar to anyone acquainted with the process of test development (Hambleton & Swaminathan, 1985, p. 225). During the test development process, an item set is written and administered to a representative sample from the target population. The items are then analyzed to determine how much information about the trait or ability being measured they yield. Items that yield little or no information are discarded or revised, and the process is repeated. Typically the decision to retain or discard an item is made after the parameters have been estimated (Hambleton & Swaminathan, 1985, p. 225). Our goal is to allow the model to make this decision during the estimation process.

To use regularized estimation we need to address two important questions. The first, and most obvious, is what are we penalizing and which penalty should we use? Since our goal is to identify and drop poorly performing items, penalizing the slope or discrimination parameters is a logical choice. Equation (6) shows that the information provided by an item is a function of the discrimination parameter, thus an item with low information must also discriminate poorly. Furthermore, it would appear that the best approach is to use either the LASSO penalty (Tibshirani, 1996) as Paolino (2013) did, or the elastic net penalty (Zou & Hastie, 2005). Either of these penalties can act as a selection operator, but for our purposes we favor the elastic net penalty. This is because item in an IRT model are not directly analogous to variables in a

regression model. The elastic net penalty allows us to examine a myriad of penalties between the extremes typified by the LASSO and ridge regression penalties, giving us the option to allow the data to guide the penalty selection (Friedman *et al.*, 2010).

The second question in need of an answer is how are we to optimize the penalized log-likelihood function? Recall that the LASSO and elastic net penalties are not generally differentiable (Tibshirani, 1996; Zou & Hastie, 2005). Therefore, the penalized log-likelihood function cannot be optimized using the Newton-Raphson algorithm recommended by Birnbaum (1968) and Bock and Aitkin (1981). Additionally, the IRT model under consideration is a unidimensional model (Birnbaum, 1968), in the sense that the items measure a single trait or ability. As a result, the cyclical coordinate descent algorithm (van der Kooij, 2007) recommended by Friedman *et al.* (2010) is also inadmissible. We address this issue in the next section, where we present the details of RMML.

3.1 Regularized Marginal Maximum Likelihood

Recall from Section 2.1.2 that the EM algorithm (Dempster *et al.*, 1977) consists of two steps. In the first step, the expectations of the missing data are computed based on the current parameter estimates and the assumed missing data distribution (i.e., the prior distribution). Then, in the second step, new parameter estimates are calculated conditional on the expected values of the missing data. In the IRT context, the missing data are the ability or trait values, θ, and the parameters to be estimated are the item parameters γ and β (Baker & Kim, 2004; Bock & Aitkin, 1981).

The E-step in RMML is essentially identical to the E-step for MML as described earlier. Like Bock and Aitkin (1981; see also Harwell *et al.* 1989), we assume that θ follows the standard normal distribution. Based on this assumption, we generate a Gauss-Hermite quadrature as described earlier and compute the expected number of examinees at each quadrature node and the expected number of correct answers to

each item from the examinees at each node using (42) and (43), respectively. This brings us to the M-step and the point of departure between MML as described by Bock and Aitkin (1981) and RMML.

In the M-step, our aim is to find values for γ and β that maximize the penalized log-likelihood function conditional on the expected values computed in the E-step. In order to allow the data to guide our selection of a penalty to the greatest extent possible, we impose the elastic net penalty (Zou & Hastie, 2005) on the item slope parameters. Recall that the elastic net penalty,

$$P_{\lambda,\alpha}\left(\hat{\beta}_j\right) = \lambda\left[\alpha\sum_{j=1}^{m}|\beta_j| + \frac{(1-\alpha)}{2}\sum_{j=1}^{m}\beta_j^2\right] \tag{76}$$

is a combination of the ridge penalty (Hoerl & Kennard, 1970) and the LASSO penalty (Tibshirani, 1995), where the first term inside the brackets is the LASSO penalty and the second term is the ridge penalty. The tuning parameters λ and α govern the strength of the penalty relative to the objective function and the relative strength of the penalty's components, respectively. In the special case of the IRT model, we penalize the log-likelihood function, resulting in

$$\begin{aligned}\ell_p\left(\boldsymbol{\gamma},\boldsymbol{\beta}|\boldsymbol{x},\bar{\boldsymbol{n}},\bar{\boldsymbol{R}},\lambda,\alpha\right) &= \sum_{k=1}^{K}\sum_{j=1}^{n}\bar{r}_{jk}\log\left[P_j\left(x_k\right)\right] + \left(\bar{n}_k - \bar{r}_{jk}\right)\log\left[Q_j\left(x_k\right)\right] \\ &\quad -\lambda\left[\alpha\sum_{j=1}^{m}|\beta_j| + \frac{(1-\alpha)}{2}\sum_{j=1}^{m}\beta_j^2\right].\end{aligned} \tag{77}$$

This brings us to the question of maximizing this function in order to obtain parameter estimates. As previously noted, the Newton-Raphson algorithm used by Bock and Aitkin (1981) does not fit our purpose for two reasons: the gradient of the penalized log-likelihood is degenerate when any of the $\beta_j = 0$, and for the penalty to properly affect all of the slope parameter estimates, they must be estimated simultaneously. Recall that in Bock and Aitkin's (1981) algorithm, the log-likelihood

is maximized separately for each item due to the structure of the Hessian. Simultaneous estimation using the Newton-Raphson algorithm requires the inversion of a potentially large Hessian matrix, a computationally daunting prospect that is prone to inaccuracies and round off errors.

There are any number of potential solutions to these problems. The solution implemented in the R (R Core Team, 2015) program in Appendix B uses the variable metric algorithm (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970), as operationalized in the R function `optim`. The variable metric algorithm is a quasi-Newton method that solves the problem of inverting the Hessian by approximating the inverse Hessian numerically. By combining this powerful optimization algorithm with box constraints (Byrd, Lu, Nocedal, & Zhu, 1995), it can also provide a work around for the discontinuity in (77). In other respects, the variable metric algorithm is very similar to the Newton-Raphson algorithm described earlier. This similarity is attractive as it allows us to stay true to Bock and Aitkin's (1981) MML algorithm.

The variable metric algorithm approximates the Hessian matrix by using iterative rank one updates specified by evaluations of the gradient. Let $\hat{\boldsymbol{\delta}}^{(k)}$ be the parameter estimates at the algorithm's k^{th} iteration. To calculate $\hat{\boldsymbol{\delta}}^{(k+1)}$, we perform a line search in the direction $\boldsymbol{p}^{(k)}$, which is found by solving the analogue Newton equation

$$\boldsymbol{B}^{(k)}\boldsymbol{p}^{(k)} = -\nabla f\left(\hat{\boldsymbol{\delta}}^{(k)}\right) \tag{78}$$

where $\boldsymbol{B}^{(k)}$ is an approximation to the Hessian matrix and $\nabla f\left(\cdot\right)$ is the gradient of the function $f\left(\cdot\right)$ evaluated at $\hat{\boldsymbol{\delta}}^{(k)}$. Instead of requiring the full Hessian matrix at the point $\hat{\boldsymbol{\delta}}^{(k+1)}$ to be computed, the approximated Hessian matrix at stage k is updated by the addition of two matrices,

$$\boldsymbol{B}^{(k+1)} = \boldsymbol{B}^{(k)} + \boldsymbol{U}^{(k)} + \boldsymbol{V}^{(k)}, \tag{79}$$

where both $\boldsymbol{U}^{(k)}$ and $\boldsymbol{V}^{(k)}$ are symmetric, rank-one matrices with different bases. The requirement that $\boldsymbol{U}^{(k)}$ and $\boldsymbol{V}^{(k)}$ be symmetric and rank-one assumption means that we may write

$$\boldsymbol{C} = \boldsymbol{a}\boldsymbol{b}^T \tag{80}$$

so that $\boldsymbol{U}^{(k)}$ and $\boldsymbol{V}^{(k)}$ construct a rank-two update matrix that is robust against the scale problem often observed in the gradient descent method (Shanno, 1970). The quasi-Newton condition imposed on this update is

$$\boldsymbol{B}^{(k+1)}\left(\hat{\boldsymbol{\delta}}^{(k+1)} - \hat{\boldsymbol{\delta}}^{(k)}\right) = \nabla f\left(\hat{\boldsymbol{\delta}}^{(k+1)}\right) - \nabla f\left(\hat{\boldsymbol{\delta}}^{(k)}\right). \tag{81}$$

From an initial guess $\boldsymbol{\beta}_0$ and an approximate Hessian $\boldsymbol{B}_0$, the following steps are repeated as $\boldsymbol{\beta}_k$ converges to a solution:

1. Obtain a search direction, $\boldsymbol{p}^{(k)}$, by solving $\boldsymbol{B}^{(k)}\boldsymbol{p}^{(k)} = -\nabla f\left(\hat{\boldsymbol{\delta}}^{(k)}\right)$.
2. Perform a line search to find an acceptable step size $\alpha^{(k)}$ in the direction $\boldsymbol{p}^{(k)}$. Once the step size has been found, update $\hat{\boldsymbol{\delta}}^{(k+1)} = \hat{\boldsymbol{\delta}}^{(k)} + \alpha^{(k)}\boldsymbol{p}^{(k)}$.
3. Set $\boldsymbol{s}^{(k)} = \alpha^{(k)}\boldsymbol{p}^{(k)}$.
4. Calculate $\boldsymbol{g}^{(k)} = \nabla f\left(\hat{\boldsymbol{\delta}}^{(k+1)}\right) - \nabla f\left(\hat{\boldsymbol{\delta}}^{(k)}\right)$.
5. Calculate $\boldsymbol{B}_{k+1} = \boldsymbol{B}_k + \frac{\boldsymbol{g}^{(k)}\boldsymbol{g}^{(k)T}}{\boldsymbol{g}^{(k)T}\boldsymbol{s}^{(k)}} - \frac{\boldsymbol{B}^{(k)}\boldsymbol{s}^{(k)}\boldsymbol{s}^{(k)T}\boldsymbol{B}^{(k)}}{\boldsymbol{s}^{(k)T}\boldsymbol{B}^{(k)}\boldsymbol{s}^{(k)}}$.

Practically $\boldsymbol{B}^{(0)}$ can be initialized with $\boldsymbol{B}^{(0)} = \boldsymbol{I}$, resulting in the first iteration of the algorithm being equivalent to the gradient descent method. Subsequent iterations are refined by $\boldsymbol{B}^{(k)}$. The gradient, $\nabla f(\cdot)$, can be approximated by the finite difference method (Stewart, 2004, p. 236).

The solution to the problem of the degenerate gradient when $\beta_j = 0$ hinges on our knowledge of the domain of β_j. Recall from our earlier discussion that $\beta_j = a_j$, the item discrimination parameter, and that a_j is defined on the interval $[0, +\infty)$.

Thus, while a_j may get arbitrarily close to 0, it theoretically will not reach zero. Therefore, to avoid the discontinuity in the gradient, we constrain the algorithm to consider solutions for β_j greater than zero. This type of constraint is known as a box constraint (Boyd & Vandenberghe, 2004; Byrd *et al.*, 1995). A version of the variant metric algorithm that incorporates box constraints (Byrd, Lu, Nocedal, & Zhu, 1995) is included in R's `optim` library, and is used in the code in Appendix B.

After the parameter estimates have been calculated, they are passed back to the E-Step, where the expected data is computed once again. The quadrature nodes and weights do not change iteration to iteration, nor do the observed responses. The algorithm continues to iterate between the E and M-Steps until a convergence criteria is reached. It should be noted that because the two-parameter logistic IRT model is not a member of the exponential family (Harwell *et al.* 1988) convergence of the EM algorithm is not guaranteed. Empirical work (e.g., Mislevy & Bock, 1985) suggests that the algorithm does generally converge, though a large number of iterations may be necessary, especially if the sample size is small.

We now turn to evaluating the effectiveness of the RMML algorithm. In the next section we detail the simulation used to investigate different aspects of RMML, focusing particularly on the effects that the tuning parameters λ and α have of the estimation error, and on the comparison between the estimation error of the RMML estimates and item parameter estimates obtained from other, established IRT estimation methods.

4 Methods

In this section we describe two Monte Carlo simulations aimed at investigating the performance of RMML. The first simulation is a naturalistic simulation in which the sample size, test length, and tuning parameters are varied. We hypothesize that

RMML will yield more accurate parameter estimates than MML, and will have similar results to Bayesian estimation, for small samples. Our second simulation is a factorial experiment investigating the relative performance of RMML, MML, and two Bayesian models when the test contains some items that do not measure the latent trait of interest, resulting in slope (or discrimination) parameters of zero. In this second simulation, we hold the sample size and test length constant, and vary the number of items with discrimination parameters equal to zero, the non-zero discrimination parameters, and the difficulty parameters.

In both simulations, we will consider four methods of estimating the item parameters: MML (Bock & Aitkin, 1981), RMML, and two Bayesian models (Swaminathan & Gifford, 1985). MML is the current standard for the maximum likelihood item parameter estimation (de Ayala, 2004; Hambleton & Swaminathan, 1985), whereas Bayesian IRT models are frequently used with small samples (Gao & Chen, 2005). The two Bayesian IRT models are distinguished by the discrimination parameters' prior distribution. The first model uses a log-normal distribution with a log mean of 0 and a log standard deviation of 0.25 for the discrimination prior, and the second model used a uniform distribution between 0.6 and 1.9. This latter distribution was used by Swaminathan and Gifford (1985) in their simulations and was recommended by Gao and Chen (2005) as an appropriate prior distribution for the discrimination parameters. Both models use a uniform distribution between -3 and 3 as the difficulty parameters' prior distribution. We fit the Bayesian models using Markov-Chain Monte Carlo (MCMC) as implemented in STAN (STAN Development Team, 2014) via the R script given in Appendix C. Specifically, STAN is a Gibbs sampler (Geman & Geman, 1984). To estimate the posterior distributions of the parameters, it iteratively samples values from the marginal distribution of each parameter, conditional on the current estimates of the remaining parameters. This sampling is done a large number of times to gain an understanding of the parameter's marginal distributions,

and then the parameters are estimated by taking a measure of central tendency (in our case the mean and median were used) of the distribution after S taking samples. Since samples early in the sampling process are unlikely to accurately reflect the distribution, all of the Bayesian model fits were allowed a burn in period of 1,000 iterations before starting sampling. The RMML model was fit using the algorithm in Appendix B, which was described in the previous section. The RMML model was fit for 101 combinations of the two tuning parameters λ and α, where λ consisted of a sequence from 0 to 2 in steps of 0.1 and α consisted of a sequence from 0 to 1 in steps of 0.25, in order to show how selection of the tuning parameters affects performance. The MML estimates were made using a custom written R program (Appendix C). Both RMML and MML were fit using 100 quadrature nodes.

The simulation results will be assessed in by examining the accuracy of the parameter estimates. First, the root mean square error (RMSE) of the discrimination and difficulty parameter estimates were computed separately. For a generic parameter γ, the RMSE of an estimate of γ, $\hat{\gamma}$, is,

$$RMSE_{\gamma} = \sqrt{\frac{\sum_{j=1}^{M} \left(\hat{\gamma}_j - \gamma_j\right)^2}{M}}, \tag{82}$$

where M is the number of parameters (in this case the number of items). Parameter estimates with lower RMSEs are closer to the true parameter value, and are therefore more accurate. The discrimination and difficulty RMSEs show the relative accuracy of their respective parameter estimates, but they do not give a good impression of overall test performance. For such a holistic assessment, we compute the root integrated mean square error (RIMSE; Ramsay, 1991),

$$\Delta_{P(j)} = \sqrt{\int_{-\infty}^{\infty} \left[P_j\left(\theta\right) - \hat{P}_j\left(\theta\right)\right]^2 g\left(\theta\right) d\theta}, \tag{83}$$

where $\Delta_{P(j)}$ denotes the RIMSE for the j^{th} item, g denotes the distribution of θ, P_j is the probability of a correct response based on (4), and $\hat{P}_j$ is the estimated probability of a correct response based on the parameter estimates $\hat{\gamma}$ and $\hat{\beta}$. Unlike the RMSE, which is bounded below by zero but unbounded above, the RIMSE is bounded above by one because the squared error terms are multiplied by the distribution of θ. This means that, in addition to being a more holistic view of the performance of the estimation algorithm, the RIMSE is somewhat easier to interpret.

Having covered all those facets common to both simulations, we now turn to the details of the simulations themselves, starting with the naturalistic achievement testing scenario.

4.1 Simulation 1

In the first simulation the test length, sample size, and tuning parameters were varied, and their effect on the RMML, MML, and Bayesian parameter estimates accuracy was evaluated. We used a Monte Carlo simulation (Harwell *et al.*, 1996) to generated responses from samples of 100, 200, 500, and 1,000 examinees to tests of 15, 25, and 35 items. First we generated item parameters for each test and latent trait parameters for each sample by randomly sampling values from appropriate distributions. For θ and b, we sampled values from the standard normal distribution. For a, we sampled values from a log-normal distribution with log mean of zero and a log standard deviation of 0.25 for the discrimination parameters. The observed distributions of the parameters for each test are shown in Figures 6 and 7, and are given in the table in Appendix D. Note that the distribution used to generate the discrimination parameters matches one of the two prior distributions used for the Bayesian models. This was done to provide a look at the performance of the Bayesian estimates under the best possible conditions.

After generating true parameter values, we compute the probability of each ex-

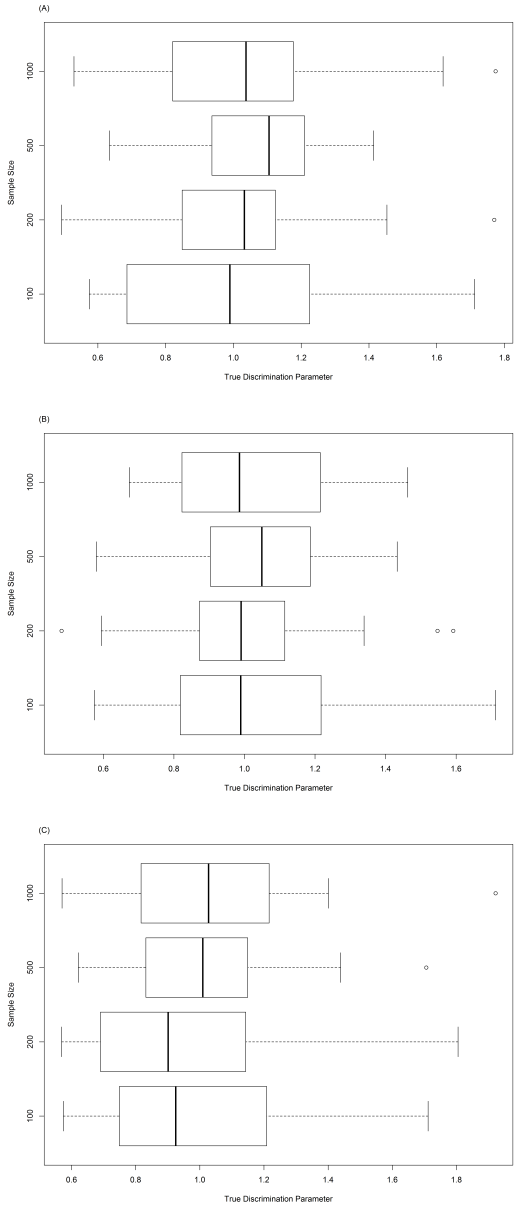

Figure 6: Distributions of Simulated Discrimination Parameter (a_j) values. Panel (A) shows the distribution of discrimination parameters (a_j) values simulated for the 15-item tests, Panel (B) shows the distributions for the 25-item tests, and Panel (C) shows the distributions for the 35-item tests.

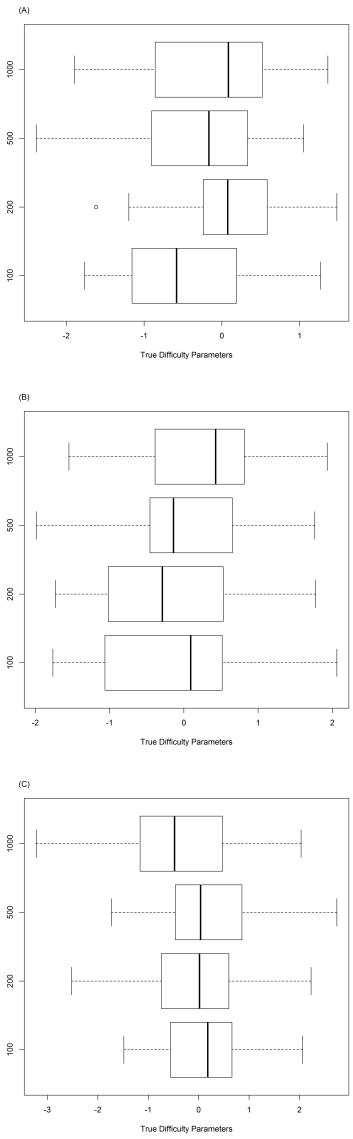

Figure 7: Distributions of simulated difficulty parameter values.
Panel (A) shows the distribution of difficulty parameters (b_j) values simulated for the 15-item tests, Panel (B) shows the distributions for the 25-item tests, and Panel (C) shows the distributions for the 35-item tests.

aminee correctly answering each item using (4). We then compare each probability to a realization of a uniform random variable on the span from 0 to 1. If the uniform random variable exceeds the calculated probability, the examinee answers the item incorrectly; otherwise, the examinee answers the item correctly. We record the responses as 0 if the response is incorrect, or 1 if the response is correct. This results in a binary response matrix simulated from the true parameter values. This process was repeated five times for each test, resulting in five unique response matrices for each combination of test length and sample size. Finally, we estimate the item parameters from each response matrix using MML, RMML, and Bayesian estimation as described above. This results in the estimated parameter values, whose accuracy we measure with the RMSE and RIMSE as described earlier.

