

Running head: POSTERIOR PREDICTIVE CHECKING

Posterior Predictive Checking of Unidimensional Item Response Theory Models

Kyong Hee Chon, Yuki Nozawa, and Su Zhang

December 4, 2006

Abstract

This study applies the posterior predictive model checking (PPMC) method (Rubin, 1984) to assess the fit of unidimensional item response theory (IRT) models for binary responses, and examines the performance of several discrepancy measures for assessing different aspects of model misfit. One dataset was generated from the three parameter logistic (3PL) model, which was then fit with the one parameter logistic (1PL), two parameter logistic (2PL), and 3PL models. The performance of the discrepancy measures examined in this study suggests that different measures detect different aspects of model misfit and that the choice of measures depends on the context and the aspects of fit to be assessed.

Introduction

Item response theory (IRT) models are widely used for the analysis of items, tests, and examinees. The one-, two-, and three-parameter logistic models (1PL, 2PL, and 3PL) are most commonly used for unidimensional dichotomous item responses. Appropriate use of these IRT models requires that several strong assumptions be met by the data, such as local independence (i.e., for examinees with the same ability or proficiency, the probability of getting one item correct is independent of the probability of getting any other item correct), a specific form of the item response function (e.g., 1PL model assumes that all items have equal discriminations and no guessing). If these assumptions are not adequately met, inferences regarding the nature of the items and examinees can be erroneous, and the potential advantages of IRT are not gained. It is therefore crucial to check the adequacy of the fit of the chosen IRT model to item responses. Several fit statistics have been proposed within the frequentist framework (Orlando & Thissen, 2003; Yen, 1981), but none is universally accepted and model checking still remains an underdeveloped area in IRT.

In Bayesian framework, model fit can be checked by: (1) examining the sensitivity of inferences to reasonable changes in the prior distribution and the likelihood, (2) checking the sensibility of posterior inferences against one's substantive knowledge, and (3) checking the plausibility of posterior predictive replicated data against observed data, often referred to as posterior predictive model checking (PPMC) (Gelman, et al., 2003). In the IRT context, there have been several applications of the PPMC method (Albert & Ghosh, 2000; Glas & Meijer, 2003; Hoijtink, 2001; Hoijtink & Molenaar, 1997; Janssen, et al., 2000; Rubin & Stern, 1994; Scheines, Hoijtink & Boomsma, 1999; Sinharay, 2005; Sinharay & Johnson, 2003; Sinharay, Johnson & Stern, 2006; van Onna, 2003). The choice of discrepancy measures in PPMC is

crucial, and different measures tend to capture different aspects of model misfit; this theme is echoed again and again in these studies. Sinharay and Johnson (2003), for example, examined the power of four discrepancy measures in detecting five types of model misfits. The biserial correlation coefficient and the item pair odds ratio were both found to be effective in detecting the inadequacy of the 1PL model for data from 2PL and 3PL models, whereas when the 2PL model was fit to data from the 3PL model, the item pair odds ratio was not effective at all, with the biserial correlation coefficient still powerful.

Under the appeal of the Bayesian model checking tool in the IRT context, the purpose of this study is to 1) apply the PPMC method to assess the fit of unidimensional IRT models for dichotomous item responses, and 2) examine the performance of several discrepancy measures for assessing different aspects of model misfit.

Method

Posterior Predictive Model-Checking (PPMC) Method

The idea of PPMC is to generate simulated values from the posterior predictive distributions of replicated data and to compare these samples to the observed data. If the replicated data and the observed data differ systematically, it is an indication of a potential model misfit.

Letting y be the observed data and θ be the vector of all the parameters in the model, we then define $p(y|\theta)$ as the likelihood and $p(\theta)$ as the prior distribution on the parameters. The PPMC method suggests checking a model by examining whether the observed data y appear extreme with respect to the posterior predictive distribution of replicated data y^{rep} , which is obtained by

$$p(y^{rep} | y) = \int p(y^{rep} | \theta)p(\theta | y)d\theta. \quad (1)$$

A discrepancy measure or a test quantity, $T(y)$, is then defined and computed from the replicated data, which is compared with the observed values of $T(y)$ (i.e., computed from the observed data). The PPMC method allows a reasonable summary of such comparisons with the posterior predictive p-value (PPP-value):

$$\Pr(T(y^{rep}, \theta) \geq T(y, \theta) | y) = \int_{T(y^{rep}, \theta) \geq T(y, \theta)} p(y^{rep} | \theta) p(\theta | y) dy^{rep} d\theta. \quad (2)$$

PPP-values that are close to 0 or 1 are indicative of model misfits.

Data

One dataset was generated from the 3PL model, with the probability of getting item j correct for examinees with ability or proficiency θ represented as

$$P_j(\theta) = c_j + \frac{1 - c_j}{1 + \exp[-Da_j(\theta - b_j)]}, \quad (3)$$

where c_j is the pseudo-guessing (or lower asymptote) parameter, b_j is the item difficulty parameter, a_j is the item discrimination (or slope) parameter, and D is the scaling factor of 1.7. For the 1PL model, c_j is assumed to be zero and a_j is fixed as a constant for all items. For the 2PL model, c_j is assumed to be zero for all items and a_j is allowed to vary. The 3PL model can be considered equivalent to the Generalized Linear Mixed Model (GLMM) for binary responses with random intercepts and slopes.

The simulated dataset contains responses of 1000 examinees to 15 dichotomous items. The ability parameters were generated from a standard normal distribution. The item parameters used to generate the responses were based on the real item parameter estimates from the National Assessment in Educational Progress (NAEP), as provided in Sinharay, Johnson, and Stern (2006).

Analysis

The prior distributions on the item parameters are as follows: $\log(a_j) \stackrel{iid}{\sim} N(0,10)$, $\log(b_j) \stackrel{iid}{\sim} N(0,1)$, $\log(c_j) \stackrel{iid}{\sim} N(-1.4,1)$, which can all be considered noninformative priors. The data generated from the 3PL model were analyzed by the 1PL, 2PL, and 3PL models. For each analysis, due to the intensity of computations and the inefficiency of WinBUGS in this context, only one chain of length 6,000 was run, with the first 4,000 iterations burnt in and a thinning rate of 1. The WinBUGS code for fitting the 3PL model can be found in Appendix A.

Discrepancy Measures Considered

Based on the findings of Sinharay and Johnson (2003), Sinharay, Johnson, and Stern (2006), and von Schrader (personal communication, November 12, 2006), this study considered the following discrepancy measures.

Proportion correction item score. This refers to the proportion of examinees who get an item correct, or the item difficulty parameter, b_j . If a model fits the data well, the predicted proportion correct item scores should be close to the observed proportion correct item scores. However, because item difficulty is accounted for by all three analysis models, this measure is not expected to be effective in detecting model-data misfit. It was used in the study simply as a computational check—if this measure suggests misfit in any of the three cases, it is a strong indication that something is seriously wrong with the program code.

Point biserial correlation coefficient. The point biserial correlation coefficient, together with the biserial correlation coefficient, is used as a common index for item discrimination in classical test theory. It is closely related to the item discrimination parameter, a_j , in 2PL and

3PL models. Because a_j is fixed in the 1PL model, this statistic is expected to detect misfit when fitting the 1PL model to the 3PL data, but not when fitting the 2PL model.

Item pair odds ratio (OR). Let $n_{kk'}$ denote the number of examinees scoring k on the one item and k' on another item, with $k, k' = 0, \text{ or } 1$. The sample item pair OR has the form

$$OR = \frac{n_{11}n_{00}}{n_{01}n_{10}} \quad (4)$$

As noted by Chen and Thissen (1997), the OR can be helpful in detecting violation of the local independence assumption. The logic of using the OR is that if local independence is not satisfied, the item pair ORs from replicated datasets will be larger or smaller than those from the observed data.