Simulation 1 is designed to explore how RMML will perform in comparison to MML and Bayesian estimation in a realistic, achievement testing scenario. Although this information is vital for demonstrating the efficacy of the algorithm, it does not allow us to investigate certain uncommon testing situations in which RMML might have an advantage. Specifically, if a test contains items that do not measure the latent trait of interest, RMML can hypothetically ignore these items and yield better parameter estimates for items that do measure the latent trait. Items that do not measure the latent trait can be thought of as non-discriminating items, since they will have very low discrimination parameters. To explore this interesting scenario, we designed a second simulation that allowed us to include non-discriminating items in a more structured manner. This simulation is detailed in the next section.

4.2 Simulation 2

Our second simulation was a factorial design in which we compared the performance of MML, RMML, and Bayesian estimation in the presence of non-discriminating items (i.e., items for which $a_j = 0$). Such an item may have face validity (Crocker & Algina,

1986) (i.e., it appears to measure the desired trait), but does not measure the latent trait in practice. During the instrument development process, items that do not measure the latent trait are identified and either revised or removed (Hambleton & Swaminathan, 1985), thus identifying non-discriminating items is an important step in the test development process. As noted earlier, the goal of RMML is to shrink the discrimination parameter estimates in the hope of reducing their variance. Our hypothesis is that RMML will more readily identify non-discriminating items because the penalty naturally shrinks the discrimination parameter estimates towards zero. In contrast, MML does not shrink the estimates at all and the Bayesian models bias the estimates towards the centers of their priors.

As in the previous simulation, we simulated the response matrices using a Monte Carlo simulation (Harwell, *et al.*, 1996). However, rather than randomly sample the item parameters as in Simulation 1, we fixed the item parameters as factors in our design. This was done for two reasons. First, we wished to clarify some of the observations described in the next section regarding the relationship between the item parameters and the tuning parameters that resulted from Simulation 1. Fixing the item parameters allowed us to better understand how these factors interact. Second, if the difficulty parameters were randomly sampled, we would have no clear means of choosing which items were non-discriminating, and thus which difficulty parameters were paired with zero discrimination parameters. In other words, it is possible that a non-discriminating item with a difficulty close to the center of the latent trait distribution has a different effect than a non-discriminating item in either tail of the distribution. The factor levels used in Simulation 2 are shown in Table 2. In addition to varying the item parameters, each condition also contained a number of non-discriminating items. The number of non-discriminating items on the test was varied from 2 (10% of the items) to 10 (50% of the items).

The three factors shown in Table 2 were fully crossed for a total of 54 conditions,

Table 2: Simulation 2 Factors

Factor	Levels
Discrimination	0.5, 1.0, 1.5
Difficulty	-1.0, 0.0, 1.0
Number of Non-Discriminating Items	2, 4, 6, 8,10

each of which was replicated 5 times. The results from Simulation 1, described in the next section, suggest that RMML is only really helpful in very small samples. Based on this observation, Simulation 2 used a single sample of 100 examinees generated by randomly sampling latent trait parameter values from the standard normal distribution. This resulted in a total of $54 \times 5 = 270$ unique response matrices. Item parameters were estimated from each matrix using MML, RMML, and the two Bayesian models using the settings described above.

In the next section we examine the results of these two simulations, beginning with the naturalistic simulation followed by the factorial simulation.

5 Simulation Results

5.1 Simulation 1

Simulation 1 had two aims, to compare the performance of RMML to the performance of MML and Bayesian estimation with the kind of data commonly encountered in the achievement testing domain, and to investigate how varying the tuning parameters affects the accuracy of the parameter estimates. As described earlier, the performance of the estimation procedures was quantified in two ways. First, the RMSE of the discrimination and difficulty parameters were computed separately to evaluate the accuracy of the individual parameter estimates, and second, the RIMSE of the estimates was computed to evaluate how well the estimation procedures had recovered the test information functions. The averages of the discrimination parameter

estimate RMSEs across the five replications of each test are shown in Figures 8, 9, and 10. The averages of the difficulty parameter estimate RMSEs across the five replications of each test are shown in Figures 11, 12, and 13.

Figure 8 shows the average discrimination estimate RMSE across the five replications of the 15-item test at all 101 combinations of λ and α. Starting in the top left corner and moving clockwise, each panel shows the RMSEs for the estimates taken from samples of 100, 200, 500, and 1,000 examinees. The λ value for each model is shown on the x-axis, increasing from left to right. Each of the solid colored lines shows the estimates for a different value of α. The red lines show the RMSEs for RMML models fitted with the ridge penalty, that is, when $\alpha = 0$. At the other end of the α scale, the purple lines show the RMSEs for RMML models fit with the LASSO penalty, that is, when $\alpha = 1$. The blue, green, and yellow lines show the RMSEs for the RMML models fit with $\alpha = 0.25$, $\alpha = 0.5$, and $\alpha = 0.75$, respectively. The average RMSE of the MML estimates is at the extreme right, where $\lambda = 0$. The RMSEs of the four Bayesian estimates are shown by the horizontal dashed lines. The two orange lines denote the RMSEs of the mean and median of the Bayesian posterior distribution when a_j was estimated using a log-normal prior, and the two blue lines denote the RMSEs of the mean and median of the Bayesian posterior distribution when a_j was estimated using a uniform prior.

Figure 8 shows the discrimination parameter RMSE results for the 15-item tests. From the top left and moving clockwise, Panel A shows the results for the 100-examinee sample, Panel B shows the results for the 200-examinee sample, Panel C shows the results for the 500-examinee sample, and Panel D shows the results for the 1,000-examinee sample. Within each sample size, the trend is that the average discrimination parameter RMSE initially decreases as λ increase, but subsequently increases again. The decrease in the average RMSE is most notable for the item parameters estimated from the 100-examinee sample. As the sample size increases,

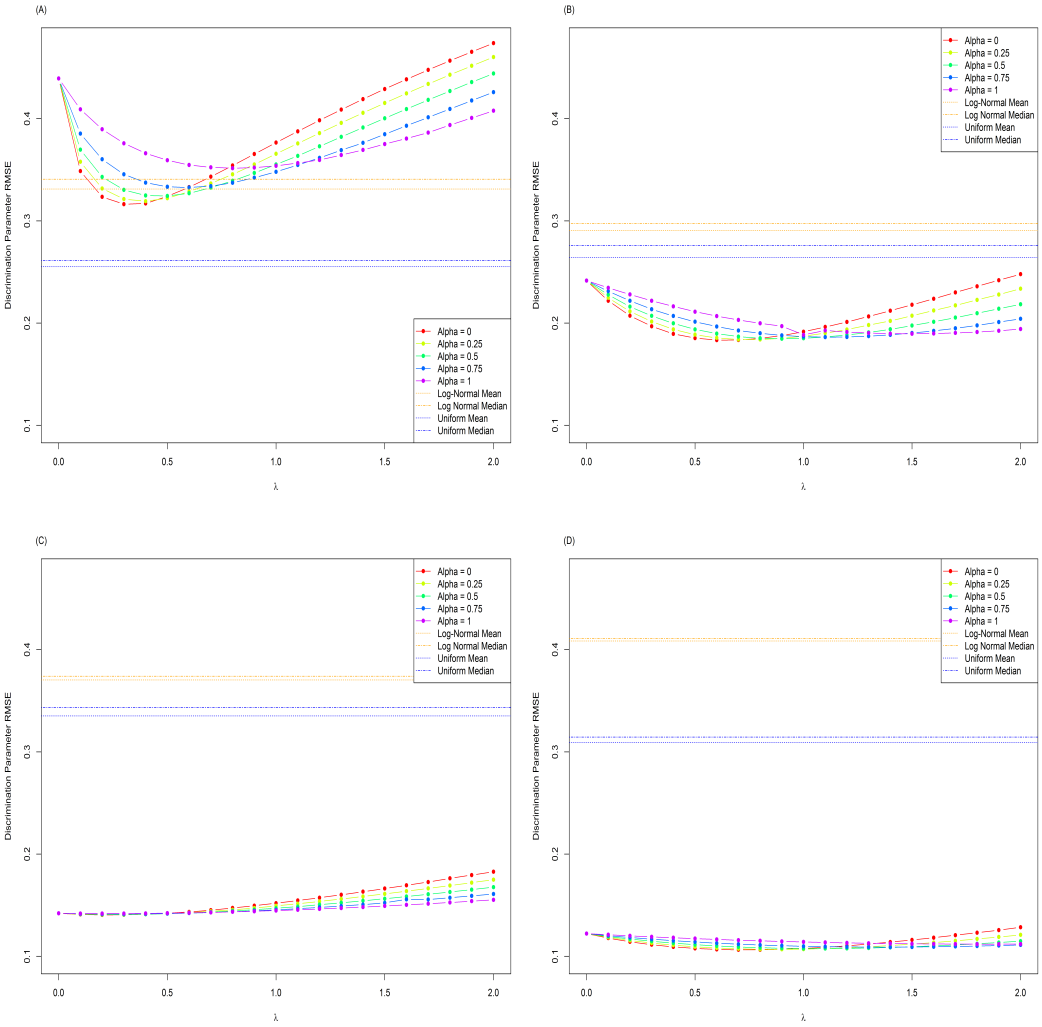

Figure 8: 15-Item Average Discrimination Parameter Estimate RMSEs
The red line shows the average RMSE when $\alpha = 0$ (the ridge penalty), the yellow line shows the average RMSE when $\alpha = 0.25$, the green line shows the average RMSE when $\alpha = 0.5$, the blue line shows average test RMSE when $\alpha = 0.75$, and the purple line shows the average RMSE when $\alpha = 1$ (the LASSO penalty). The blue and orange dashed lines show the average discrimination RMSE for the log-normal and uniform Bayesian models, respectively. Panel A shows the average RMSEs of the 100 examinee sample, Panel B shows the average RMSEs of the 200 examinee sample, Panel C shows the average RMSEs of the 500 examinee sample, and Panel D shows the average RMSEs of the 1,000 examinee sample.

the decrease in the average RMSE becomes much smaller. For all four sample sizes, the smallest RMSE was observed for the ridge penalty ($\alpha = 0$).

In all four sample size conditions, the discrimination parameter RMSE of the RMML estimates was less than the discrimination parameter RMSE of the MML estimates, for well chosen values of λ and α. Furthermore, in all four conditions, the RMML discrimination parameter RMSEs were lower than the discrimination parameter RMSEs of the Bayesian model using the log-normal prior for a, again for well chosen λ and α values. In the 100-examinee condition, the Bayesian model using the uniform prior for a had a lower discrimination parameter RMSE than any of the RMML models. For the 200-, 500-, and 1,000-examinee conditions, RMML yielded lower RMSEs. However, it should be noted that in all three of these conditions, the MML discrimination parameter estimates also had a lower average RMSE than the uniform prior Bayesian model.

Figure 9 shows the discrimination parameter RMSE results for the 25-item tests. As in Figure 8, Panel A shows the results for the 100-examinee sample, Panel B shows the results for the 200-examinee sample, Panel C shows the results for the 500-examinee sample, and Panel D shows the results for the 1,000-examinee sample. Once again, we observe that the general trend across all four sample size conditions is that the average RMML discrimination parameter RMSEs are lower than the average MML discrimination parameter RMSE when λ is low, but increase as λ increases. As in the 15-item condition shown in Figure 8, the greatest difference between the average MML discrimination parameter RMSE and the average RMML discrimination parameter RMSE was observed for the 100-examinee sample. As the sample size increase, the decrease in the RMSE became less marked, so that with 1,000 examinees it is barely noticeable. Although some decrease in the RMSE was observed for all five values of α, the greatest decrease was once again observed for the $\alpha = 0$ condition (equivalent to Hoerl and Kennard's (1970) ridge penalty).

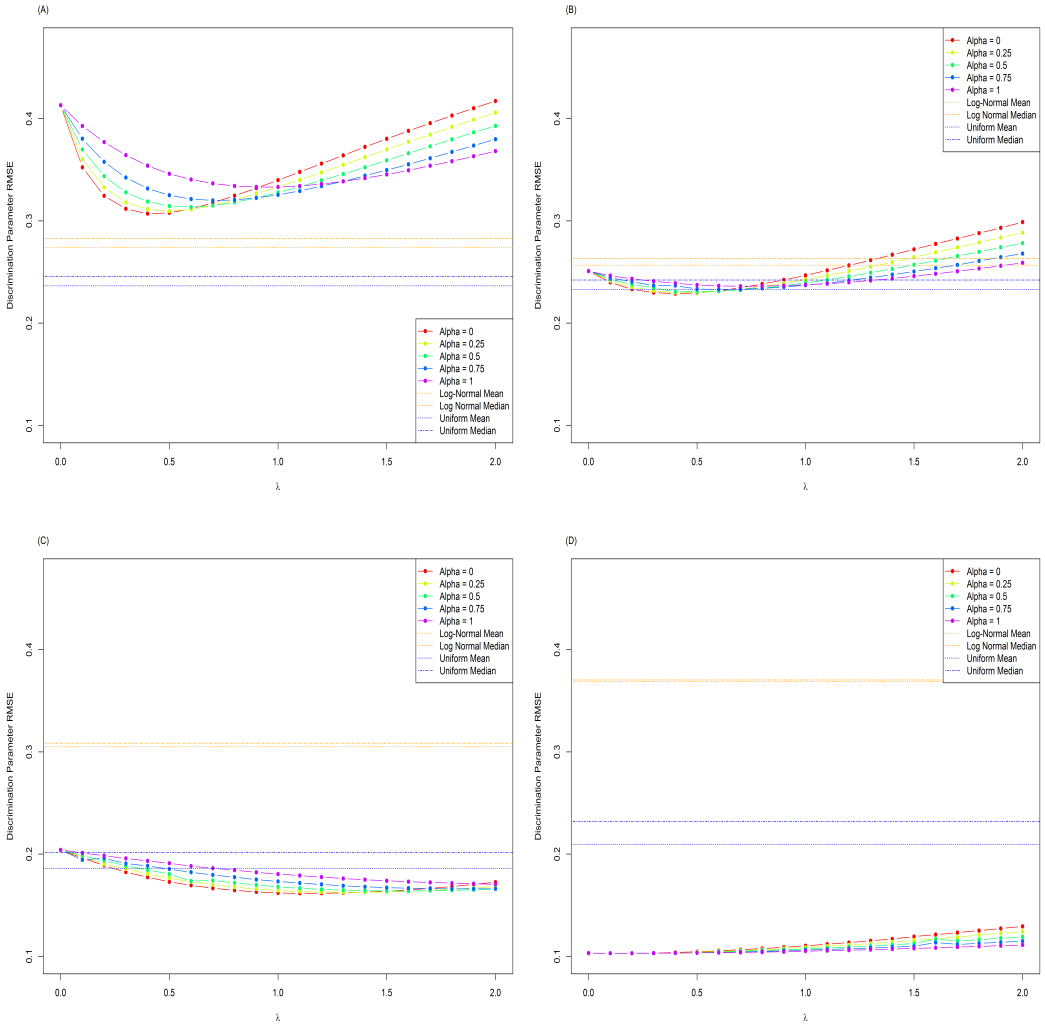

Figure 9: 25-Item Average Discrimination Parameter Estimate RMSEs
The red line shows the average RMSE when $\alpha = 0$ (the ridge penalty), the yellow line shows the average RMSE when $\alpha = 0.25$, the green line shows the average RMSE when $\alpha = 0.5$, the blue line shows average test RMSE when $\alpha = 0.75$, and the purple line shows the average RMSE when $\alpha = 1$ (the LASSO penalty). The blue and orange dashed lines show the average discrimination RMSE for the log-normal and uniform Bayesian models, respectively. Panel A shows the average RMSEs of the 100 examinee sample, Panel B shows the average RMSEs of the 200 examinee sample, Panel C shows the average RMSEs of the 500 examinee sample, and Panel D shows the average RMSEs of the 1,000 examinee sample.

The comparison with MML and Bayesian estimation for the results in Figure 9 exhibits some important differences from the results shown in Figure 8. In all four sample size conditions, the average RMML discrimination parameter RMSE was lower than the average MML discrimination parameter RMSE. However, in the 100-examinee condition, the average discrimination parameter RMSE of both of the Bayesian models was lower than for any of the RMML models. In the 200-examinee condition, the average RMML discrimination parameter RMSE was lower than either of the Bayesian models for low λ values. For high λ values, the average RMML discrimination parameter RMSE was higher not only than both of the Bayesian models, but in the most extreme cases when $\alpha = 0$ and λ was greater than 1.4, than the average MML discrimination parameter RMSE. In the 500- and 1,000-examinee conditions, the average RMML discrimination parameter RMSE was lower than the other three models, for well chosen values of λ and α. However, as in the 200-examinee model, over penalizing the model (i.e., increasing λ too much) led to a higher average discrimination parameter RMSE.

Finally, Figure 10 shows the average discrimination parameter RMSEs for the 35-item tests. The results are organized identically to the two previous conditions, so that the 100-, 200-, 500-, and 1,000-examinee conditions are shown in Panels A, B, C, and D, respectively. These results follow the same pattern as the two previous conditions: the average RMML discrimination parameter RMSE is lower than the average MML discrimination parameter RMSE when λ is low, but rises as λ increases, if λ is increased too much, the average RMML discrimination parameter RMSE exceeds the average MML discrimination parameter RMSE, and the decrease in the average RMSE is greatest for the 100-examinee sample.

In the 100-examinee condition, the RMML RMSE is greater than the Bayesian RMSEs for both models apart for a small range of λ values between 0.3 and 0.5 in which RMML performs better than the uniform Bayesian model posterior mean. In

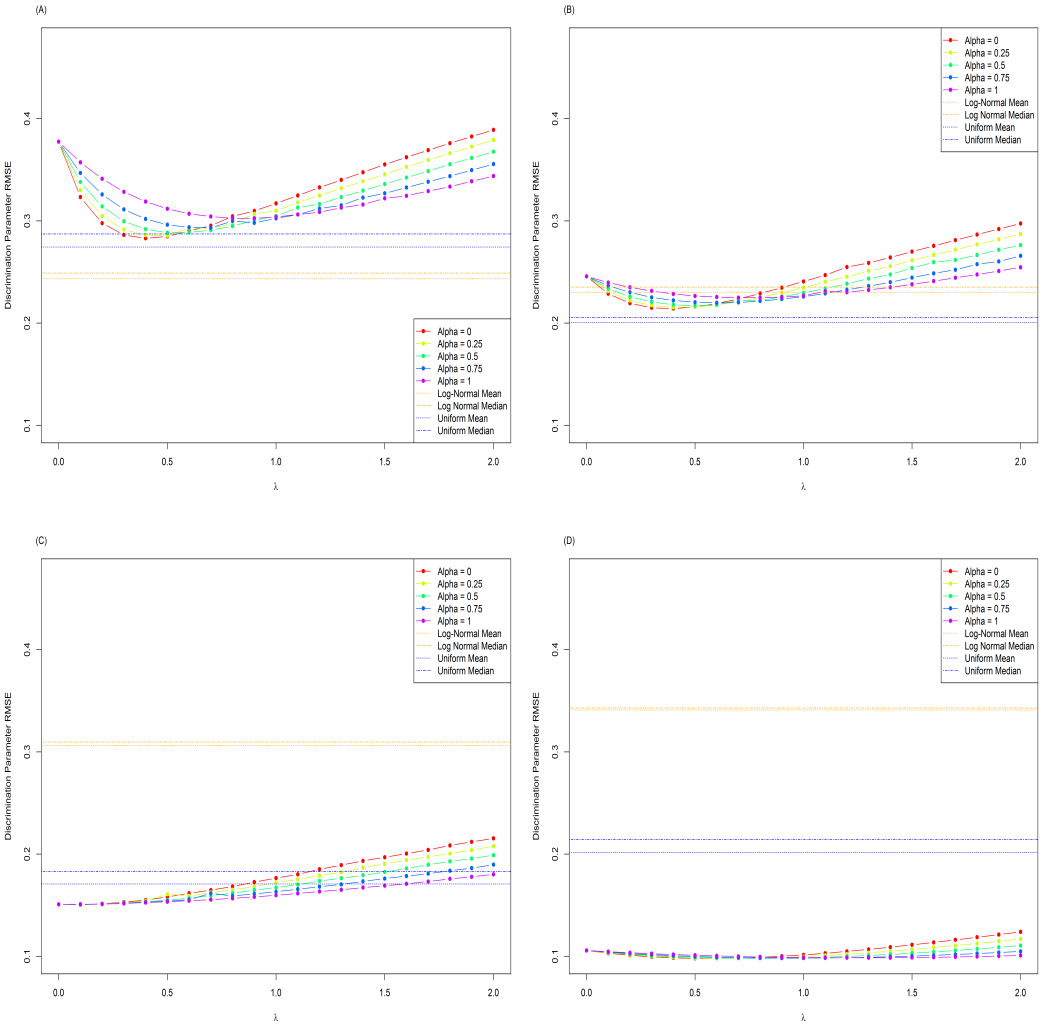

Figure 10: 35-Item Average Discrimination Parameter Estimate RMSEs
The red line shows the average RMSE when $\alpha = 0$ (the ridge penalty), the yellow line shows the average RMSE when $\alpha = 0.25$, the green line shows the average RMSE when $\alpha = 0.5$, the blue line shows average test RMSE when $\alpha = 0.75$, and the purple line shows the average RMSE when $\alpha = 1$ (the LASSO penalty). The blue and orange dashed lines show the average discrimination RMSE for the log-normal and uniform Bayesian models, respectively. Panel A shows the average RMSEs of the 100 examinee sample, Panel B shows the average RMSEs of the 200 examinee sample, Panel C shows the average RMSEs of the 500 examinee sample, and Panel D shows the average RMSEs of the 1,000 examinee sample.

the 200-examinee condition, RMML performs better than MML and the log-normal Bayesian model for λ between 0.1 and 0.8 for all five α values. For λ greater than 0.8, RMML is over-penalized, resulting in less accurate parameter estimates than either MML or the Bayesian models. The uniform Bayesian model out-performs all of the MML, RMML, and log-normal Bayesian models in this condition. In the final two conditions, with 500- and 1,000-examinees, RMML out performs the other models, but only very slightly. The decrease in RMSE for penalizing the discrimination parameter estimates in these conditions is truly negligible, and RMML becomes over penalized very quickly, resulting in increased RMSE. Furthermore, we once more observe that RMML out performing the Bayesian models in this condition is unsurprising, since MML outperforms the Bayesian models.