There are 15 items in the original dataset, which results in 105 pairs of items. For every IRT model, 105 item pair ORs were obtained from each simulated posterior predictive replicated dataset, and then they were compared to the observed ORs. We consider PPP-values that are smaller than .25 or larger than .75 extreme values and indicative of model-data misfit.

Total score distribution. With 15 dichotomous items on the test and 1 point per item, the possible total scores for examinees are 0, 1, 2, ..., and 15. The total score distribution refers to the distribution of the number of examinees who get each of the 16 total scores. From a frequentist perspective, Hambleton and Han (2004) suggest measuring overall model fit by comparing observed and predicted total score distributions. This can be done easily within the Bayesian framework, since the predicted total score distribution, or a variation of it, can easily be obtained. In this study, from each posterior predictive replicated dataset, a total score distribution was obtained. Then from 2000 such replicated datasets (4000 out of 6000 iterations were burnt in), the median number of examinees who get each of the 16 total scores was plotted as the

predicted total score distribution, which was compared to the observed total score distribution to assess model fit.

Results

Fit of IPL Model

As expected, the results for proportion correct item scores did not detect model-data misfit. The PPP-values for the 15 items were within the range of .336 and .622. The point-biserial correlation measure was indeed very effective in detecting the misfit of the IPL model (see the PPP-values in Table 1):

Table 1

PPP-values for Point Biserial Correlation Coefficient by Fitting the IPL Model

Item	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PPP-values	1.00	0.00	0.00	0.22	0.59	1.00	0.20	0.95	0.20	1.00	0.57	1.00	0.02	0.00	0.00

As seen below in Table 2, out of the one hundred and five PPP-values for the item pair ORs, eighty-four were smaller than .25 or larger than .75 (those in the highlighted cells), which was clear evidence that the IPL model was not adequate for the data.

Table 2

PPP-values for Item Pair Odds Ratios by Fitting 1PL Model

Item ID	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.46	0.87	0.85	0.97	1.00	0.38	0.99	0.72	1.00	0.41	0.99	1.00	0.89	0.44
2	.	0.00	0.01	0.05	0.93	0.02	0.14	0.01	0.97	0.01	0.13	0.00	0.00	0.00
3	.	.	0.22	0.09	0.98	0.10	0.38	0.01	1.00	0.77	0.68	0.04	0.00	0.00
4	.	.	.	0.70	0.99	0.12	0.89	0.28	0.81	0.02	0.72	0.23	0.08	0.00
5	1.00	0.24	0.70	0.46	0.98	0.86	0.95	0.36	0.03	0.00
6	0.99	1.00	0.83	1.00	1.00	1.00	0.94	0.93	0.02
7	0.48	0.53	0.93	0.32	0.77	0.13	0.24	0.00
8	0.86	1.00	0.95	0.96	0.03	0.04	0.43
9	0.96	0.27	0.22	0.50	0.48	0.01
10	0.96	1.00	0.99	0.76	0.36
11	0.59	0.11	0.26	0.10
12	0.90	0.91	0.03
13	0.00	0.00
14	0.00

In Figure 1 are the total score distributions based on the observed responses and the predicted replicated datasets with the median number of examinees plotted for each total score. Overall, it indicates an inadequate model fit, especially for total scores between 7 and 12.