Simulation 1's results show that penalizing the discrimination parameter estimates can result in a lower RMSE than MML and Bayesian estimation, so long as we are careful in our choice of λ and α. The greatest RMSE reduction, relative to the RMSE of the MML discrimination parameter estimates, was observed when the sample size was small, regardless of test length. However, the Bayesian discrimination parameter estimates tended to have lower RMSEs than the RMML estimates for conditions typified by shorter tests. Doubling the sample size allows RMML to perform better, and for 200 examinees we observed that, on average, there were RMML estimates that had lower discrimination parameter RMSEs than all of the Bayesian estimates. Further increasing the sample size appears to result in situations where penalizing the discrimination parameter estimates does not yield much change in the RMSE, and where the MML estimates themselves have a lower RMSE than their Bayesian equivalents.

It should be noted that, while the difficulty and discrimination parameter estimates are independent across items by assumption, the difficulty and discrimination parameter estimates for a single item are not independent. This can be seen both in

the structure of the Hessian matrix given by (18), and in the conversion formula between the threshold/slope and difficulty/discrimination parameterization of the IRT model. Consequently, penalizing the discrimination or slope parameter estimates effects the difficulty or threshold parameter estimates. Thus, in addition to examining the effect of penalizing the discrimination parameters on the discrimination parameters themselves, it is also crucial to examine the effect of penalizing the discrimination parameters on the difficulty parameter estimates. This is shown in Figures 11, 12, and 13 for the 15, 25, and 35 item tests, respectively. As in the previous figures, Figures 11, 12, and 13 show the average difficulty parameter RMSE across the five replications of each test. The λ value is shown on the x-axis of each plot, and the α values are plotted as separate series. The horizontal orange and blue lines indicate the average RMSE across the five replicates of the Bayesian difficulty parameter estimates. Only one prior was used to estimate the difficulty parameters, a uniform distribution between -3 and 3. However, as in the discrimination parameter plots, we show the difficulty RMSEs separately for the log-normal and uniform discrimination parameter prior distributions. As before, Panel (A) in each figure shows the results for the 100 examinee samples, Panel (B) shows the results for the 200 examinee samples, Panel (C) shows the results for the 500 examinee samples, and Panel (D) shows the results for the 1,000 examinee samples.

Figure 11 shows the effects of penalizing the discrimination parameter estimates on the accuracy of the difficulty parameter estimates of the 15-item tests. For the two smallest sample size conditions, shown in Panels A and B, penalizing the discrimination parameter estimates resulted in the difficulty parameter estimates being less accurate than the MML difficulty parameter estimates. The loss of accuracy was more pronounced for higher λ values, show that the more we penalized the discrimination parameter estimates, the more accuracy we lost in the difficulty parameter estimates. Increasing the sample size appears to mitigate the loss of accuracy, as

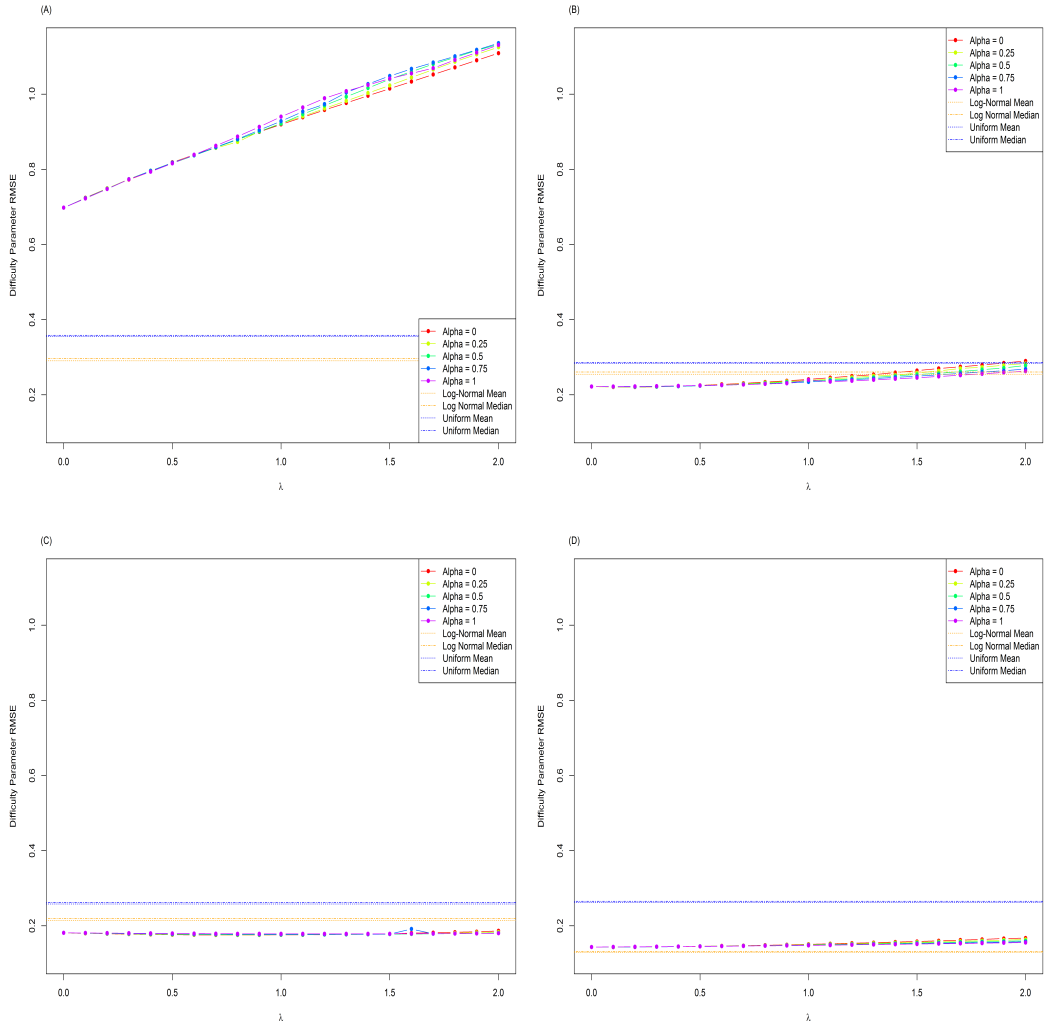

Figure 11: 15-Item Average Difficulty Parameter Estimate RMSEs
The red line shows the average RMSE when $\alpha = 0$ (the ridge penalty), the yellow line shows the average RMSE when $\alpha = 0.25$, the green line shows the average RMSE when $\alpha = 0.5$, the blue line shows average test RMSE when $\alpha = 0.75$, and the purple line shows the average RMSE when $\alpha = 1$ (the LASSO penalty). The blue and orange dashed lines show the average discrimination RMSE for the log-normal and uniform Bayesian models, respectively. Panel A shows the average RMSEs of the 100 examinee sample, Panel B shows the average RMSEs of the 200 examinee sample, Panel C shows the average RMSEs of the 500 examinee sample, and Panel D shows the average RMSEs of the 1,000 examinee sample.

shown in Panels C and D for the 500- and 1,000-examinee samples, respectively.

Another interesting observation made from the results shown in Figure 11 is that, if we were to account for the accuracy of the difficulty parameter estimates in our selection of tuning parameters, our choice of α is less clear than if we only accounted for the accuracy of the discrimination parameters. Recall that the discrimination parameter RMSEs were always minimized by selecting $\alpha = 0$. For the 15-items tests, the difficulty parameter RMSEs are minimized for $\alpha = 0$ only in Panel A, when the sample size of 100 examinees was used. In Panel B, the difficulty parameter RMSEs for all values of α are very similar for low values of λ. For high values of λ, the difficulty parameter RMSE is actually highest for $\alpha = 0$. The results in Panels C and D do not show any differences in the difficulty parameter estimates' RMSEs for different values of α in the range of λ tested.

Figure 12 shows the effects of penalizing the discrimination parameter estimates on the accuracy of the difficulty parameter estimates for the 25-item tests. With one notable exception, we see the same pattern of results observed for the 15-item tests in Figure 11. For small samples, the accuracy of the difficulty parameter estimates decreases as λ is increased, resulting in MML yielding the most accurate difficulty parameter estimates, as shown by Panels A and B. This effect is less pronounced when the sample size is increased to 500 or 1,000 examinees, as shown in Panels C and D. Importantly, however, Panels C and D do show some increase in the difficulty parameter estimate RMSE as λ is increased for the 25-item tests. This is in contrast to the results shown in Figure 11, which showed virtually no RMSE increase as λ was increased.

For all four sample sizes, the difficulty parameter RMSEs were lowest when the LASSO penalty ($\alpha = 1$) was used to penalize the discrimination parameter estimates. At the 100-examinee sample size (Panel A), this differs from the results shown in Panel A of Figure 11, which showed that the smallest RMML difficulty parameter RMSEs

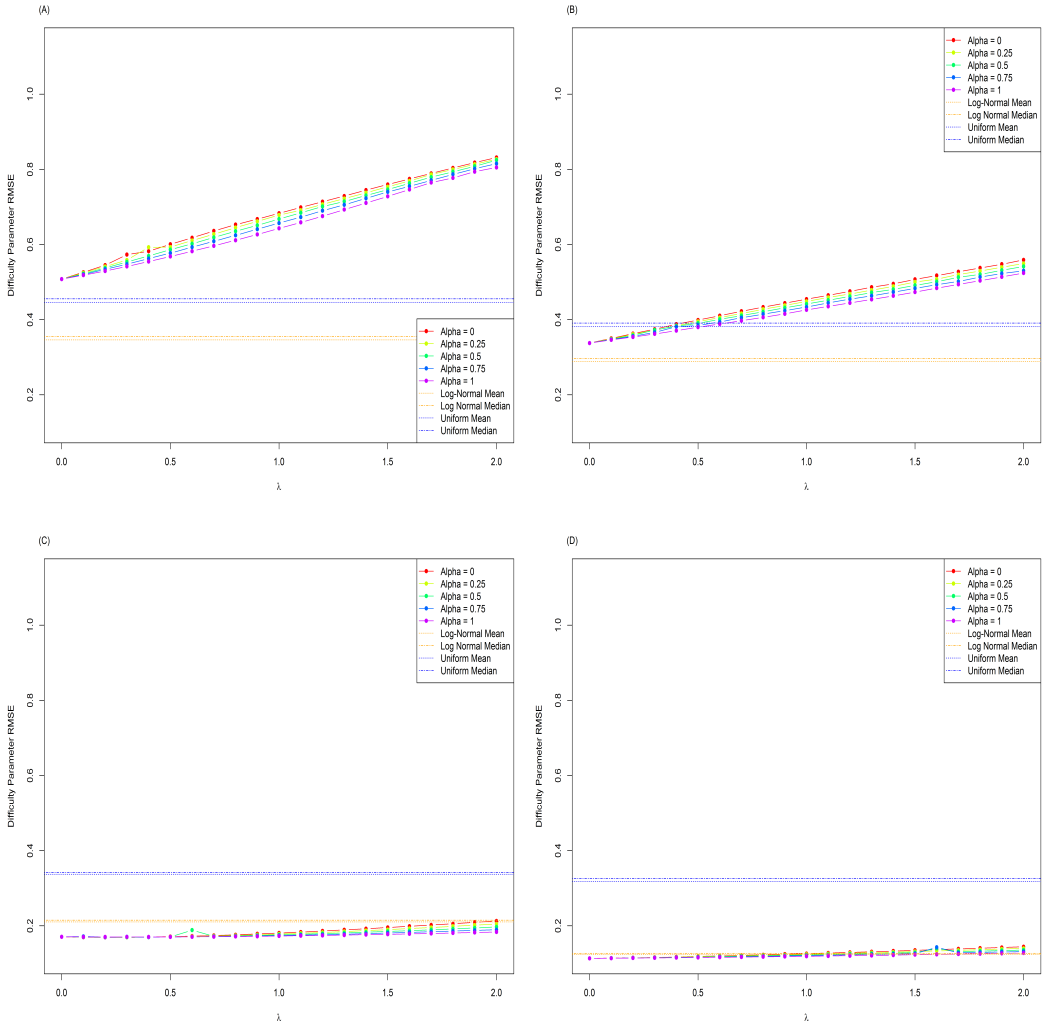

Figure 12: 25 Item-Test Average Difficulty Parameter Estimate RMSEs. The red line shows the average RMSE when $\alpha = 0$ (the ridge penalty), the yellow line shows the average RMSE when $\alpha = 0.25$, the green line shows the average RMSE when $\alpha = 0.5$, the blue line shows average test RMSE when $\alpha = 0.75$, and the purple line shows the average RMSE when $\alpha = 1$ (the LASSO penalty). The blue and orange dashed lines show the average discrimination RMSE for the log-normal and uniform Bayesian models, respectively. Panel A shows the average RMSEs of the 100 examinee sample, Panel B shows the average RMSEs of the 200 examinee sample, Panel C shows the average RMSEs of the 500 examinee sample, and Panel D shows the average RMSEs of the 1,000 examinee sample.

were observed when the ridge penalty ($\alpha = 0$) was used. They also differ from the results shown in Figures 8, 9, and 10, which showed that the RMML discrimination parameter RMSEs were also minimized by the ridge penalty.

Finally, Figure 13 shows the RMSEs for the 35-item test difficulty parameter estimates. As in the two previous figures, the RMSE results from the RMML models are shown as colored points, with position on the x-axis indicating the λ value used in fitting the model, and the color indicating the α value used in fitting the model as shown by the legend. The MML results are equivalent to the RMML when $\lambda = 0$, shown on the left hand side of each panel. The RMSEs for the Bayesian difficulty parameter estimates are shown as horizontal lines. Each of the Bayesian models yielded two estimates, the mean and median of the posterior distribution, as indicated in the legend. Blue lines represent the RMSEs of estimates from the Bayesian model with a log-normal prior on the discrimination parameters, and orange lines represent the RMSEs of estimates from the Bayesian model with a uniform prior on the discrimination parameters.

Once again, the RMSEs of the difficulty parameter estimates for the 35-item tests increase as the strength of the penalty on the discrimination parameter estimates increases. RMSE increases were observed in all four sample size conditions, although the difficulty parameter RMSE increase is least marked with the 1,000-examinee sample. With 1,000 examinees the log-likelihood function appears to overwhelm the penalty, as indicated by no evidence of differentiation between the difference α values in Panel D of Figure 13 unless λ is at least 1. In Panels A, B, and C, the increase in the difficulty parameter estimate RMSE marked for all λ greater than zero. We also observe that in all four sample size conditions, if we hold λ constant, the LASSO penalty ($\alpha = 1$) resulted in the least increase in the difficulty parameter RMSE and the ridge penalty ($\alpha = 0$) resulted in the greatest increase in the difficulty parameter RMSE.

These results make it abundantly clear that considering the accuracy of the dis-

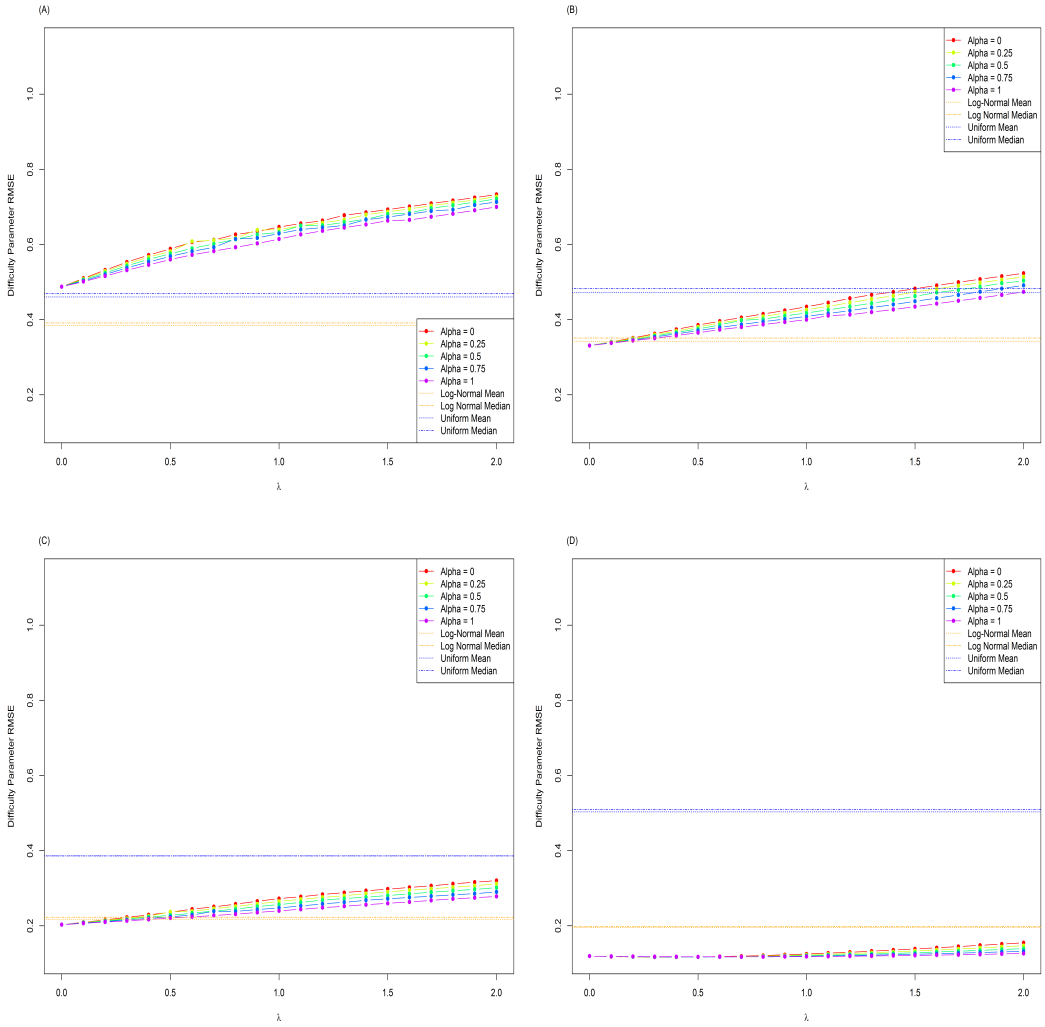

Figure 13: 35-Item Test Average Difficulty Parameter Estimate RMSEs
The red line shows the average RMSE when $\alpha = 0$ (the ridge penalty), the yellow line shows the average RMSE when $\alpha = 0.25$, the green line shows the average RMSE when $\alpha = 0.5$, the blue line shows average test RMSE when $\alpha = 0.75$, and the purple line shows the average RMSE when $\alpha = 1$ (the LASSO penalty). The blue and orange dashed lines show the average RMSE for the log-normal and uniform Bayesian models, respectively. Panel A shows the average RMSEs of the 100 examinee sample, Panel B shows the average RMSEs of the 200 examinee sample, Panel C shows the average RMSEs of the 500 examinee sample, and Panel D shows the average RMSEs of the 1,000 examinee sample.

crimination parameter estimates alone in evaluating the efficacy of RMML misses potentially important effects of the penalty function. Indeed, they raise the question of whether penalizing the discrimination parameter estimates really achieves anything, since in all of the conditions where the penalty function resulted in a significant drop in the discrimination parameter estimates' RMSE we also observed an increase in the difficulty parameter estimates RMSE. In order to answer this question, we need to examine the effects of penalizing the discrimination parameter estimates holistically, taking the accuracy of both sets of estimates into account. One method of doing this is to examine the root integrated mean square error (RIMSE), which measures the discrepancy between the true and estimated test information functions (TIFs). The RIMSE was assessed for every item using the estimates from all five replications of that test. The RIMSEs for an item were then averaged across replications, and the average RIMSE for a test was computed by taking the average across items. These are plotted in Figures 14, 15, and 16 for each combination of λ and α. In addition, each plot shows the average RIMSE for the MML model on the left hand side where $\lambda = 0$, and for the two Bayesian models, using the posterior mean as a point estimate. Figure 14 shows the results for the 15 item tests, Figure 15 shows the results for the 25 item tests, and 16 shows the results for the 35 item tests.