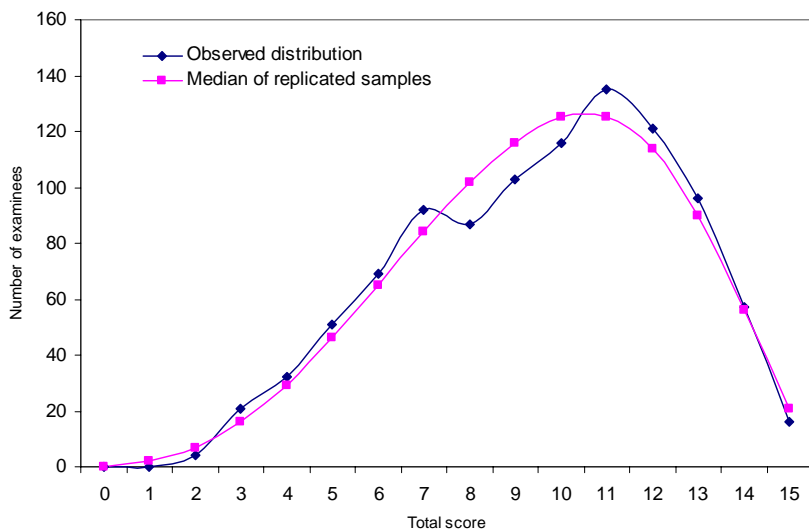


Figure 1. Total Score Distributions of Observed and Replicated Data by Fitting 1PL Model

Fit of 2PL Model

The 15 PPP-values for the measure of proportion correct item scores lie within a similar range to the previous case, which gives us confidence in the computations that were conducted. As expected, the point biserial correlation coefficient was not a useful measure when detecting the misfit of the 2PL model, with the PPP-values within the range of .295 and .583.

Out of the 105 PPP-values for the item pair ORs, only 30 were extreme values (see Table 3), which suggests that this measure was not effective when fitting the 2PL model. This finding is consistent with that in Sinharay and Johnson (2003), where the item pair OR was found to have low power when it was used to detect misfit of 2PL model to data from the 3PL model.

Table 3

PPP-values for Item Pair Odds Ratios by Fitting 2PL Model

Item ID	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.35	0.51	0.40	0.55	0.47	0.15	0.61	0.18	0.79	0.14	0.40	0.92	0.58	0.47
2	.	0.38	0.49	0.66	0.65	0.52	0.54	0.54	0.59	0.15	0.29	0.16	0.63	0.65
3		.	0.63	0.40	0.56	0.48	0.49	0.16	0.83	0.88	0.54	0.63	0.33	0.37
4			.	0.70	0.78	0.23	0.76	0.37	0.19	0.03	0.39	0.51	0.40	0.40
5				.	0.72	0.33	0.36	0.42	0.27	0.82	0.61	0.59	0.22	0.30
6					.	0.88	0.46	0.13	0.34	0.98	0.57	0.40	0.41	0.03
7						.	0.42	0.69	0.48	0.41	0.58	0.43	0.66	0.41
8							.	0.63	0.37	0.87	0.48	0.04	0.10	0.92
9								.	0.27	0.27	0.04	0.75	0.83	0.72
10									.	0.57	0.90	0.68	0.10	0.14
11										.	0.30	0.23	0.49	0.66
12											.	0.72	0.80	0.31
13												.	0.29	0.61
14													.	0.48

From the total score distributions in Figure 2, it seems that the 2PL model fit slightly better than the 1PL model, but for total scores in the range of 6 and 10 there were still obvious discrepancies between the two distributions.

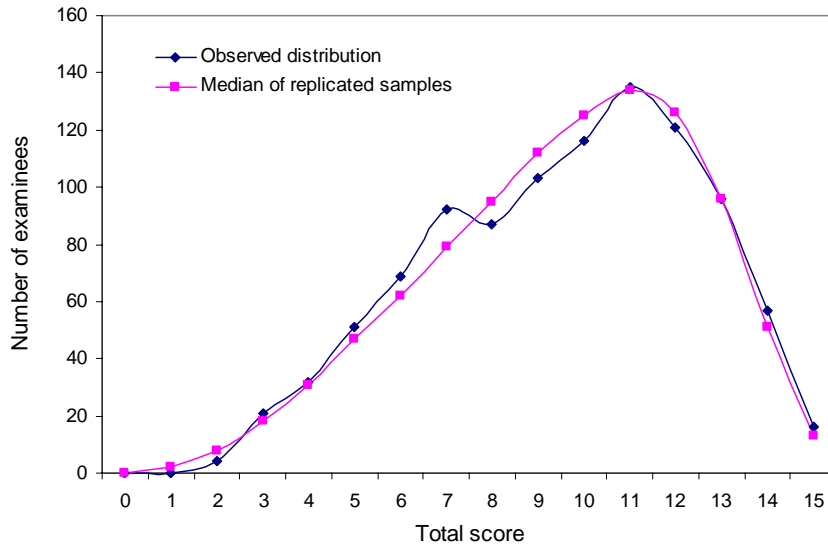


Figure 2. Total Score Distributions of Observed and Replicated Data by Fitting 2PL Model

Fit of 3PL Model

In this case, if a discrepancy measure is effective, it is expected not to indicate model-data misfit unless a Type I error has been made (given the null hypothesis that the model fits), since the 3PL model was fit to the data generated from the 3PL model. The measure of proportion correct item scores indicates that the model fit the data well. The PPP-values for the point biserial correlation coefficients were between .339 and .638 for the first 13 items, but were .243 and .184 for the last two items, which is indicative of model misfit. It could be that Type I errors have been made for these two items. However, if the original item parameters for generating the 3PL response data were closely examined, it is clear that these two items have unusual discrimination parameters: 2.01 and 2.40, respectively. They are so unusual that even the 3PL model cannot adequately account for the item responses.

In Table 4, 33 out of the 105 PPP-values for the item pair ORs were extreme values, which could be either due to Type I errors or simply an indication that the item pair OR was not a very effective discrepancy measure in this context.

Table 4

PPP-values for Item Pair Odds Ratios by Fitting 3PL Model

Item ID	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.27	0.42	0.37	0.44	0.40	0.11	0.66	0.13	0.73	0.11	0.37	0.86	0.44	0.33
2	.	0.38	0.42	0.58	0.58	0.48	0.60	0.51	0.51	0.10	0.35	0.17	0.61	0.50
3	.	.	0.58	0.32	0.47	0.45	0.65	0.14	0.77	0.87	0.57	0.59	0.27	0.26
4	.	.	.	0.69	0.77	0.30	0.67	0.34	0.16	0.04	0.45	0.52	0.37	0.40
5	0.66	0.33	0.37	0.37	0.20	0.81	0.64	0.51	0.12	0.24
6	0.89	0.46	0.11	0.27	0.97	0.58	0.32	0.33	0.01
7	0.30	0.73	0.46	0.47	0.61	0.45	0.66	0.48
8	0.66	0.38	0.81	0.59	0.05	0.13	0.76
9	0.22	0.28	0.06	0.76	0.82	0.72
10	0.55	0.89	0.57	0.06	0.09
11	0.31	0.20	0.46	0.67
12	0.75	0.84	0.33
13	0.25	0.57
14	0.38

The total score distributions in Figure 3 did not seem to indicate significantly better fit of the 3PL model than the other two models. At the lower and upper ends, it fit quite well, but again in the middle range of the total scores, it was clearly not an adequate model.