Figure 14 shows the RIMSE results for the 15-item tests. As in the previous figures, Panel A shows the results when the parameters are estimated using a sample of 100 examinees, Panel B shows the results when the parameters are estimated using a sample of 200 examinees, Panel C shows the results when the parameters are estimated using a sample of 500 examinees, and Panel D shows the results when the parameters are estimated using a sample of 1,000 examinees. In Panel A we see that the lowest RIMSE was observed when $\lambda = 0.1$ and $\alpha = 0$, demonstrating that penalizing the discrimination parameter estimates can improve recovery of the item information functions. However, this improvement is modest at best, and there is

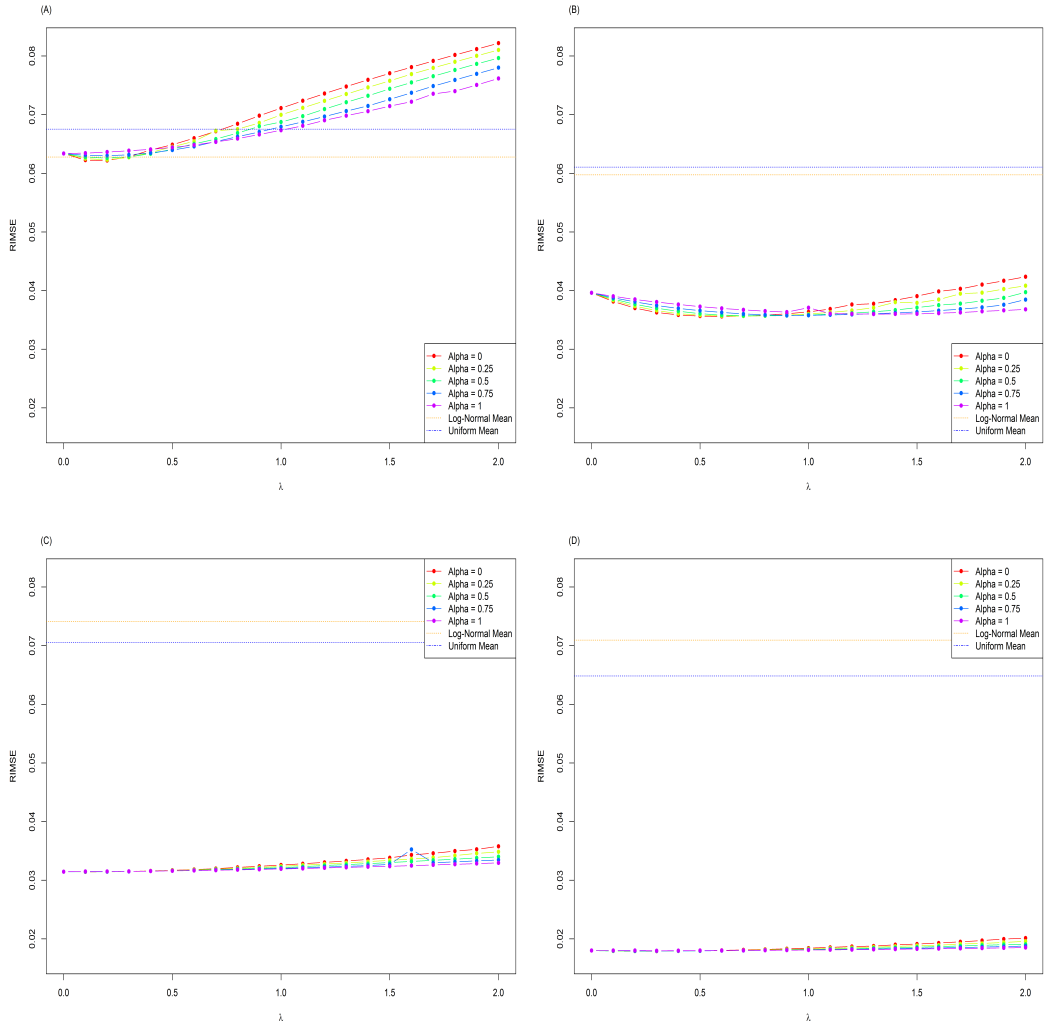

Figure 14: 15-Item Average RIMSEs

The red line shows the average RIMSE when $\alpha = 0$ (the ridge penalty), the yellow line shows the average RIMSE when $\alpha = 0.25$, the green line shows the average RIMSE when $\alpha = 0.5$, the blue line shows average test RIMSE when $\alpha = 0.75$, and the purple line shows the average RIMSE when $\alpha = 1$ (the LASSO penalty). The blue and orange dashed lines show the average discrimination RIMSE for the log-normal and uniform Bayesian models, respectively. Panel A shows the average RIMSEs of the 100 examinee sample, Panel B shows the average RIMSEs of the 200 examinee sample, Panel C shows the average RIMSEs of the 500 examinee sample, and Panel D shows the average RIMSEs of the 1,000 examinee sample.

very little difference between the average RIMSE of the MML, the Bayesian model with the log-normal prior on the discrimination parameters, and RMML using these optimal value of λ and α. Panel B shows similar results, albeit with some important differences. The RIMSE of all of the RMML estimates in this condition were lower than the RIMSE of the two Bayesian estimates. There is also evidence that RMML has better item information function recovery than MML, depending on the λ can α values selected. However, unlike our previous results, Panel B shows that both λ and α effect the RIMSE. Specifically, the RIMSE was minimized at both $\lambda = 0.5$ and $\alpha = 0$ and at $\lambda = 1.5$ and $\alpha = 1$. This demonstrate that the selection of tuning parameters in the IRT context is not necessarily straight forward, and that if both tuning parameters are allowed to vary, the minima may not be unique. This interesting result is not replicated in either Panel C or Panel D, both of which show that the RIMSE increases as the discrimination parameter estimates are penalized.

The results shown in Figure 15 for the 25-item tests follow a similar pattern to those shown in Figure 14. Panels A and B show that, for the 100- and 200-examinee samples, respectively, the unpenalized MML estimates had a higher RIMSE than the penalized RMML when λ is low. In both conditions, the RMML RIMSE was minimized by choosing $\lambda = 0.2$ and $\alpha = 0$. Increasing λ resulted in a higher RIMSE, indicating that the item information function is recovered less accurately with a stronger penalty. There is also no evidence that choosing a different α and a higher λ could yield equally accurate results, as in Panel B of Figure 14. Furthermore, RMML using the optimal λ and α values yielded a lower RIMSE than either of the Bayesian models in both of these conditions. Interestingly, Panels C and D show that, even with larger samples, RMML can slightly improve the recovery of the item information functions. Increasing the sample size appears to increase the λ value that minimizes the RIMSE: with 500 examinees the RIMSE is minimized for $\lambda = 0.9$ and with 1,000 examinees the RIMSE is minimized for $\lambda = 0.8$. However, the difference

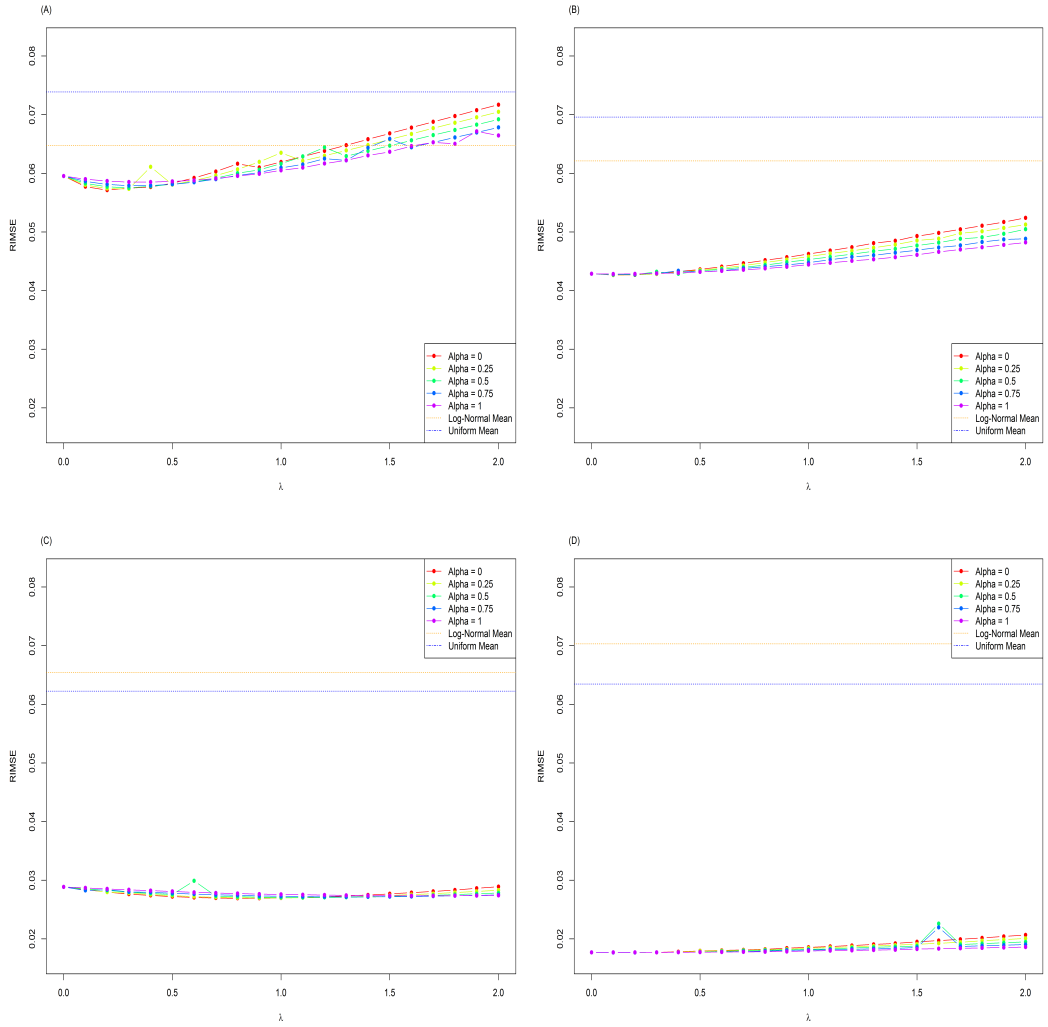

Figure 15: 25 Item-Test Average RIMSEs.
The red line shows the average RIMSE when $\alpha = 0$ (the ridge penalty), the yellow line shows the average RIMSE when $\alpha = 0.25$, the green line shows the average RIMSE when $\alpha = 0.5$, the blue line shows average test RIMSE when $\alpha = 0.75$, and the purple line shows the average RIMSE when $\alpha = 1$ (the LASSO penalty). The blue and orange dashed lines show the average discrimination RIMSE for the log-normal and uniform Bayesian models, respectively. Panel A shows the average RIMSEs of the 100 examinee sample, Panel B shows the average RIMSEs of the 200 examinee sample, Panel C shows the average RIMSEs of the 500 examinee sample, and Panel D shows the average RIMSEs of the 1,000 examinee sample.

between the RIMSE for the RMML estimates and the RIMSE for the MML estimates in these conditions is very slight.

Figure 16 shows that the RIMSEs for the 35-item tests follow much the same pattern as for the 15- and 25-item tests. Specifically, Panel A shows that penalizing the discrimination parameter estimates when working with a sample of 100 examinees reduces the RIMSE and improves recovery of the Item Information Function. However, as in the two previous test length conditions, this is only true for low λ values. If a λ larger than 0.5 is used, the RIMSE using the penalized RMML estimates exceeds the RIMSE calculated with the MML estimates. With 200 or more examinees in the sample, penalizing the discrimination parameter estimates either had no effect on the RIMSE, or increased the RIMSE. Furthermore, with 200 or more examinees, the penalty associated with the lowest RIMSE is the LASSO, $\alpha = 1$, rather than the ridge penalty, $\alpha = 0$, which was associated with the lowest discrimination parameter RMSEs and the lowest RIMSE among the penalized models in the 15-item and 25-item results and the 35-item results with a sample of 100 examinees.

Simulation 1 demonstrated that under certain conditions, the estimation error of the RMML discrimination parameter estimates is lower than the estimation error of the MML discrimination parameter estimates. Furthermore, using RMML improves the recovery of the test information function compared to the results of MML. In some case, RMML also yielded lower RMSE than the Bayesian models used in this simulation. These improvements were most notable for small samples and short tests. However, Simulation 1 also showed that penalizing the discrimination parameters alone increases the estimation error of the difficulty parameters. For small samples, the improvement in the discrimination parameter estimates appears to outweigh the additional error incurred in the difficulty parameter estimates, as is shown in the RIMSE results for the 100 examinee samples. However, when the parameters are estimated from large samples, the improvement in the discrimination parameter esti-

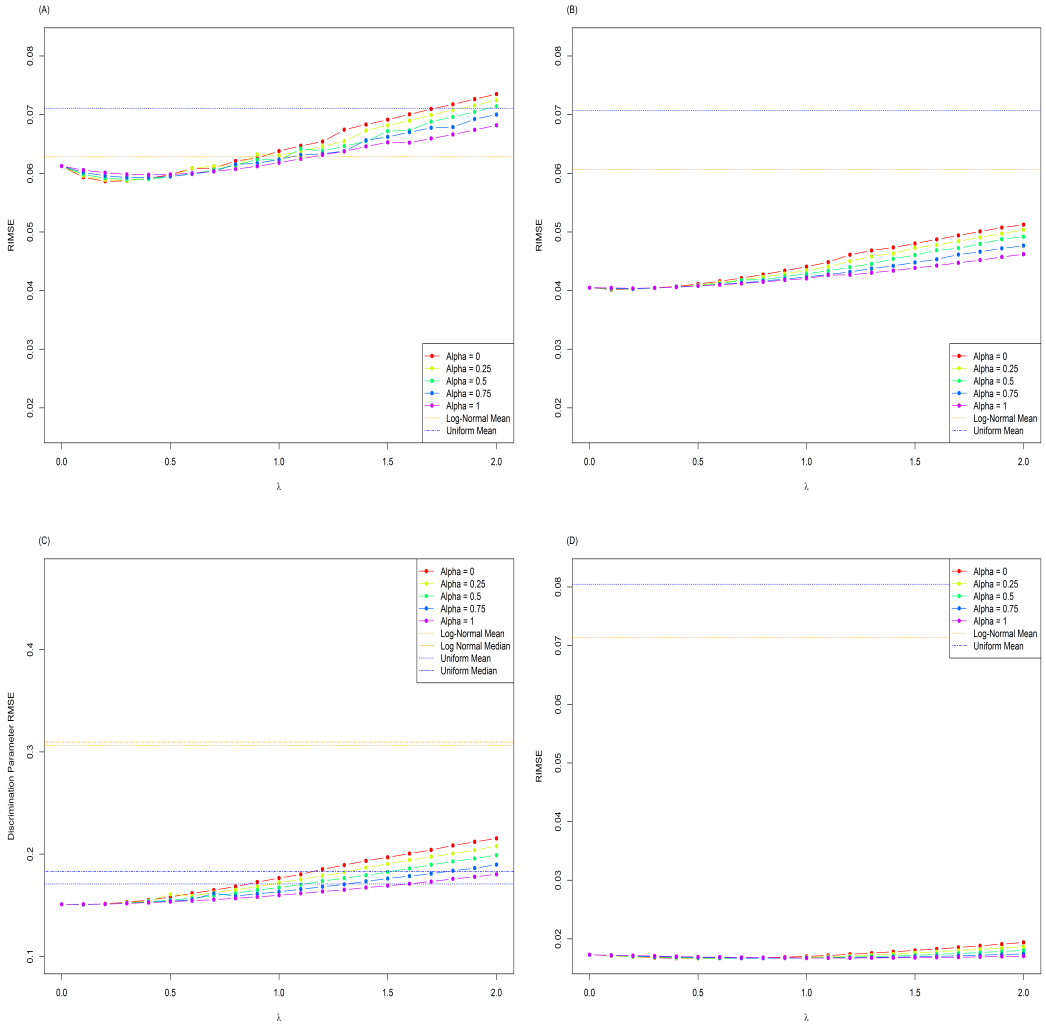

Figure 16: 35-Item Test RIMSEs

The red line shows the average RIMSE when $\alpha = 0$ (the ridge penalty), the yellow line shows the average RIMSE when $\alpha = 0.25$, the green line shows the average RIMSE when $\alpha = 0.5$, the blue line shows average test RIMSE when $\alpha = 0.75$, and the purple line shows the average RIMSE when $\alpha = 1$ (the LASSO penalty). The blue and orange dashed lines show the average RIMSE for the log-normal and uniform Bayesian models, respectively. Panel A shows the average RIMSEs of the 100 examinee sample, Panel B shows the average RIMSEs of the 200 examinee sample, Panel C shows the average RIMSEs of the 500 examinee sample, and Panel D shows the average RIMSEs of the 1,000 examinee sample.

mates is insufficient to cancel out the loss of accuracy in the difficulty parameters. At best, there is no benefit to using RMML as opposed to MML in these situations. At worst, the recovery of the test information function suffers, sometimes markedly, for penalizing the discrimination parameters. Simulation 1's results also show that there is some interplay between the two tuning parameters, sample size, and test length. In all of the conditions where RMML produced more accurate estimates than MML, the most accurate estimates came from a model using the ridge penalty. However, in the condition with 200 examinees taking 15 items, the LASSO penalty yielded a RMSE almost as low as the ridge penalty, albeit for a much higher λ. This pattern was not observed under any other conditions, though it is interesting to note that the LASSO penalty generally had a more consistent effect on the RIMSE of the test over the different values of λ tested.

It should be noted that Simulation 1 was designed in such a way that the simulated tests mimic conditions commonly observed in the achievement testing domain (Gao & Chen, 2005). However, there is another condition in which RMML could hypothetically out-perform MML and Bayesian estimation, namely when one or more of the items has a discrimination parameter value of zero. In practice this could occur when an item has face validity (i.e., it appears to measure the trait of interest), but actually provides no information about the trait being measured. Such items are removed from tests during the development stage, and the ability to accurately identify them is important. In Simulation 2 we generated tests with varying number of non-discriminating items to see how well RMML, MML, and Bayesian methods could estimate the item parameters under these conditions. Additionally, the difficulty of the test items and the discrimination of the discriminating items were varied to see how these factors interact with the number of non-discriminating items.

5.2 Simulation 2

The purpose of Simulation 2 was to compare the performance of MML, RMML, and Bayesian estimation under more controlled, though less naturalistic, conditions. Specifically, in Simulation 2 we simulated response matrices generated using fixed item parameter values, as detailed in Table 2. This allows us to investigate under what conditions RMML is more accurate than MML or Bayesian estimation more explicitly than was done in Simulation 1. Further, in Simulation 2, we included conditions with non-discriminating items, that is, items with $a_j = 0$. Such an item can be interpreted as having face validity (i.e., it appears to measure the trait subjectively), but does not actually measure the latent trait. Non-discriminating items are typically revised or removed during the instrument development process, and the ability to identify them is important.

RMML should hypothetically work well for identifying non-discriminating items because it shrinks all of the discrimination parameters towards zero. In contrast, MML does not shrink the discrimination parameters, and the Bayesian models shrink the parameter estimates towards the means of their prior distributions. This may lead to the Bayesian methods over-estimating the discrimination parameter when it is small, which in turn may lead to items needing revision not being identified. The estimation procedures used in Simulation 2 are the same procedures described at the beginning of this section.