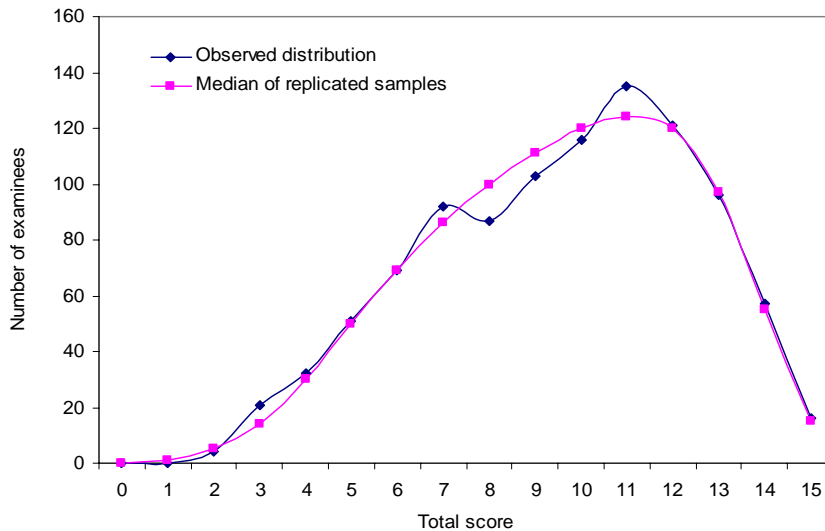


Figure 3. Total Score Distributions of Observed and Replicated Data by Fitting 3PL Model

Discussion and Future Research

In this study the PPMC method was applied in the IRT model-checking context, to assess the fit of three unidimensional IRT models (1PL, 2PL, and 3PL) to the response data generated from the 3PL model. A crucial decision that must be made in the PPMC method is the choice of discrepancy measures. Depending on the context and the aspects of fit to be assessed, the effectiveness of discrepancy measures differs; often a combination of them need be used to adequately assess model fit.

Four discrepancy measures were examined in this study. The proportion correct item score was not found to be effective, nor was it expected to be. But the results based on it provided some assurance about the intensive computations involved. The point biserial correlation coefficient was effective in detecting the misfit of the 1PL model, but not the 2PL model. The same observation can be made about the item pair OR. Sinharay and Johnson (2003) found this measure to be most effective in detecting the misfit of unidimensional IRT models to multidimensional data. The overall findings regarding the total score distributions indicate that this measure was at best moderately effective, which are, again, consistent with those in Sinharay and Johnson (2003).

This study is rather limited in the extent to which the second research purpose was addressed. To evaluate the performance of a discrepancy measure, mainly the size and power, one needs to generate a large number of datasets from a certain IRT model and fit IRT models to these data. In fact, one of the original purposes of this study was to first generate 100 datasets from the 3PL model, then evaluate the power of the discrepancy measures by fitting 1PL and 2PL models to the 100 datasets, and evaluate the size by fitting the 3PL model to the data. Only by fitting the models to a large number of datasets can the size and power be evaluated. However,

based on pilot running by driving WinBUGS (<http://www.mrc-bsu.cam.ac.uk/bugs/>) from R (<http://cran.r-project.org/>), it was estimated to take at least days to complete the simulations and analyses, and that had not included the time it would take to process a huge amount of output. So this part of the plan had to be aborted given the time and resources constraints. A more feasible and efficient alternative is to use the software Ox, which was developed mainly by Jurgen A. Doornik from the University of Oxford (<http://www.doornik.com/products.html#Ox>).

Besides exploring other discrepancy measures and examining the size and power of the measures, another direction for future research is to examine the sensitivity of inferences to changes in the priors and the likelihood. Sinharay, Johnson, and Stern (2006) found that as long as the prior variances are not close to zero, the PPMC results are robust to prior changes. In the current study, only one dataset was generated from the 3PL model with a fixed set of item parameters based on NAEP data. What happens when a different set of item parameters is used? What happens when the data are generated from the 2PL model, from a multidimensional IRT (MIRT) model? What happens when the item response data are polytomous?

As George Box put it, “All models are wrong; some are useful.” In IRT modeling where statistical models with strong assumptions are fit to data, this statement, in particular, has the ring of truth; it asserts the crucial role of evaluating model-data fit. The PPMC method provides an alternative, which may be more effective, to the frequentist approach to assessing model-data fit. Before it can be employed by IRT practitioners, however, more research is needed, especially with regard to the choice of discrepancy measures for specific contexts.