Tables 3 through 9 contain the results of Simulation 2. Each table represents the result for tests containing a different number of non-discriminating items. The tests reported in Table 3 contained zero non-discriminating items (i.e., all of the items on the test had the discrimination parameter value reported in the left most column). Each subsequent table increases the number of non-discriminating items by 2 so that Table 5 shows the results for tests containing two non-discriminating items, Table 6 shows the results for tests containing four non-discriminating items, and so on. In the

final condition, half of the items on the test were non-discriminating. The results for this condition are shown in Table 9. Admittedly, this condition is unrealistic in the number of non-discriminating items on the test. However, it provides some interesting insight into the performance of RMML.

Rather than showing the results of all 101 RMML models as we did for Simulation 1, Tables 3 through 9 show only the lowest RMSE_a and RMSE_b observed of each test, along with the λ and α tuning parameters used to obtain that RMSE. The pattern of results observed across λ and α in Simulation 2 mirrored the pattern observed in Simulation 1 for the 100 examinee samples. Penalizing the discrimination parameter estimates resulted in a lower RMSE than was observed for the unpenalized MML estimates. Under some conditions, the RMML penalized estimates also yielded lower RMSEs for the discrimination parameter estimates than the Bayesian models. These conditions are described in detail below. The RMSEs of the difficulty parameters were generally not reduced by penalizing the discrimination parameters. As in Simulation 1, the difficulty RMSEs typically increased as λ increased, though there were a few conditions in Simulation 2 were penalizing the discrimination parameters to some degree also yielded lower difficulty parameter RMSEs than were observed for the MML estimates. However, overall RMML estimated the difficulty parameters less accurately than the Bayesian models.

Table 3 shows the results of the conditions containing zero non-discriminating items. These conditions represent a base line from which to compare the effects of including non-discriminating items on the three estimation procedures. The first two columns give the true values of a_j and b_j used to generate the response matrices. As described in the previous section, three values of a_j and b_j were used to represent low ($a_j = 0.5$), medium ($a_j = 1.0$) and high ($a_j = 1.5$) discrimination and low ($b_j = -1$), medium ($b_j = 0$) and high ($b_j = 1$) difficulty. The next six columns present the RMSEs for the discrimination parameter estimates from the three procedures. As

Table 3: Discrimination and Difficulty Parameter RMSEs for Tests With Zero Non-Discriminating Items

True a_j	True b_j	Discrimination Parameter ($\hat{a}_j$) RMSEs						Difficulty Parameter ($\hat{b}_j$) RMSEs					
		MML	RMML			Bayesian		MML	RMML			Bayesian	
			λ	α	$RMSE_a$	Log-Normal	Uniform		λ	α	$RMSE_b$	Log-Normal	Uniform
0.5	-1	0.38	2	0	0.27	0.29	0.36	1.49	0	0	1.49	0.54	0.65
	0	0.31	2	0	0.23	0.29	0.35	1.14	1	0	1.13	0.18	0.17
	1	0.47	2	0	0.25	0.35	0.43	1.43	1.2	0	1.28	0.63	0.68
1	-1	0.41	0.5	0	0.35	0.41	0.47	0.67	0	0	0.67	0.25	0.35
	0	0.34	0.4	0	0.32	0.41	0.50	0.32	0	0	0.32	0.19	0.19
	1	0.35	0.3	0	0.33	0.45	0.52	0.63	0	0	0.63	0.35	0.43
1.5	-1	0.47	0.1	0	0.44	0.51	0.57	0.44	0	0	0.44	0.23	0.26
	0	0.33	0.1	0	0.32	0.52	0.60	0.18	0	0	0.18	0.16	0.18
	1	0.47	0.2	0	0.35	0.55	0.61	0.23	0.1	0.25	0.21	0.17	0.25

noted previously, Table 3 shows only the lowest observed RMSE for RMML, along with its λ and α. In all conditions, the RMSEs yielded by the using the posterior means of both Bayesian models were lower than the RMSEs yielded by using the posterior medians. Consequently, only the RMSEs of the posterior means are reported in Table 3.

Table 3 shows that RMML yielded more accurate discrimination parameter estimates than MML in all nine conditions with zero non-discriminating items. The greatest reduction in the discrimination parameter RMSE was observed for the low-discrimination tests, particularly when the difficulty was high. As the discrimination increases, the differences between the average RMSEs of the RMML and MML estimates are less pronounced, though still not trivial. The smallest difference between the average RMML and MML discrimination parameter RMSEs was approximately 0.01 for the high discrimination – medium difficulty condition, and the largest difference between the RMML and MML discrimination parameter RMSEs was approximately 0.22 in the low discrimination – high difficulty condition. The difference between the average RMSEs of the RMML and MML parameter estimates with zero non-discriminating items was approximately 0.07. The RMML discrimination parameter estimates were also more accurate on average than the Bayesian discrimination parameter estimates in all nine zero non-discriminating items conditions. The largest difference between the average RMML discrimination parameter RMSE and the average RMSE of the Bayesian model with the log-normal prior was approximately 0.2, for the high discrimination, medium and high difficulty conditions. The largest difference between the average RMML discrimination parameter RMSE and the average RMSE of the Bayesian model with the uniform prior was approximately 0.28, for the high discrimination, medium difficulty condition.

As in Simulation 1, $\alpha = 0$ yielded the lowest discrimination parameter estimate RMSE in all nine conditions shown in Table 3. In the low discrimination conditions,

$\lambda = 2$ yielded the lowest discrimination parameter estimate RMSE. Note that this was the end of the λ range tested, implying that lower RMSEs could potentially be obtained by continuing to raise λ. Increasing the true discrimination parameter results in a decrease in the λ that yielded the lowest discrimination parameter estimate RMSE. Across replications and difficulty conditions, the average λ at the lowest RMSE for the medium discrimination tests was $\bar{\lambda} \approx 0.39$, and the average λ at lowest RMSE for the high discrimination tests was $\bar{\lambda} \approx 0.26$. This suggests that as the items become more discriminating, and more information is contained in the log-likelihood function, the penalty function becomes progressively less beneficial to the discrimination parameter estimates.

In the low discrimination conditions, the RMML discrimination parameter estimate RMSEs were very similar to the Bayesian discrimination parameter estimate RMSEs using the log-normal prior distribution. They were also smaller than the Bayesian discrimination parameter estimate RMSEs using the uniform prior distribution in the same conditions. In the medium and high discrimination conditions, the RMML discrimination parameter estimates had lower RMSEs than either of the Bayesian models. Across all three discrimination conditions, the log-normal Bayesian model yielded more accurate parameter estimates than the uniform Bayesian model.

As expected from the Simulation 1 results, penalizing the discrimination parameter estimates to any degree resulted in less accurate difficulty parameter estimates in a majority of the conditions presented in Table 3. In two thirds of the conditions, the lowest difficulty parameter RMSE occurred when $\lambda = 0$, which is equivalent to the MML model. In three conditions, small decreases in the difficulty parameter RMSE were achieved. In the low discrimination - medium difficulty condition, the smallest RMSE was observed when $\lambda = 1$ and $\alpha = 0$; in the low discrimination - high difficulty condition, the smallest RMSE was observed when $\lambda = 1.2$ and $\alpha = 0$; and in the high discrimination - high difficulty condition, the smallest difficulty parameter RMSE was

observed when $\lambda = 0.1$ and $\alpha = 0.25$. These three conditions are something of an anomaly given our previous observations about the behavior of the difficulty parameter RMSEs in Simulation 1. In the first condition, the average difficulty parameter RMSEs for $\alpha = 0$ are approximately the same as the MML average difficulty parameter RMSE regardless of λ, as shown in Figure 17. The RMSE at $\lambda = 1$ is a departure from the overall trend, and is influenced by an outlier for which the difficulty parameter RMSE was markedly lower than the other tests in this condition. Figure 18 shows that in the low discrimination - high difficulty condition, the RMML difficulty parameter RMSEs when $\alpha = 0$ are slightly lower than the MML difficulty parameter RMSE across all values of λ. The RMSE when $\lambda = 1.2$ is similarly influenced by an outlier, resulting in an observation which deviates from the broader trend. The difficulty parameter RMSEs in the high discrimination - high difficulty condition show a pattern of behavior atypical of our previous results. In this condition, there was a slight decrease in the difficulty parameter RMSE for low values of λ and α, followed by a sharp increase as λ increased. The increase was more marked for lower α values, as shown in Figure 19, but also began at a higher λ.

In addition to investigating the impact of the discrimination and difficulty parameters on the accuracy of the RMML, MML, and Bayesian parameter estimates, Simulation 2 also sought to assess the influence of non-discriminating items on the accuracy of these methods. Table 4 shows the average RMSEs of the four estimation methods collapsed across the discrimination and difficulty conditions for each non-discriminating item condition ($N_{a_j=0}$). The standard deviations of the RMSEs are shown in parentheses next to the average. As before, the first block shows the RMSEs for the discrimination parameter estimates, and the second block shows the difficulty parameter estimate RMSEs.

The MML discrimination and difficulty parameter estimate RMSEs were higher RMSEs for tests with more non-discriminating items. With zero non-discriminating

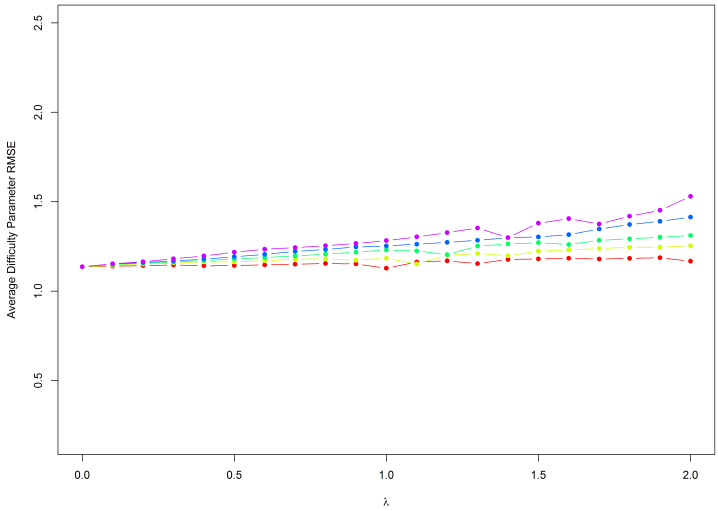

Figure 17: Average RMML Difficulty Parameter RMSEs for the Low Discrimination - Medium Difficulty Condition
The red line shows the average difficulty parameter RMSE when $\alpha = 0$, the yellow line when $\alpha = 0.25$, the green line when $\alpha = 0.5$, the blue line when $\alpha = 0.75$, and the purple line when $\alpha = 1$.

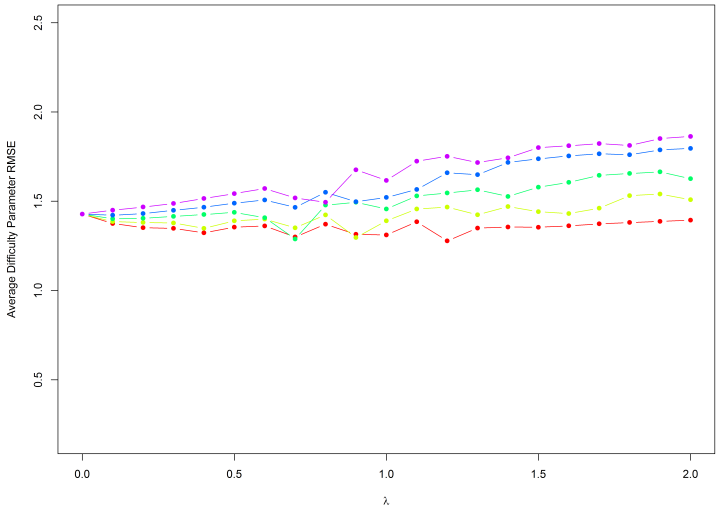

Figure 18: Average RMML Difficulty Parameter RMSEs for the Low Discrimination - High Difficulty Condition
The red line shows the average difficulty parameter RMSE when $\alpha = 0$, the yellow line when $\alpha = 0.25$, the green line when $\alpha = 0.5$, the blue line when $\alpha = 0.75$, and the purple line when $\alpha = 1$.

Table 4: Average Discrimination and Difficulty Parameter RMSEs By Number of Non-Discriminating Items

$N_{a_j=0}$	Discrimination Parameter RMSEs				Difficulty Parameter RMSEs			
	MML	RMML	Bayesian		MML	RMML	Bayesian	
			Log-Normal	Uniform			Log-Normal	Uniform
0	0.39 (0.06)	0.32 (0.06)	0.35 (0.18)	0.37 (0.13)	0.72 (0.51)	0.70 (0.48)	0.30 (0.17)	0.35 (0.20)
2	0.38 (0.05)	0.32 (0.06)	0.39 (0.14)	0.39 (0.07)	1.08 (0.37)	1.04 (0.35)	0.39 (0.20)	0.45 (0.23)
4	0.40 (0.06)	0.31 (0.06)	0.42 (0.11)	0.44 (0.05)	1.29 (0.33)	1.28 (0.32)	0.45 (0.22)	0.52 (0.27)
6	0.39 (0.05)	0.30 (0.05)	0.45 (0.07)	0.47 (0.05)	1.58 (0.15)	1.57 (0.16)	0.50 (0.24)	0.55 (0.28)
8	0.40 (0.12)	0.29 (0.03)	0.49 (0.05)	0.53 (0.04)	1.71 (0.32)	1.68 (0.28)	0.53 (0.28)	0.59 (0.31)
10	0.41 (0.08)	0.29 (0.04)	0.53 (0.04)	0.58 (0.03)	1.77 (0.17)	1.75 (0.15)	0.58 (0.31)	0.62 (0.34)

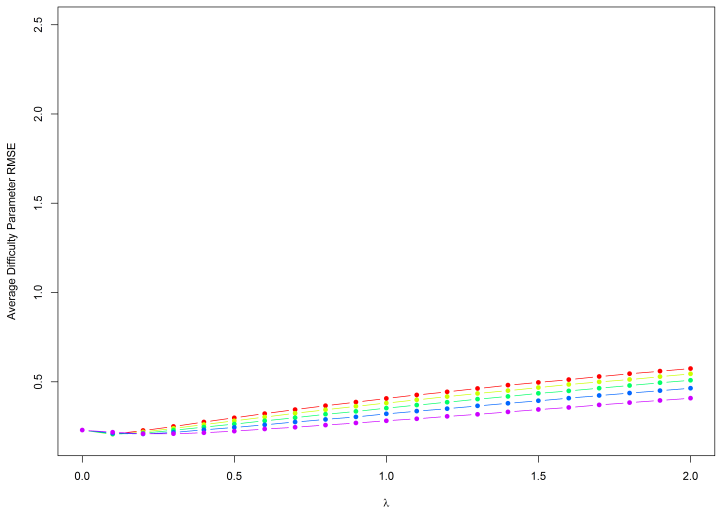

Figure 19: Average RMML Difficulty Parameter RMSEs for the High Discrimination - High Difficulty Condition
The red line shows the average difficulty parameter RMSE when $\alpha = 0$, the yellow line when $\alpha = 0.25$, the green line when $\alpha = 0.5$, the blue line when $\alpha = 0.75$, and the purple line when $\alpha = 1$.

items, the average MML discrimination parameter estimate RMSE was approximately 0.39 (SD = 0.06). With ten non-discriminating items, the average MML discrimination parameter estimate RMSE increased slightly to approximately 0.41, a slight increase. In contrast, the average RMSE of the RMML discrimination parameter estimates decreases from 0.32 (SD = 0.06) with zero non-discriminating items to 0.29 (SD = 0.04) with ten non-discriminating items. Like the average RMSE of the MML discrimination parameter estimates, the average Bayesian discrimination parameter estimate RMSEs increase from 0.35 (SD = 0.18) for the log-normal prior and 0.37 (SD = 0.13) for the uniform prior with zero non-discriminating items to 0.53 (SD = 0.04) for the log-normal prior and 0.58 (SD = 0.03) for the uniform prior with ten non-discriminating items. For all four methods, the average difficulty parameter RMSEs increased as the number of non-discriminating items increased. On average, the RMML difficulty parameter estimates' RMSEs were lower than the MML difficulty

parameter estimates' RMSEs. Additionally, the average Bayesian difficulty parameter estimate RMSEs in all conditions were lower than the average RMML and MML difficulty parameter estimate RMSEs. Of the two Bayesian models, the model with the log-normal prior on the discrimination parameters had lower average difficulty parameter estimate RMSEs.

Tables 5, 6, 7, 8, and 9 show the results of each condition in greater detail. Each table is laid out in the same manner as Table 3, with the true discrimination and difficulty parameters in the first two columns, followed by the average RMSEs of the MML, RMML, and Bayesian discrimination parameter estimates, and finally the average RMSEs of the MML, RMML, and Bayesian difficulty parameter estimates. The trends shown by these results are also similar to those shown in Table 3. For a fixed number of non-discriminating items, the RMML discrimination parameter estimates have the lowest average RMSEs when the non-zero discrimination parameters are low. The Bayesian estimates have lower RMSEs in the medium discrimination conditions, though not always lower than the average RMML estimate RMSEs. Across tables, the RMML discrimination parameter RMSEs are remarkably consistent, always falling between 0.25 and 0.4. However, the average MML and Bayesian discrimination parameter RMSEs increase more dramatically as more items are made non-discriminating. We also note that when the non-zero discrimination parameters are low, the λ tuning parameter that yields the smallest RMSE tends to be high, whereas when the non-zero discrimination parameters are high, the optimal λ value is smaller. In all but one condition, the α tuning parameter value that yielded the smallest average RMML discrimination parameter RMSE was zero. These observations are consistent with the observations made about Simulation 1's results and the results presented in Table 3.

The pattern among the difficulty parameter RMSEs was also similar to that observed in Table 3. For a fixed number of non-discriminating items, the average RMML

difficulty parameter RMSEs decrease as the non-zero discrimination parameter value increases. However, as more non-discriminating items are added to the test, this trend is mitigated, and the RMML difficulty parameter RMSEs increased overall. Additionally, penalizing the discrimination parameter estimates only improves the accuracy of the difficulty parameter estimates when the non-zero discrimination parameters were low. As before, this is indicated by the optimal λ being zero. In one of the conditions in Table 5, both the difficulty and discrimination parameter estimates were optimized for the same λ and α combination. However, this appears to be largely a fluke, as no pattern of both sets of parameter estimates being optimized by the same penalty emerges either within Table 5 or across tables.

In this section we have presented the results of two simulations evaluating the RMML algorithm under a variety of conditions. Simulation 1 showed that, in small samples, RMML yielded more accurate discrimination parameter estimates than MML, though these were not always as accurate as Bayesian parameter estimates, and came at the cost of less accurate difficulty parameter estimates. A holistic look at the recovery of the test information function (TIF) showed that RMML performed well only in very small sample sizes. In larger samples, the reduced difficulty parameter estimate accuracy outweighed any increase in the discrimination parameter estimates' accuracy achieved by penalizing the latter. Simulation 1's results also suggest that, from a holistic standpoint, both the λ and α tuning parameters should be set low. Specifically λs less than 1, and possibly less than 0.5, work best for regularized IRT. In almost all of the conditions across both simulations, the discrimination parameters' RMSE was minimized for $\alpha = 0$, which is equivalent to the ridge penalty.