References

- Albert, J., & Ghosh, M. (2000). Item response modeling. In D. K. Dey, S. Ghosh, & B. Mallick (Eds.), *Generalized linear models: A Bayesian perspective* (pp. 173-193). New York: Marcel-Dekker.
- Chen, W., & Thissen, D. (1997). Local dependence indexes for item pairs using item response theory. *Journal of Educational and Behavioral Statistics*, 22(3), 265–289.
- Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2003). *Bayesian data analysis* (2nd ed.). New York: Chapman & Hall/CRC.
- Glas, C. A. W., & Meijer, R. R. (2003). A Bayesian approach to person fit analysis in item response theory models. *Applied Psychological Measurement*, 27(3), 217–233.
- Hambleton, R. K., & Han, N. (2004). *Assessing the fit of IRT models: Some approaches and graphical displays*. Paper presented at the annual meeting of the National Council on Measurement in Education, San Diego, CA.
- Hojtink, H. (2001). Conditional independence and differential item functioning in the twoparameter logistic model. In A. Boomsma, M. A. J. van Duijn, & T. A. B. Snijders (Eds.), *Essays in item response theory*. New York: Springer-Verlag.
- Hojtink, H., & Molenaar, I. W. (1997). A multidimensional item response model: Constrained latent class analysis using the Gibbs sampler and posterior predictive checks. *Psychometrika*, 62(2), 171-189.
- Janssen, R., Tuerlinckx, F., Meulders, M., & De Boeck, P. (2000). A hierarchical IRT model for criterion-referenced measurement. *Journal of Educational and Behavioral Statistics*, 25(3), 285–306.
- Orlando, M., & Thissen, D. (2003). Further investigation of the performance of $S - X^2$: An item

- fit index for use with dichotomous item response theory models. *Applied Psychological Measurement*, 27(4), 289–298.
- Rubin, D. B., & Stern, H. S. (1994). Testing in latent class models using a posterior predictive check distribution. In A. von Eye & C. C. Clogg (Eds.), *Latent variables analysis: Applications for developmental research* (pp. 420-438). Thousand Oaks, CA: Sage Publications.
- Scheines, R., Boomsma, A., & Hoijtink, H. (1999). Bayesian estimation and testing of structural equation models. *Psychometrika*, 64(1), 37-52.
- Sinharay, S., & Johnson, M. S. (2003). *Simulation studies applying posterior predictive model checking for assessing fit of the common item response theory models*. Retrieved December 3, 2006, from <http://www.ets.org/Media/Research/pdf/RR-03-28-Sinharay.pdf>.
- Sinharay, S. (2005). Assessing fit of unidimensional item response theory models using a Bayesian approach. *Journal of Educational Measurement*, 42, 375-394.
- Sinharay, S., Johnson, M. S., & Stern, H. S. (2006). Posterior predictive assessment of item response theory models. *Applied Psychological Measurement*, 30, 298-321.
- van Onna, M. J. H. (2003). Bayesian estimation and model selection in ordered latent class models for polytomous items. *Psychometrika*, 67(4), 519-538.
- Yen, W. (1981). Using simulation results to choose a latent trait model. *Applied Psychological Measurement*, 5, 245–262.