Simulation 2 helped to elucidate our previous results. In this simulation we concentrated on a single sample size and manipulated the discrimination, difficulty, and number of non-discriminating items. Our results show that RMML works best when the item discrimination is low, though any improvements in the accuracy of the dis-

Table 5: Discrimination and Difficulty Parameter RMSEs for Tests With Two Non-Discriminating Items

True a_j	True b_j	Discrimination Parameters ($\hat{a}_j$) RMSEs						Difficulty Parameters ($\hat{b}_j$) RMSEs					
		MML	RMML			Bayesian		MML	RMML			Bayesian	
			λ	α	RMSE$_a$	Log-Normal	Uniform		λ	α	RMSE$_b$	Log-Normal	Uniform
0.5	-1	0.41	1.4	0	0.29	0.36	0.45	1.78	1.4	0	1.75	0.65	0.71
	0	0.36	2	0	0.24	0.35	0.43	1.14	0.2	0.25	0.92	0.17	0.17
	1	0.38	1.4	0	0.26	0.35	0.43	1.50	0.2	0.25	1.43	0.70	0.78
1	-1	0.37	0.2	0	0.36	0.26	0.32	1.22	0	0	1.22	0.39	0.47
	0	0.30	0.2	0	0.30	0.25	0.31	0.70	0	0	0.70	0.15	0.17
	1	0.31	0.3	0	0.28	0.24	0.30	0.91	0	0	0.91	0.47	0.57
1.5	-1	0.44	0.1	0	0.41	0.57	0.45	0.93	0	0	0.93	0.37	0.47
	0	0.42	0.2	0	0.38	0.56	0.44	0.76	0	0	0.76	0.19	0.17
	1	0.45	0.2	0	0.37	0.56	0.43	0.76	0	0	0.76	0.37	0.50

Table 6: Discrimination and Difficulty Parameter RMSEs for Tests With Four Non-Discriminating Items

True a_j	True b_j	Discrimination Parameters ($\hat{a}_j$) RMSEs						Difficulty Parameters ($\hat{b}_j$) RMSEs					
		MML	RMML			Bayesian		MML	RMML			Bayesian	
			λ	α	RMSE$_a$	Log-Normal	Uniform		λ	α	RMSE$_b$	Log-Normal	Uniform
0.5	-1	0.39	2	0	0.23	0.41	0.47	1.66	0.5	0.25	1.59	0.67	0.74
	0	0.34	2	0	0.24	0.40	0.48	1.45	0	0	1.45	0.17	0.17
	1	0.51	2	0	0.27	0.41	0.50	1.86	0.9	0	1.84	0.73	0.81
1	-1	0.41	0.5	0	0.34	0.31	0.37	1.25	0	0	1.25	0.50	0.62
	0	0.31	0.3	0	0.29	0.30	0.37	1.04	0	0	1.04	0.19	0.20
	1	0.35	0.5	0	0.29	0.30	0.38	1.18	0	0	1.18	0.60	0.74
1.5	-1	0.41	0.1	0	0.37	0.55	0.46	1.15	0	0	1.15	0.49	0.60
	0	0.45	0.2	0	0.39	0.55	0.47	0.76	0	0	0.76	0.14	0.13
	1	0.44	0.2	0	0.39	0.54	0.43	1.22	0	0	1.22	0.52	0.66

Table 7: Discrimination and Difficulty Parameter RMSEs for Tests With Six Non-Discriminating Items

True a_j	True b_j	Discrimination Parameters ($\hat{a}_j$) RMSEs						Difficulty Parameters ($\hat{b}_j$) RMSEs					
		MML	RMML			Bayesian		MML	RMML			Bayesian	
			λ	α	RMSE$_a$	Log-Normal	Uniform		λ	α	RMSE$_b$	Log-Normal	Uniform
0.5	-1	0.36	2	0	0.22	0.46	0.53	1.72	0.1	1	1.68	0.72	0.79
0.5	0	0.51	1.5	0	0.22	0.45	0.52	1.47	0.4	0	1.39	0.16	0.18
0.5	1	0.38	2	0	0.24	0.46	0.54	1.69	0.9	0	1.66	0.75	0.79
1	-1	0.42	0.5	0	0.34	0.37	0.43	1.67	0	0	1.67	0.61	0.69
1	0	0.36	0.5	0	0.33	0.37	0.44	1.62	0	0	1.62	0.19	0.18
1	1	0.31	0.3	0	0.29	0.36	0.43	1.68	0	0	1.68	0.66	0.76
1.5	-1	0.38	0.1	0	0.35	0.53	0.47	1.66	0	0	1.66	0.60	0.69
1.5	0	0.36	0.2	0	0.33	0.52	0.46	1.51	0	0	1.51	0.20	0.19
1.5	1	0.41	0.2	0	0.35	0.53	0.45	1.24	0	0	1.24	0.59	0.73

Table 8: Discrimination and Difficulty Parameter RMSEs for Tests With Eight Non-Discriminating Items

True a_j	True b_j	Discrimination Parameters ($\hat{a}_j$) RMSEs						Difficulty Parameters ($\hat{b}_j$) RMSEs					
		MML	RMML			Bayesian		MML	RMML			Bayesian	
			λ	α	RMSE$_a$	Log-Normal	Uniform		λ	α	RMSE$_b$	Log-Normal	Uniform
0.5	-1	0.50	2	0	0.25	0.51	0.57	2.03	0.8	0.5	1.98	0.77	0.82
	0	0.37	2	0	0.25	0.51	0.53	2.01	0.4	0.25	1.86	0.18	0.22
	1	0.69	2	0	0.29	0.52	0.60	2.20	0.8	0	2.13	0.81	0.88
1	-1	0.36	0.6	0	0.32	0.43	0.50	1.77	0	0	1.77	0.71	0.79
	0	0.30	0.5	0	0.26	0.41	0.49	1.42	0	0	1.42	0.17	0.18
	1	0.32	0.5	0	0.27	0.42	0.50	1.70	0	0	1.70	0.70	0.78
1.5	-1	0.35	0.2	0.25	0.32	0.55	0.50	1.32	0	0	1.32	0.63	0.73
	0	0.34	0.1	0	0.32	0.53	0.52	1.45	0	0	1.45	0.17	0.16
	1	0.41	0.3	0	0.32	0.54	0.53	1.46	0	0	1.46	0.68	0.77

Table 9: Discrimination and Difficulty Parameter RMSEs for Tests With Ten Non-Discriminating Items

True a_j	True b_j	Discrimination Parameters ($\hat{a}_j$)						Difficulty Parameters ($\hat{b}_j$)					
		MML	RMML			Bayesian		MML	RMML			Bayesian	
			λ	α	RMSE$_a$	Log-Normal	Uniform		λ	α	RMSE$_b$	Log-Normal	Uniform
0.5	-1	0.54	2	0	0.28	0.56	0.63	1.99	0	0	1.99	0.81	0.87
0.5	0	0.38	2	0	0.25	0.55	0.61	1.89	0.9	0	1.78	0.17	0.19
0.5	1	0.39	1.8	0	0.24	0.55	0.61	1.94	0.6	0.25	1.87	0.86	0.87
1	-1	0.33	0.7	0	0.26	0.49	0.55	1.71	0	0	1.71	0.75	0.83
1	0	0.32	0.8	0	0.26	0.47	0.55	1.51	0	0	1.51	0.16	0.16
1	1	0.46	0.9	0	0.31	0.49	0.55	1.93	0	0	1.93	0.77	0.83
1.5	-1	0.45	0.3	0	0.35	0.58	0.57	1.62	0	0	1.62	0.73	0.82
1.5	0	0.33	0.1	0	0.32	0.56	0.55	1.69	0	0	1.69	0.17	0.16
1.5	1	0.47	0.3	0	0.36	0.57	0.58	1.67	0	0	1.67	0.76	0.81

crimination parameters still come at the cost of less accurate difficulty parameter estimates. Our results suggest that the RMML difficulty parameter estimates are least accurate when the discrimination parameter estimates are most accurate, and vice verse. Compared to MML and Bayesian estimation, the average RMSE of the RMML discrimination parameter estimates is highly consistent regardless of the number of non-discriminating items. Unfortunately, the same cannot be said for the difficulty parameter estimates, which become less accurate the more non-discriminating items appear on the test. In the final section, we examine the findings from our simulations, discuss what they mean for applying RMML, and suggest some directions for future research.

6 Conclusion

Regularized parameter estimation is a rapidly expanding field that promises to allow statisticians and other scientists to fit models under a wider variety of conditions than is possible with classical estimation methods. We have sought to apply regularized estimation to IRT in order to improve small sample parameter estimation. The method we have proposed, RMML, builds on the well-established MML method proposed by Bock and Lieberman (1970). However, we have augmented the estimation of the discrimination parameter with the elastic net penalty (Zou & Hastie, 2005). Penalizing the discrimination parameter estimates theoretically allows us to both constrain the estimates to an acceptable range and to algorithmically drop items that do not measure the latent trait of interest (Paolino, 2013). Dropping poorly performing items from the model leaves more data from which to estimate the parameters of the remaining items, hypothetically improving their accuracy. Our results have shown that this is the case when the item parameters are estimated using a very small sample (e.g., 100 examinees), but that there is little benefit to using RMML when the

sample size is more than 500 examinees. In this section we conclude by discussing the implications of our findings, some possible methods of improving RMML, and directions for future research on regularized IRT estimation.

In our first study, we sought to simulated an achievement testing situation in which a sample of examinees takes a test comprised of items all measuring a single latent trait. To examine RMML's efficacy under a variety of conditions, we varied both the sample size and the test length. We then reviewed the accuracy of the discrimination and difficulty parameter estimates, and the TIF recovery as measures of RMML's performance. We also compared these metrics to three other estimation methods as described in Section 4. The results of this first study show that estimating a test's item parameters with RMML can improve the accuracy of the discrimination parameters compared to MML, when the estimates are derived from a small sample. However, this improvement comes at a cost in terms of the accuracy of the difficulty parameter estimates. Furthermore, the performance of RMML in terms of estimation accuracy was often worse than that of the two Bayesian models used for comparison.

This finding is of considerable interest both in terms of how regularized estimation works in the IRT context and its performance with other statistical models. Within the context of IRT, it appears that penalizing a subset of the item parameters causes the unpenalized parameters to be estimated with less accuracy. To date, the only other researchers to penalized a subset of the parameters are Tutz and Schauberger (2015), who penalized the parameters associated with a set of indicator variables for group membership as a means of detecting DIF. However, Tutz and Schauberger do not investigate the effect of their penalty on the difficulty parameter estimates their model yields, so it is unknown whether their results mirror ours.

In most other applications of regularized estimation, all of the parameters are penalized with the exception of the model intercept, which is often fixed by standardizing the data (Friedman *et al.*, 2010; Tibshirani, 1996; Zou & Hastie, 2005). Some

authors (e.g., Friedman *et al.*, 2010) have speculated about omitting a subset of the parameters from the estimation, but no research on the effects of penalizing only a subset of the parameters has been published. It would be intriguing to see whether the effect observed in our first study is also observed in linear or logistic regression.

Our simulation results also show that, holistically, RMML only improves the recovery of the TIF when the parameters are estimated from a small sample. With 100 examinees, RMML exhibited a lower average RIMSE than MML. With 200 or more examinees, however, penalizing the discrimination parameter estimates made little improvement to the recovery of the TIF and often resulted in worse TIF recovery. This is because the estimated TIF is a function of both the discrimination and difficulty parameter estimates. Conditions in which RMML resulted in a higher RIMSE than MML or the Bayesian models are the conditions in which the loss of accuracy in the difficulty estimates was greater than the improvement in the discrimination estimates, resulting in an overall higher RIMSE. Therefore, our results suggest that RMML is really only applicable when working with a very small sample.

Finally, our first simulation sought to provide some guidance on what values of the elastic net penalty tuning parameters, λ and α, work well in the IRT context. We estimated the item parameters for all the conditions in our first simulation using λ between zero and two (2) in increments of 0.1, and using α between zero and one (1) in increments of 0.25. Loosely speaking, λ determines the strength of the penalty, with zero yielding an unpenalized result equivalent to the MML estimates (Bock & Aitkin, 1981). The upper-bound of two used in our simulation is arbitrary, as λ can be increased without bound, and was chosen based on early simulation work showing that setting λ higher than two yielded no benefit in terms of the accuracy of the discrimination estimates. The interpretation of α is clearer. Fitting a penalized model with $\alpha = 0$ results in the same fit as if the ridge penalty (Hoerl & Kennard, 1970) had been used, and fitting a penalized model with $\alpha = 1$ results in the same fit

as if the LASSO penalty (Tibshirani, 1996) had been used (Zou & Hastie, 2005).

When RMML improved the recovery of the TIF or the accuracy of the discrimination parameter estimates, the λ that yielded the greatest improvement over MML was less than 0.5. This suggests that the penalty function need not be strong to improve the item parameter estimates, and that if a search for λ is to be conducted, it could be restricted to the interval between 0 and 0.5. In all of these conditions, the α that yielded the greatest improvement over MML was $\alpha = 1$. This result came as something of a surprise, given the goal of the endeavor, because the ridge penalty does not act as a selection operator (Friedman *et al.*, 2010; Tibshirani, 1996; Zou & Hastie, 2005). However, it should be noted that the most consistent results across different λ values were obtained by using $\alpha = 1$. This is important due to the uncertainty about the optimal value of λ in actual application of RMML.

Our second simulation investigated RMML's performance under less naturalistic conditions. Our goal was to examine how RMML performed when items that failed to measure the latent trait were present on the test. To do this we held the test length and sample size constant at 15 items and 100 examinees, and varied the number of non-discrimination items present on the test. In order to do this we also fixed the discrimination and difficulty values of the test items to common values, as described in Section 4. Each item parameter was set at either low, medium, or high, and then the parameter levels were fully crossed with the number of non-discriminating items to create a total of 54 conditions.

The results of our second simulation showed that RMML yields more accurate discrimination parameter estimates than either MML or the two Bayesian models examined in the presence of non-discriminating items. In fact, the RMSE of the discrimination parameter estimates was remarkably stable across conditions. Generally speaking, the difficulty parameter results were also akin to those observed in our first simulation. When some of the items are non-discriminating and the remaining items

had low discrimination, RMML yielded more accurate difficulty parameter estimates than MML. However, with more discriminating items RMML could not out perform MML in terms of the accuracy of the difficulty parameter estimates.

The reason RMML performed so well when the items had low discrimination can be seen through an analogy to Bayesian estimation. The effect of placing a ridge penalty on a parameter estimate can be replicated by using a normal prior distribution with a mean of zero and variance of $1/\lambda$ for the parameter in a Bayesian estimation (Friedman *et al.*, 2010). Similarly, the effect of placing a LASSO penalty on a parameter estimate can be replicated by using a Laplace prior distribution with a mean of zero and variance of $1/\lambda$. As both of these distributions are centered at zero, they will work best for low discrimination items.

Typically neither a normal or Laplace distribution centered at zero would be considered an appropriate prior distribution for the discrimination parameter in the 2PL because we know that the discrimination parameter must be positive. Swaminathan and Gifford (1985) suggested using a chi-distribution for the discrimination parameter, on the basis that, in the normal ogive model, it is inversely proportional to the square root of the variance. Mislevy (1986) and Gao and Chen (2005) recommend using either a log-normal distribution or a uniform distribution for the discrimination parameter prior distribution. All of these distributions are parameterized so that they are centered at or around one, rather than zero.

As noted earlier, Houseman *et al.* (2007) experimented with a modified version of the ridge penalty that shifted the penalty's center to some non-zero value. However, it should be noted that doing this fundamentally changes the purpose of the penalty function. In Houseman *et al.*'s model, the purpose of the penalty was to select between the 2PL and the 1PL at the item level. In our model, and the model proposed by Paolino (2013), the purpose is the select the most discriminating items so that we can focus on estimating their parameters, allowing us to extract the most information

from a response set for a limited number of examinees. The fact that the lowest RMSE and RIMSE were observed when $\alpha = 0$ suggests that using the penalty to select items for estimation may not be an ideal approach, since when $\alpha = 0$ the penalty will not eliminate any items. Ultimately it may be that the approach espoused by Houseman *et al.* (2007) is the more appropriate for IRT.

Regardless of whether the penalty is used to select the model or the items we wish to estimate, there are still considerable gaps in our knowledge around regularized estimation for IRT. From a pragmatic vantage point, the foremost of these is the lack of a reliable means of selecting the tuning parameters λ and α. In linear and logistic regression, λ is often selected by cross-validation, and α is selected subjectively based on the desired effect of the penalty (Friedman *et al.,* 2010; Tibshirani, 1996; Zou & Hastie, 2005). However, cross-validation relies on our being able to subset both the inputs and the outcomes. In the IRT context, the input to the equation is the latent trait, which is unknown. Houseman *et al.* (2007) and Tutz and Schauberger (2015) have suggested alternatives based on model fit statistics. Houseman *et al.* have even gone so far as to show that their method, based on an analogue to the AIC, yields the same λ as selecting based on the minimum squared error of estimation, as we have done. However, Houseman *et al.*'s approach requires that the objective function be twice differentiable, so it can only be used with the ridge penalty. Likewise, the approach recommended by Tutz and Schauberger requires us to know the model's degrees of freedom. Zou, Hastie, and Tibshirani (2007) showed that the number of non-zero parameter estimates is an unbiased estimate for the degrees of freedom of the LASSO penalty, and a similar result holds true for the ridge penalty. At this time there is no such neat solution for the elastic net penalty (Zou *et al.*, 2007, p. 2191). Thus more work is needed to find a more general solution.

It should also be noted that the results of our second simulation suggest that an entirely different approach to selecting the tuning parameters may be necessary. In

Simulation 2 we varied the discrimination parameters on the test as a factor with three levels, low ($a_j = 0.5$), medium ($a_j = 1$), and high ($a_j = 1.5$). This had the effect of fixing all of the items to the same discrimination parameter when the test contained no non-discriminating items. Under these conditions, we found that in the low discrimination condition, the lowest discrimination RMSE value was observed for the highest λ value tested. This implies that further reduction in the discrimination RMSE might have been possible if λ had increased further. It also implies that it might be more efficacious to penalize the items individually, rather than the test as a whole. Such an algorithm will require a more sophisticated method for selecting the tuning parameters, since trying all possible combinations of λ and α even for a limited range would become computation expensive for even modest tests.

Regularized Marginal Maximum Likelihood has demonstrated its potential to become another tool for modern test designers. While work remains to be done in several important areas, our study has shown that penalizing the discrimination parameters in the two-parameter logistic IRT model can reduce the RMSE of the parameter estimates. This reduction is especially pronounced when the sample of examinees available to estimate the parameters from is small, a situation commonly encountered both in psychology and in other fields.

References

Agresti, A. (2003). *Categorical Data Analysis.* Hoboken, NJ: Wiley-Interscience.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control, 19,* 716-723. doi: 10.1109/TAC.1974.1100705.

Andersen, E.B. (1970). Asymptotic properties of conditional maximum likelihood estimates. *The Journal of the Royal Statistical Society, Series B, 32,* 283-301.

Andersen, E.B. (1972). The numerical solution of a set of conditional estimation equations. *The Journal of the Royal Statistical Society, Series B, 34,* 42-54.

Andersen, E.B. (1973). Conditional inference in multiple choice questionnaires. *British Journal of Mathematical and Statistical Psychology, 26,* 31-44.

de Ayala, R.J. (2004). *The Theory and Practice of Item Response Theory.* New York: Guilford Press.

Baker, F.B. & Kim, S.H. (2004). *Item Response Theory: Parameter Estimation Techniques* (2nd Ed.). New York: Marcel Dekker, Inc.

Binet, A. & Simon, T. (1916). *The Development of Intelligence in Children: The Binet-Simon Scale.* Baltimore: Williams & Wilkins.

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In F. Lord & M. Novick (Eds.), *Statistical Theory of Mental Test Scores* (pp. 397-472). Reading, MA: Addison-Wesley.

Bock, R. & Aitkin, M. (1981). Marginal maximum likelihood estimation of item parameters: Application of an EM algorithm. *Psychometrika, 46,* 443-459. doi: 10.1007/BF02293801.

Bock, R. & Lieberman, M. (1970). Fitting a response model for n dichotomously scored items. *Psychometrika, 35,* 179-197. doi: 10.1007/BF02291262.

Boyd, S. & Vandenberghe, L. (2004). *Convex Optimization.* Cambridge: Cambridge University Press.

Broydan, C.G. (1970). The convergence of a class of double-rank minimization algo-

rithms. *Journal of the Institute of Mathematics and Its Applications, 6,* 79-90. doi: 10.1093/imamat/6.1.76.

Byrd, R.H., Lu, P., Nocedal, J., & Zhu, C. (1995). A limited memory algorithm for bounded constrained optimization. *SIAM J. Scientific Computing, 16,* 1190-1208.

Chang, H.H. (2015). Psychometrics behind computerized adaptive testing. *Psychometrika, 80,* 1-20. doi: 10.1007/s11336-014-9401-5.

Chen, S.S., Donoho, D., & Saunders, M. (1998). Atomic decomposition by basis pursuit. *SIAM Journal of Scientific Computing, 20*, 2313-2351.

Cohen, J. (1968). Multiple regression as a general data-analytic system. *Psychological Bulletin, 70,* 426.

Crocker, L. & Algina, J. (1986). *Introduction to Classical and Modern Test Theory.* Belmont, CA: Thompson-Wadsworth Publishing Inc.

DeGroot, M.H. & Schervish, M.J. (2002). *Probability and Statistics: Third Edition.* Boston, MA: Addison-Wesley.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B, 39,* 1-38.

Efron, B., Hastie, T., Johnstone, E., & Tibshirani, R. Least angle regression. *The Annals of Statistics, 32,* 407-499.

Efron, B. & Tibshirani, R. (1994). *An Introduction to the Bootstrap.* London, England: Chapman & Hall CRC Press.

Fan, X. (1998). Item response theory and classical test theory: An empirical comparison of their item/person statistics. *Educational and Psychological Measurement, 58*, 357-381. doi:10.1177/0013164498058003001.

Fletcher, R. (1970). A new approach to variable metric algorithms. *Computer Journal, 13,* 317-322.

Fraley, R., Waller, N., & Brennan, K. (2000). An item response theory analysis

of self-report measures of adult attachment. *Journal of Personality and Social Psychology, 78,* 350-365.

Friedman, J. H. (1988). Regularized discriminant analysis. *Journal of the American statistical association, 84,* 165-175.

Friedman, J. Hastie, T., Hoefling, H., & Tibshirani, R. (2007). Pathwise coordinate descent optimization. *Annals of Applied Statistics, 1,* 302-332.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software, 33*, 1-22. Retrieved from: http://www.jstatsoft.org/v33/i01.

Gao, F. & Chen, L. (2005). Bayesian or non-Bayesian: A comparison study of item parameter estimation in the three-parameter logistic model. *Applied Measurement in Education, 18*, 351-380. doi: 10.1207/s1532481ame1804_2.

Geman, D. & Geman S. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence. 6* (6), 721-741.

Goeman, J.J. (2012). L1 penalized estimation in the Cox proportional hazards model. *Biometrical Journal, 52*, 70-84.

Goldfarb, D. (1970). A family of variable metric updates derived by variational means. *Mathematics of Computation, 24,* 23-26.

Hambleton, R. & Swaminathan, H. (1985). *Item Response Theory: Principles and Applications.* Boston, MA: Kluwer-Nijhoff Publishing.

Hambleton, R. K., & Traub, R. E. (1970). Information Curves and Efficiency of Three Logistic Test Models.

Hanson, B.A. (2000). Estimation toolkit for item response models (ETIRM) [computer software]. Retrieved from: http://www.b-a-h.com/software/cpp/etirm.html.

Harwell, M.R., Baker, F., & Zwarts, M. (1988). Item parameter estimation via marginal maximum likelihood and an EM algorithm: A didactic. *Journal of Edu-*

cational Statistics, 13, 243-271. Retrieved from: http://www.jstor.org/stable/1164654.

Harwell, M.R. & Janosky, J.E. (1991). An empirical study of the effects of small data sets and varying prior variances on item parameter estimation in BILOG. *Applied Psychological Measurement, 15,* 279-291.

Harwell, M.R., Stone, C.A., Hsu, T.C., & Kirisci, L. (1996). Monte Carlo studies in item response theory. *Applied Psychological Measurement, 20,* 101-125.

Hastie, T., Rosset, S., Tibshirani, R. & Zhu, J. (2004). The entire regularization path for the support vector machine. *Journal of Machine Learning Research, 5,* 1391-1415.

Hastie, T., Tibshirani, R. & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* New York: Springer.

Hays, W. (1988). *Statistics.* Fort Worth, TX: Holt, Rinehart, & Winston.

Hoerl, A. & Kennard, R. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics, 12,* 55-67. Retrieved from: www.jstor.org/stable/1267351.

Houseman, E., Coull, B., & Betensky, R. (2006). Feature-specific penalized latent class analysis for genomic data. *Biometrics, 62,* 1062-1070. Retrieved from: www.jstor.org/stable/4124527.

Houseman, E., Marsti, C., Karagas, M., & Ryan, L. (2007). Penalized item response theory models: Application to epigenetic alterations in bladder cancer. *Biometrika, 63,* 1269-1277. Retrieved from: www.jstor.org/stable/4541483.

Hulin, C.L., Lissak, R.I., & Drasgow, F. (1982). Recover of two- and three-parameter logistic item characteristic curves: A Monte Carlos study. *Applied Psychological Measurement, 6,* 249-260.

Karagas, M., Tosteson, T., Morris, J., Demidenko, E., Mott, L., Heaney, J., & Schned, A. (2004). Incidence of transitional cell carcinoma of the bladder and arsenic in New Hampshire. *Cancer Causes and Control, 15,* 465-472. doi: 10.1023/B:CACO.0000036452.5

Kendall, M. & Stuart, A. (1973). *The Advance Theory of Statistics.* New York:

Oxford University Press.

van der Kooji, A. (2007). *Prediction accuracy and stability of regression with optimal scaling transformations.* Leiden: Department of Data Theory, Leiden University.

Little, R.J. & Rubin, D.B. (1983). On jointly estimating the parameters and missing data by maximizing the complete-data likelihood. *The American Statistician, 37,* 218-220.

Lord, F. (1952). A theory of test scores. *Psychometric Monograph, 7,* 1-10.

Lord, F. (1968). An analysis of the verbal scholastic aptitude test using Birnbaum's three-parameter logistic model. *Educational and Psychological Measurement, 28,* 989-1020. doi: 10.1177/001316446802800401.

Lord, F. & Novick, M. (1968). *Statistical Theory of Mental Test Scores.* Reading, MA: Addison-Wesley.

Maxwell, A.E. (1959). Maximum likelihood estimates of item parameters using the logistic functions. *Psychometrika, 24,* 221-227.

McDonald, R. (1999). *Test Theory: A Unified Treatment.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Mislevy, R. (1986). Bayes modal estimation in item response models. *Psychometrika, 51,* 177-195. doi: 10.1007/BF02293979.

Mislevy, R. & Bock, R. (1984). *BILOG Version 2.2: Item Analysis and Test Scoring with Binary Logistic Models.* Mooresville, IN: Scientific Software.

Neyman, J. & Scott, E.L. (1948). Consistent estimates based on partially consistent observations. *Econometrika, 16,* 1-32. Retrieved from: http://www.jstor.org/stable/1914288.

Nyquist, H. (1991). Restricted estimation of generalized linear models. *Journal of Applied Statistics, 40,* 133-141.

Paolino, J. (2013). *Penalized Joint Maximum Likelihood Estimation Applied to the Two Parameter Logistic Item Response Model* (Unpublished Doctoral Thesis). Columbia University, New York.

R Core Development Team. (2013). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. Retrieved from http://www.R-project.org/.

Ramsay, J.O. (1991). Kernel smoothing approaches to nonparametric item characteristic curve estimations. *Psychometrika, 56*, 611-630.

Rasch, G. (1960). *Probabilistic Models for Some Intelligence and Attainment Tests.* Chicago, IL: University of Chicago Press.

Rasch, G. (1966). An item analysis which takes individual differences into account. *British Journal of Mathematical and Statistical Psychology, 19,* 49-57.

Richardson, M.W. (1936). The relationship between difficulty and differential validity of a test. *Psychometrika, 1,* 33-49.

Rizopoulos, D. (2006). ltm: An r package for latent variable modeling and item response theory analysis. *Journal of Statistical Software, 17,* 1-25. Retrieved from http://www.jstatsoft.com/.

Ross, J. (1966). An empirical study of the logistic mental test model. *Psychometrika, 31,* 325-340.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics, 6*, 461-464.

Shanno, D. (1970). Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation, 24,* 647-656.

Shea, T., Tennant, A., & Pallant, J. (2009). Rasch model analysis of the Depression, Anxiety, and Stress Scales (DASS). *BMC Psychiatry, 9,* 21. doi: 10.1186/1471-244X-9-21.

Spergel, D. & Curry, G. (2005). Studying youth gangs: Alternative method and conclusions. *Journal of Contemporary Criminal Justice, 21,* 98-119.

Stan Development Team. (2014). Stan Modeling Language: User's Guide and Reference Manual (2.2.0). Retrieved from file:///C:/Users/Chris/Downloads/stan-

reference-2.2.0.pdf.

Stewart, J. (2010). *Calculus: Early Transcendentals* (7th Ed.). Independence, KY: Cengage Learning.

Swaminathan, H. & Gifford, J.A. (1982). Bayesian estimation in the Rasch model. *Journal of Educational Statistics, 7,* 175-191. Retrieved from: http://links.jstor.org/sici?sici=03 9791%28198223%297 -170%3A3%3C175%3ABEITRM%3E2.0.CO%3B2-1.

Swaminathan, H. & Gifford, J.A. (1985). Bayesian estimation in the two parameter logistic model. *Psychometrika, 50,* 349-364. doi:: 10.1007/BF02294110.

Swaminathan, H. & Gifford, J.A. (1986). Bayesian estimation in the three parameter logistic model. *Psychometrika, 51,* 589-601. doi: 10.1007/BF02295598.

Thissen, D. & Wainer, H. (1982). Some standard errors in item response theory. *Psychometrika, 47,* 397-412. doi: 10.1007/BF02293705.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B, 58,* 267-288. Retrieved from: www.jstor.org/stable/2346178.

Tibshirani, R. (1997). The lasso method for variable selection in the Cox model. *Statistics in Medicine, 16,* 385-395. doi: 10.1002/(SICI)1097-0258(19970228)16:4<385::AID-SIM380>3.0.CO;2-3.

Tucker, L.R. (1946). Maximum validity of a test with equivalent items. *Psychometrika, 11,* 1-13.

Tutz, G. & Schauberger, G. (2015). A penalty approach to differential item functioning in Rasch models. *Psychometrika, 80,* 21-43.

Waller, N.G. & Reise, S. (2010). Measuring psychopathology with non-standard IRT models: Fitting the four-parameter model to the MMPI. In S. Embretson (Ed.), *Measuring Psychological Constructs with Model-Based Approaches* (pp. 147-173). American Psychological Association.

Weiss, D.J. (1982). Improving measurement quality and efficiency with adaptive testing. *Applied Psychological Measurement, 6,* 473–495.

Wood, R. L., & Lord, F. M. (1976). A user's guide to LOGIST. Research Memorandum, 76(4).

Wood, R. L., Wingersky, M. S., & Lord, F. M. (1976). Research Memorandum: Logist: a Computer Program for Estimating Examinee Ability and Item Characteristic Curve Parameters. Educational Testing Service.

Yen, W.M. (1987). A comparison of the efficiency and accuracy of BILOG and LOGIST. *Psychometrika, 52,* 275-291. doi: 10.1007/BF02294241.

Zellner, A. (1971). *An Introduction to Bayesian Inference in Econometrics.* New York: Wiley.

Zou, H. & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, 67,* 301-320. Retrieved from: www.jstor.org/stable/3647580.

Zumbo, B. (1999). *A handbook on the theory and methods of differential item functioning (DIF).* Ottawa: National Defense Headquarters.

A Appendix A: Derivation of the First Partial Derivatives of the 2PL

Let the probability of examinee i answering item j correctly be

$$P_j^{\star}(\theta_i) = \frac{\exp(\gamma_j + \beta_j\theta_i)}{1 + \exp(\gamma_j + \beta_j\theta_i)}$$

and the probability of examinee i answering item j incorrectly be

$$\begin{aligned} Q_j^{\star}(\theta_i) &= 1 - P_j^{\star}(\theta_i), \\ &= \frac{1}{1 + \exp(\gamma_j + \beta_j\theta_i)}. \end{aligned}$$

We begin by taking the first partial derivative of $P_j^{\star}(\theta_i)$ and $Q_j^{\star}(\theta_i)$ with respect to the i^{th} latent trait parameter, θ_i. By quotient rule, the first partial derivative of $P_j^{\star}(\theta_i)$ is,

$$\begin{aligned} \frac{\partial}{\partial\theta_i}P_j^{\star}(\theta_i) &= \frac{\partial}{\partial\theta_i}\frac{\exp(\gamma_j + \beta_j\theta_i)}{1 + \exp(\gamma_j + \beta_j\theta_i)}, \\ &= \frac{\frac{\partial \exp(\gamma_j+\beta_j\theta_i)}{\partial\theta_i}\left[1 + \exp(\gamma_j + \beta_j\theta_i)\right] - \exp(\gamma_j + \beta_j\theta_i)\frac{\partial[1+\exp(\gamma_j+\beta_j\theta_i)]}{\partial\theta_i}}{\left[1 + \exp(\gamma_j + \beta_j\theta_i)\right]^2}, \\ &= \frac{\beta_j \exp(\gamma_j + \beta_j\theta_i)\left[1 + \exp(\gamma_j + \beta_j\theta_i)\right] - \beta_j \exp(\gamma_j + \beta_j\theta_i)^2}{\left[1 + \exp(\gamma_j + \beta_j\theta_i)\right]^2}, \\ &= \beta_j\frac{\exp(\gamma_j + \beta_j\theta_i) + \exp(\gamma_j + \beta_j\theta_i)^2 - \exp(\gamma_j + \beta_j\theta_i)^2}{\left[1 + \exp(\gamma_j + \beta_j\theta_i)\right]^2}, \\ &= \beta_j\frac{\exp(\gamma_j + \beta_j\theta_i)}{\left[1 + \exp(\gamma_j + \beta_j\theta_i)\right]^2}, \\ &= \beta_j P_j^{\star}(\theta_i)\, Q_j^{\star}(\theta_i). \end{aligned}$$

Similarly, for $Q_j^\star(\theta_i)$ we have

$$\begin{aligned}
\frac{\partial}{\partial\theta_i}Q_j^\star(\theta_i) &= \frac{\partial}{\partial\theta_i}\frac{1}{1+\exp(\gamma_j+\beta_j\theta_i)}, \\
&= \frac{-\frac{\partial}{\partial\theta_i}\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]}{\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]^2}, \\
&= -\beta_j\frac{\exp(\gamma_j+\beta_j\theta_i)}{\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]^2}, \\
&= -\beta_j P_j^\star(\theta_i)\,Q_j^\star(\theta_i).
\end{aligned}$$

The first partial derivative of $P_j^\star(\theta_i)$ with respect the the intercept parameter γ_j is,

$$\begin{aligned}
\frac{\partial}{\partial\gamma_j}P_j^\star(\theta_i) &= \frac{\partial}{\partial\gamma_j}\frac{\exp(\gamma_j+\beta_j\theta_i)}{1+\exp(\gamma_j+\beta_j\theta_i)}, \\
&= \frac{\frac{\partial\exp(\gamma_j+\beta_j\theta_i)}{\partial\gamma_j}\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]-\exp(\gamma_j+\beta_j\theta_i)\frac{\partial[1+\exp(\gamma_j+\beta_j\theta_i)]}{\partial\gamma_j}}{\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]^2}, \\
&= \frac{\exp(\gamma_j+\beta_j\theta_i)\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]-\exp(\gamma_j+\beta_j\theta_i)^2}{\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]^2}, \\
&= \frac{\exp(\gamma_j+\beta_j\theta_i)+\exp(\gamma_j+\beta_j\theta_i)^2-\exp(\gamma_j+\beta_j\theta_i)^2}{\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]^2}, \\
&= \frac{\exp(\gamma_j+\beta_j\theta_i)}{\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]^2}, \\
&= P_j^\star(\theta_i)\,Q_j^\star(\theta_i),
\end{aligned}$$

and the first partial derivative of $Q_j^\star(\theta_i)$ with respect to γ_j is

$$\begin{aligned}
\frac{\partial}{\partial\gamma_j}Q_j^\star(\theta_i) &= \frac{\partial}{\partial\gamma_j}\frac{1}{1+\exp(\gamma_j+\beta_j\theta_i)}, \\
&= \frac{-\frac{\partial}{\partial\gamma_j}\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]}{\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]^2}, \\
&= -\frac{\exp(\gamma_j+\beta_j\theta_i)}{\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]^2}, \\
&= -P_j^\star(\theta_i)\,Q_j^\star(\theta_i).
\end{aligned}$$

Finally, the first partial derivative of $P_j^\star(\theta_i)$ with respect the the slope parameter β_j is,

$$\begin{aligned}
\frac{\partial}{\partial\beta_j}P_j^\star(\theta_i) &= \frac{\partial}{\partial\beta_j}\frac{\exp(\gamma_j+\beta_j\theta_i)}{1+\exp(\gamma_j+\beta_j\theta_i)},\\
&= \frac{\frac{\partial\exp(\gamma_j+\beta_j\theta_i)}{\partial\beta_j}\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]-\exp(\gamma_j+\beta_j\theta_i)\frac{\partial[1+\exp(\gamma_j+\beta_j\theta_i)]}{\partial\beta_j}}{\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]^2},\\
&= \frac{\theta_i\exp(\gamma_j+\beta_j\theta_i)\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]-\theta_i\exp(\gamma_j+\beta_j\theta_i)^2}{\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]^2},\\
&= \theta_i\frac{\exp(\gamma_j+\beta_j\theta_i)+\exp(\gamma_j+\beta_j\theta_i)^2-\exp(\gamma_j+\beta_j\theta_i)^2}{\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]^2},\\
&= \theta_i\frac{\exp(\gamma_j+\beta_j\theta_i)}{\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]^2},\\
&= \theta_i P_j^\star(\theta_i)\,Q_j^\star(\theta_i),
\end{aligned}$$

and the first partial derivative of $Q_j^\star(\theta_i)$ with respect to β_j is,

$$\begin{aligned}
\frac{\partial}{\partial\beta_j}Q_j^\star(\theta_i) &= \frac{\partial}{\partial\beta_j}\frac{1}{1+\exp(\gamma_j+\beta_j\theta_i)},\\
&= \frac{-\frac{\partial}{\partial\beta_j}\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]}{\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]^2},\\
&= -\frac{\theta_i\exp(\gamma_j+\beta_j\theta_i)}{\left[1+\exp(\gamma_j+\beta_j\theta_i)\right]^2},\\
&= -\theta_i P_j^\star(\theta_i)\,Q_j^\star(\theta_i).
\end{aligned}$$

B Appendix B: R Code for Regularized Marginal Maximum Likelihood

```
RMML = function(Y,D=1.702,tol=1e-7,lambda=NULL,alpha=0.5,maxOptimCycle=10,
                maxEMCycle=100,quadLim=c(-4,4),quadLen=100,quadPar=c(0,1),
                stopDiff=0.01,verbose=TRUE,lambdaStep=0.1){


     # Function to compute the probability of a correct response
     cmpProb <- function(ip,x=NULL,D=1.702,tol=1e-7,xlim=c(-4,4),
theta.length=100,irt.param=FALSE){
        if(length(ip)%%2!=0){
stop("Each item must have exactly 2 parameters")
}
        if(!is.matrix(ip)) ip <- matrix(ip,ncol=2)
        if(is.null(x)){
x <- seq(from=xlim[1],to=xlim[2],length.out=theta.length)
}

        num.item <- nrow(ip)
        num.exam <- length(x)
        if(irt.param) ip[,1] <- -1*ip[,1]*ip[,2]

        p <- matrix(99,nrow=num.exam,ncol=num.item)
        for(i in 1:num.item) p[,i] <- 1/(1+exp(-D*(ip[i,1]+ip[i,2]*x)))
        ind <- which(p>1-tol)
        if(length(ind)>0) p[ind] <- 1-tol
        ind <- which(p<tol)
```

```
        if(length(ind)>0) p[ind] <- tol

        return(p)
    }
    # Function to compute the penalized log likelihood
    penalizedLL <- function(ip,x=NULL,f=NULL,r,lambda,alpha,D=1.702,
tol=1e-7){
        if(!is.matrix(ip)&length(ip)%%2!=0|is.matrix(ip)&ncol(ip)!=2){
            stop("All items must be specified by 2 parameters")
        }
        if(!is.matrix(ip)) ip <- matrix(ip,ncol=2)

        if(!is.matrix(r)){
stop("Responses must be supplied in matrix form")
}
        if(ncol(r)!=nrow(ip)){
stop("Responses to some items are missing")
}

        if(is.null(f)) f <- rep(1,times=nrow(r))
        if(nrow(r)!=length(f)){
stop("Responses are missing for some examinees")
}

        probMat <- cmpProb(ip,x,D,tol)

        penalty <- lambda*(alpha*sum(abs(ip[,2]))+
```

```
(1-alpha)*sum((ip[,2])^2)))
        logLik <- -1*(sum(r*log(probMat)+(f-r)*log(1-probMat))- penalty

        return(logLik)
    }

    cmpSV <- function(resp,start.val=NULL,verbose=FALSE){
        p <- ncol(resp)
        n <- nrow(resp)
        cs <- colSums(resp)
        if(verbose&&any(cs==0)||any(cs==n)){
            warning("Some items do not have mixed response patterns.")
        {
        rndStrVal <- length(start.val)==1&&start.val=="random"
        if(rndStrVal){
             Z <- data.frame(z1=rnorm(n))
        }
        if(!rndStrVal){
             rs <- as.vector(rowSums(resp,na.rm=TRUE))
             len.unique <- length(unique(rs))
             rs <- factor(rs,labels=1:len.unique)
             rs <- as.numeric(levels(rs))[as.integer(rs)]
             Z <- data.frame(z1=seq(-3,3,length.out=len.unique)[rs])
        }
        form. <- as.formula("y~z1")
        old <- options(warn=(2))
        on.exit(options(old))
```

```
        coefs <- matrix(0,p,2)
        for(i in 1:p){
           Z$y <- resp[,i]
           fm <- try(glm(form.,family=binomial(),data=Z),silent=TRUE)
           if(!inherits(fm,"try-error")){
               coefs[i,] <- fm$coef
           }else{
               coefs[i,] <- c(0,1)
           }
        }
        if(any(cs==0)||any(cs==n)){
           coefs[which(cs==0),] <- c(0,1)
           coefs[which(cs==n),] <- c(0,1)
        }
        dimnames(coefs) <- NULL
        return(coefs)
    }

    n.items = ncol(Y) # test length
    n.exams = nrow(Y) # sample size
    if(verbose){
       cat(paste("Number of Items:",n.items,
"\nNumber of Examinees:",n.exams))
    }

    make.lambda.seq = is.null(lambda)
    if(is.null(lambda)){lambda = 0}
```

```
    nmrp = NULL
    warn = NULL
    cs = apply(Y,2,sum)
    if(any(cs==0)|any(cs==n.exams)){
        nmrp = c(which(cs==0),which(cs==n.exams))
        warn = paste("Items",nmrp,
"omitted due to non-mixed response patterns.")
        temp = Y[,nmrp]
        Y = Y[,-nmrp]
        nmrp = temp
        rm(temp)
    }
    if(verbose){
cat(paste("Number of non-mixed response patterns:",
if(!is.null(ncol(nmrp))){ncol(nmrp)}else{0},"\n"))
    }

    Y. = 1-Y
    startVal = cmpSV(Y) # compute starting estimates

    quadNodes = seq(quadLim[1],quadLim[2],length.out=quadLen)
    quadWeights = dnorm(quadNodes)
estCoef = oldCoef = startVal
    nextLambda = TRUE

    allSlope = allInter = oldAIC = newAIC = NULL
    while(nextLambda){
```

```
if(verbose){cat(paste("Lambda =",lambda,"\n"))}
numIter = 0
stopEM = FALSE
llold = NULL
while(!stopEM){
        numIter = numIter + 1
        if(verbose){cat(paste("EM Iteration:",numIter,"\n"))}
        if(verbose){cat("Beginning E-Step\n")}
        pMat = cmpProb(estCoef,quadNodes,D,tol)
        qMat = 1 - pMat
        LMat = exp(Y%*%t(log(pMat))+Y.%*%t(log(qMat)))
        LA = postProb = matrix(NA,nrow(LMat),ncol(LMat))
        for(k in 1:nrow(LMat)){LA[k,] = LMat[k,]*quadWeights}
        rs = apply(LA,1,sum)
        for(k in 1:ncol(LMat)){postProb[,k] = LA[,k]/rs}
        expectedF = apply(postProb,2,sum)
        expectedR = t(t(Y)%*%postProb)
        if(verbose){cat("Beginning M-Step\n")}
        if(!is.numeric(lambda)){stop("Invalid lambda value.")}
        if(!is.numeric(alpha)){stop("Invalid alpha value.")}
        if(alpha > 1 | alpha < 0){stop("Invalid alpha value.")}
        optimConverge = 2
        optimCycle = 0
        while(optimConverge!=0 & optimConverge!=1){
            fit = optim(par=c(estCoef),fn=penalizedLL,x=quadNodes,
                        f=expectedF,r=expectedR,lambda=lambda,
                        alpha=alpha,D=D,tol=tol,
```

```
                                method="L-BFGS-B",
                                lower=c(rep(quadLim[1],times=n.items),
                                rep(lower.disc,times=n.items))
                    estCoef = matrix(fit$par,ncol=2)
                    optimConverge = fit$convergence
                    if(optimCycle==maxOptimCycle&optimConverge!=0){
optimConverge = 1
}
                    optimCycle = optimCycle + 1
                }
                if(verbose){cat("Checking Convergence\n")}
                llNew = fit$value
                if(!is.null(llOld)){
                  if(abs(llNew - llOld)<stopDiff){
                      stopEM = TRUE
                      EMConv = 0
                  }
                }
                if(all(abs(oldCoef - estCoef)<stopDiff)){
                    stopEM = TRUE
                    EMConv = 0
                }
                if(numIter==maxEMCycle){
                    stopEM = TRUE
                    EMConv = 1
                    warning(paste("EM Algorithm did not converge after",
                            maxEMCycle,"Cycles."))
```

```
            }
            oldCoef = estCoef
            llOld = llNew
        }
        ipEst = cbind(estCoef[,2],-estCoef[,1]*estCoef[,2])
        ipEst[ipEst[,2]>quadLim[2],2] = quadLim[2]
        ipEst[ipEst[,2]<quadLim[1],2] = quadLim[1]
        estCoef = cbind(-ipEst[,1]*ipEst[,2],ipEst[,1])

        allSlope = as.matrix(cbind(allSlope,estCoef[,2]))
        allInter = as.matrix(cbind(allInter,estCoef[,1]))
        colnames(allSlope) = if(ncol(allSlope)==1){
           lambda
        }else{
           c(colnames(allSlope)[colnames(allSlope)!=""],lambda)
        }
        colnames(allInter) = if(ncol(allInter)==1){
           lambda
        }else{
           c(colnames(allInter)[colnames(allInter)!=""],lambda)
        }
        if(make.lambda.seq){
            newAIC = -2*fit$value+2*length(estCoef)
            if(verbose){
cat(paste(if(!is.null(oldAIC)){
round(oldAIC,2)
}else{
```

```
NA
},";",round(newAIC,2),"\n"))
                }
                if(lambda>0&!is.null(oldAIC)){
                    if(newAIC>oldAIC){nextLambda = FALSE}
                    oldAIC = newAIC
                }else{
                   oldAIC = newAIC
                }
                lambda = lambda + lambdaStep
            }else{
                nextLambda = FALSE
            }
        }
        return(list(lambda=as.numeric(colnames(allSlope)),alpha=alpha,
                    convergence=EMConv,nmrp=nmrp,warnings=warn,
                    slope=allSlope,intercept=allInter,
log.lh=fit$value))
}
```

C Appendix C: STAN Script

```
set.seed(123)
library(rstan)
library(coda)
library(xlsx)
### Bayesian models
twoPLlognorm = '
data{
int<lower=0> Nsubj;
int<lower=0> Nitems;
int<lower=0,upper=1> r[Nsubj,Nitems];
}
parameters{
real theta[Nsubj];
real<lower=0> a[Nitems];
real b[Nitems];
}
model{
for (j in 1 : Nsubj)
for (k in 1 : Nitems)
r[j,k] ~ bernoulli(inv_logit(a[k]*(theta[j]-b[k])));
for (j in 1 : Nsubj)
theta[j] ~ normal(0,1);
for(k in 1 : Nitems){
a[k] ~ lognormal(0,.25);
b[k] ~ uniform(-3,3);
}
```

```
}
'
twoPLuniform = '
data{
int<lower=0> Nsubj;
int<lower=0> Nitems;
int<lower=0,upper=1> r[Nsubj,Nitems];
}
parameters{
real theta[Nsubj];
real<lower=0> a[Nitems];
real b[Nitems];
}
model{
for (j in 1 : Nsubj)
for (k in 1 : Nitems)  r[j,k] ~ bernoulli(inv_logit(a[k]*(theta[j]-b[k])));
for (j in 1 : Nsubj) theta[j] ~ normal(0,1);
for(k in 1 : Nitems){
a[k] ~ uniform(0.6,1.9); b[k] ~ uniform(-3,3);
}
}
'
load("Data Sets/Response Matrices.rdat")
bayes.est <- NULL
for(i in 1:length(responses)){
  # i <- 1
  cat(paste("Condition",i,"\n"))
```

```
 y <- responses[[i]]
 y <- as.matrix(y)
 n.exam <- nrow(y)
 n.item <- ncol(y)
 twoPData <- list(Nsubj=n.exam,Nitems=n.item,r=y)
 bayes.mod.1 = stan(model_code=twoPLlognorm,data=twoPData, iter=1000,
chains=2,init=list(list(a=rep(1,times=n.item),
b=rep(0,times=n.item),
                                    theta=rep(0,times=n.exam)),
                               list(a=rep(1,times=n.item),
b=rep(0,times=n.item),
                                    theta=rep(0,times=n.exam))))
 bayes.mod.2 = stan(model_code=twoPLuniform,data=twoPData, iter=1000,
chains=2,init=list(list(a=rep(1,times=n.item),
b=rep(0,times=n.item),
                                        theta=rep(0,times=n.exam)),
                                   list(a=rep(1,times=n.item),
b=rep(0,times=n.item),
                                        theta=rep(0,times=n.exam))))

 ## Format the data for coda samples
 samp = extract(bayes.mod.1,pars=c("a","b"),permuted=FALSE,
inc_warmup=FALSE)
 a1 = samp[,1,1:n.item]
 a2 = samp[,2,1:n.item]
 b1 = samp[,1,(n.item+1):(2*n.item)]
 b2 = samp[,2,(n.item+1):(2*n.item)]
```

```
n.iter = length(a1) / n.item
var.names = c(paste("a",1:n.item,sep=""),paste("b",1:n.item,sep=""))
samp = structure(list(structure(c(a1,b1),
                .Dim=c(n.iter,2*n.item),.Dimnames=list(NULL,var.names),
                mcpar=c(1,n.iter,1),class="mcmc"),
                structure(c(a2,b2),
                .Dim=c(n.iter,2*n.item),.Dimnames=list(NULL,var.names),
                mcpar=c(1,n.iter,1),class="mcmc"))
                ,class="mcmc.list")

## Analyze
mod.summ = summary(samp)

## Means
gc()
rm()
s1 = matrix(unlist(mod.summ[1]),ncol=4)
a.mean = s1[1:n.item,1]
b.mean = s1[(n.item+1):(2*n.item),1]
bayes.means = cbind(a.mean,b.mean)
colnames(bayes.means) = c("a.hat.mean","b.hat.mean")
rownames(bayes.means) = paste("Item ",1:n.item,sep="")

## Medians
s2 = matrix(unlist(mod.summ[2]),ncol=5)
a.med = s2[1:n.item,3]
b.med = s2[(n.item+1):(2*n.item),3]
```

```
 bayes.med = cbind(a.med,b.med)
 colnames(bayes.med) = c("a.hat.median","b.hat.median")
 rownames(bayes.med) = paste("Item ",1:n.item,sep="")
 bayes.est <- rbind(bayes.est,cbind(rep(i,nrow(bayes.means)),
bayes.means,bayes.med))
 colnames(bayes.est) <- c("Condition","a.hat.mean","b.hat.mean",
                          "a.hat.median","b.hat.median")

 ## Format the data for coda samples
 samp = extract(bayes.mod.2,pars=c("a","b"),permuted=FALSE,
inc_warmup=FALSE)
 a1 = samp[,1,1:n.item]
 a2 = samp[,2,1:n.item]
 b1 = samp[,1,(n.item+1):(2*n.item)]
 b2 = samp[,2,(n.item+1):(2*n.item)]
 n.iter = length(a1) / n.item
 var.names = c(paste("a",1:n.item,sep=""),paste("b",1:n.item,sep=""))
 samp = structure(list(structure(c(a1,b1),
                  .Dim=c(n.iter,2*n.item),.Dimnames=list(NULL,var.names),
  mcpar=c(1,n.iter,1),class="mcmc"),
                  structure(c(a2,b2),
                  .Dim=c(n.iter,2*n.item),.Dimnames=list(NULL,var.names),
                  mcpar=c(1,n.iter,1),class="mcmc"))
                  ,class="mcmc.list")

 ## Analyze
 mod.summ = summary(samp)
```

```
  ## Means
  gc()
  rm()
  s1 = matrix(unlist(mod.summ[1]),ncol=4)
  a.mean = s1[1:n.item,1]
  b.mean = s1[(n.item+1):(2*n.item),1]
  bayes.means = cbind(a.mean,b.mean)
  colnames(bayes.means) = c("a.hat.mean","b.hat.mean")
  rownames(bayes.means) = paste("Item ",1:n.item,sep="")

  ## Medians
  s2 = matrix(unlist(mod.summ[2]),ncol=5)
  a.med = s2[1:n.item,3]
  b.med = s2[(n.item+1):(2*n.item),3]
  bayes.med = cbind(a.med,b.med)
  colnames(bayes.med) = c("a.hat.median","b.hat.median")
  rownames(bayes.med) = paste("Item ",1:n.item,sep="")
  bayes.est <- rbind(bayes.est,cbind(rep(i,nrow(bayes.means)),
 bayes.means,bayes.med))
  colnames(bayes.est) <- c("Condition","a.hat.mean","b.hat.mean",
                           "a.hat.median","b.hat.median")
}
write.csv(bayes.est,file="Results/Bayesian Estimates.csv",row.names=FALSE)
```

D Appendix D: Simulation 1 Item Parameter Values

Sample Size	Test Length	True a_j	True b_j
100	15	1.21	-0.08
100	15	1.65	0.86
100	15	0.92	0.34
100	15	1.66	-0.58
100	15	0.57	0.79
100	15	1.21	-0.69
100	15	0.72	-1.18
100	15	0.82	1.27
100	15	0.64	-0.31
100	15	0.99	0.03
100	15	1.71	-1.48
100	15	0.64	-1.13
100	15	1.24	-1.76
100	15	0.65	-1.06
100	15	1.03	-1.34
200	15	1.25	0.08
200	15	0.79	-1.62
200	15	0.97	0.44
200	15	0.80	-1.20
200	15	1.12	0.26
200	15	1.05	1.41
200	15	0.89	0.73
200	15	1.45	-0.19
200	15	0.49	-0.57

200	15	1.05	-0.01
200	15	0.79	1.48
200	15	1.13	-0.29
200	15	1.02	-0.07
200	15	1.03	0.30
200	15	1.77	1.32
500	15	1.24	0.35
500	15	0.77	0.32
500	15	1.20	0.43
500	15	1.22	-0.06
500	15	1.05	-0.90
500	15	0.92	-2.39
500	15	1.19	-1.07
500	15	1.38	0.02
500	15	1.11	-1.11
500	15	0.63	0.94
500	15	0.95	-0.91
500	15	0.72	-0.43
500	15	1.41	-0.23
500	15	1.05	-0.17
500	15	1.20	1.05
1000	15	1.62	-0.99
1000	15	0.84	1.36
1000	15	1.12	0.96
1000	15	0.53	-1.50
1000	15	1.09	0.55
1000	15	1.77	-1.89

1000	15	0.82	-0.46
1000	15	1.09	0.08
1000	15	0.87	0.19
1000	15	0.82	-1.80
1000	15	1.04	0.39
1000	15	0.81	0.49
1000	15	1.23	0.71
1000	15	0.70	-0.73
1000	15	1.53	-0.51
100	25	1.21	-1.48
100	25	1.65	-1.13
100	25	0.92	-1.76
100	25	1.66	-1.06
100	25	0.57	-1.34
100	25	1.21	0.75
100	25	0.72	-0.64
100	25	0.82	1.43
100	25	0.64	-0.62
100	25	0.99	0.23
100	25	1.71	0.26
100	25	0.64	0.43
100	25	1.24	-1.48
100	25	0.65	0.18
100	25	1.03	2.06
100	25	0.98	0.73
100	25	1.24	-0.49
100	25	1.09	-0.48

100	25	0.86	1.85
100	25	1.22	0.52
100	25	0.84	-1.11
100	25	0.74	-0.31
100	25	1.37	0.09
100	25	0.92	0.64
100	25	1.01	0.37
200	25	1.13	-0.92
200	25	1.34	0.32
200	25	1.03	0.44
200	25	0.87	-1.73
200	25	0.90	-1.47
200	25	1.28	1.07
200	25	1.02	-0.66
200	25	0.99	0.53
200	25	0.94	1.50
200	25	1.09	0.33
200	25	0.95	1.10
200	25	0.85	0.74
200	25	1.59	-1.26
200	25	1.10	-0.86
200	25	1.55	-1.38
200	25	0.59	0.05
200	25	0.81	-0.29
200	25	0.64	-0.85
200	25	0.90	-0.46
200	25	0.48	-1.73

200	25	0.69	0.26
200	25	0.88	-1.02
200	25	1.11	1.77
200	25	1.24	0.90
200	25	1.11	-1.16
500	25	0.79	-0.46
500	25	0.88	1.10
500	25	1.08	1.30
500	25	1.18	0.90
500	25	0.84	0.62
500	25	1.05	-0.35
500	25	1.43	-1.88
500	25	1.42	-1.40
500	25	0.95	-0.41
500	25	0.58	0.65
500	25	0.92	-0.19
500	25	0.81	0.28
500	25	1.00	1.76
500	25	0.77	-1.72
500	25	1.13	-0.14
500	25	1.14	-0.41
500	25	1.19	0.51
500	25	0.94	-0.44
500	25	1.24	0.23
500	25	1.11	-0.59
500	25	0.95	-0.59
500	25	1.20	-1.99

500	25	1.38	0.24
500	25	0.90	1.09
500	25	1.32	1.74
1000	25	1.03	0.43
1000	25	0.89	1.88
1000	25	0.96	1.25
1000	25	1.42	0.49
1000	25	0.77	-0.98
1000	25	0.80	0.62
1000	25	1.25	-0.79
1000	25	0.99	0.19
1000	25	0.82	-1.55
1000	25	0.70	-0.91
1000	25	0.72	0.48
1000	25	1.43	1.26
1000	25	1.20	1.93
1000	25	1.43	-0.37
1000	25	0.96	-0.86
1000	25	0.92	0.81
1000	25	1.10	0.24
1000	25	1.00	0.55
1000	25	0.69	0.36
1000	25	0.67	0.96
1000	25	1.26	0.48
1000	25	1.22	-0.69
1000	25	0.85	0.82
1000	25	1.46	-0.39

1000	25	1.13	-0.19
100	35	1.21	0.26
100	35	1.65	0.43
100	35	0.92	-1.48
100	35	1.66	0.18
100	35	0.57	2.06
100	35	1.21	0.73
100	35	0.72	-0.49
100	35	0.82	-0.48
100	35	0.64	1.85
100	35	0.99	0.52
100	35	1.71	-1.11
100	35	0.64	-0.31
100	35	1.24	0.09
100	35	0.65	0.64
100	35	1.03	0.37
100	35	0.98	0.20
100	35	1.24	1.18
100	35	1.09	-0.70
100	35	0.86	-0.61
100	35	1.22	-0.51
100	35	0.84	-0.82
100	35	0.74	-0.24
100	35	1.37	-0.77
100	35	0.92	0.72
100	35	1.01	1.28
100	35	0.69	-0.99

100	35	0.75	-0.91
100	35	0.64	-1.32
100	35	0.77	1.93
100	35	0.71	0.68
100	35	1.21	-0.40
100	35	0.85	0.19
100	35	1.43	0.40
100	35	0.86	1.64
100	35	1.06	-0.06
200	35	1.81	-0.25
200	35	1.13	-1.55
200	35	0.82	0.05
200	35	0.64	2.19
200	35	0.69	-0.64
200	35	0.95	-1.01
200	35	0.85	1.25
200	35	0.81	0.88
200	35	1.18	-0.26
200	35	1.34	-0.45
200	35	0.90	0.04
200	35	0.58	0.06
200	35	1.58	-1.08
200	35	1.46	-1.40
200	35	1.24	-0.23
200	35	1.08	1.56
200	35	0.69	0.08
200	35	1.22	-0.45

200	35	0.80	0.73
200	35	1.68	0.58
200	35	0.57	-0.84
200	35	1.02	0.79
200	35	0.67	-0.28
200	35	1.02	-0.37
200	35	0.58	-0.96
200	35	0.66	0.13
200	35	0.83	0.20
200	35	1.03	-2.52
200	35	0.86	-1.81
200	35	1.15	0.62
200	35	0.63	0.02
200	35	0.85	-1.40
200	35	0.62	1.51
200	35	1.06	2.23
200	35	0.93	0.45
500	35	1.21	0.52
500	35	0.81	-0.09
500	35	1.23	-1.26
500	35	0.94	0.02
500	35	0.81	1.19
500	35	1.15	1.56
500	35	0.85	-0.40
500	35	1.71	-1.60
500	35	0.88	0.47
500	35	1.02	1.00

500	35	1.05	0.72
500	35	1.13	-0.50
500	35	0.90	-0.45
500	35	1.24	2.74
500	35	1.29	-1.15
500	35	1.14	0.50
500	35	0.80	-1.73
500	35	0.85	0.53
500	35	0.78	-0.47
500	35	0.98	-0.84
500	35	1.00	0.16
500	35	0.81	1.59
500	35	1.25	0.04
500	35	0.77	1.51
500	35	1.15	-0.22
500	35	1.02	-0.01
500	35	0.91	0.10
500	35	1.38	0.34
500	35	0.77	-0.29
500	35	1.01	-0.45
500	35	1.10	-0.49
500	35	0.62	1.41
500	35	1.05	-1.20
500	35	1.44	1.26
500	35	0.81	2.22
1000	35	0.84	-0.61
1000	35	1.28	-1.78

1000	35	0.81	-0.10
1000	35	1.09	1.39
1000	35	1.14	-0.69
1000	35	0.75	-1.73
1000	35	1.29	-0.34
1000	35	1.36	-0.97
1000	35	0.85	0.78
1000	35	0.58	-0.43
1000	35	1.03	-1.20
1000	35	0.95	-0.84
1000	35	0.79	0.94
1000	35	1.22	1.14
1000	35	1.40	-2.03
1000	35	1.08	-0.48
1000	35	1.06	-1.62
1000	35	0.87	-3.22
1000	35	1.19	0.06
1000	35	0.70	-1.13
1000	35	0.79	-1.68
1000	35	0.57	-0.66
1000	35	0.87	1.13
1000	35	1.24	-1.82
1000	35	0.96	0.56
1000	35	0.82	0.38
1000	35	0.82	0.06
1000	35	1.15	0.06
1000	35	1.25	1.68

1000	35	1.27	2.03
1000	35	1.22	-1.03
1000	35	0.82	0.71
1000	35	1.13	-0.64
1000	35	0.80	0.37
1000	35	1.92	-1.19