Appendix A

```

model
{
  for (i in 1:N)
  {
    for (j in 1:n)
    {
      t[i,j] <- exp(-a[j]*(theta[i] - b[j]))
      p[i,j] <- c[j]+(1-c[j])/(1 + t[i,j])
      r[i,j] ~ dbern(p[i,j])
      rep[i,j] ~ dbern(p[i,j])
    }

    theta[i] ~ dnorm(0,1)
    total[i] <- sum(rep[i,])
  }

  for (j in 1:n)
  {
    log.a[j] ~ dnorm(0,0.1)
    b[j] ~ dnorm(0,0.1)
    logit.c[j] ~ dnorm(-1.4,1)
    a[j] <- exp(log.a[j])
    c[j] <- exp(logit.c[j])/(1+exp(logit.c[j]))

    item.mean[j] <- mean(rep[,j])
  }

  total.mean <- mean(total[])
  total.sd <- sd(total[])

  for(j in 1:n)
  {
    for (i in 1:N)
    {
      precov[i,j] <- (rep[i,j]-item.mean[j])*(total[i]-total.mean)
    }

    cov[j] <- sum(precov[,j])/(N-1)
    item.sd[j] <- sd(rep[,j])
    rep.pbc[j] <- cov[j]/(item.sd[j]*total.sd)

    pcs.result[j] <- step(item.mean[j]-pcs[j])
    pbc.result[j] <- step(rep.pbc[j]-pbc[j])
  }

  for(j in 1:n)
  {
    for(i in 1:N)
    {
      pattern.1.11[i,j] <- equals( rep[i,1]+rep[i,j], 2)
      pattern.1.01[i,j] <- equals( rep[i,1]-rep[i,j],-1)
      pattern.1.10[i,j] <- equals(-rep[i,1]+rep[i,j],-1)
      pattern.1.00[i,j] <- equals( rep[i,1]+rep[i,j], 0)
    }
  }
}

```

```
pattern.2.11[i,j] <- equals( rep[i,2]+rep[i,j], 2)
pattern.2.01[i,j] <- equals( rep[i,2]-rep[i,j],-1)
pattern.2.10[i,j] <- equals(-rep[i,2]+rep[i,j],-1)
pattern.2.00[i,j] <- equals( rep[i,2]+rep[i,j], 0)

pattern.3.11[i,j] <- equals( rep[i,3]+rep[i,j], 2)
pattern.3.01[i,j] <- equals( rep[i,3]-rep[i,j],-1)
pattern.3.10[i,j] <- equals(-rep[i,3]+rep[i,j],-1)
pattern.3.00[i,j] <- equals( rep[i,3]+rep[i,j], 0)

pattern.4.11[i,j] <- equals( rep[i,4]+rep[i,j], 2)
pattern.4.01[i,j] <- equals( rep[i,4]-rep[i,j],-1)
pattern.4.10[i,j] <- equals(-rep[i,4]+rep[i,j],-1)
pattern.4.00[i,j] <- equals( rep[i,4]+rep[i,j], 0)

pattern.5.11[i,j] <- equals( rep[i,5]+rep[i,j], 2)
pattern.5.01[i,j] <- equals( rep[i,5]-rep[i,j],-1)
pattern.5.10[i,j] <- equals(-rep[i,5]+rep[i,j],-1)
pattern.5.00[i,j] <- equals( rep[i,5]+rep[i,j], 0)

pattern.6.11[i,j] <- equals( rep[i,6]+rep[i,j], 2)
pattern.6.01[i,j] <- equals( rep[i,6]-rep[i,j],-1)
pattern.6.10[i,j] <- equals(-rep[i,6]+rep[i,j],-1)
pattern.6.00[i,j] <- equals( rep[i,6]+rep[i,j], 0)

pattern.7.11[i,j] <- equals( rep[i,7]+rep[i,j], 2)
pattern.7.01[i,j] <- equals( rep[i,7]-rep[i,j],-1)
pattern.7.10[i,j] <- equals(-rep[i,7]+rep[i,j],-1)
pattern.7.00[i,j] <- equals( rep[i,7]+rep[i,j], 0)

pattern.8.11[i,j] <- equals( rep[i,8]+rep[i,j], 2)
pattern.8.01[i,j] <- equals( rep[i,8]-rep[i,j],-1)
pattern.8.10[i,j] <- equals(-rep[i,8]+rep[i,j],-1)
pattern.8.00[i,j] <- equals( rep[i,8]+rep[i,j], 0)

pattern.9.11[i,j] <- equals( rep[i,9]+rep[i,j], 2)
pattern.9.01[i,j] <- equals( rep[i,9]-rep[i,j],-1)
pattern.9.10[i,j] <- equals(-rep[i,9]+rep[i,j],-1)
pattern.9.00[i,j] <- equals( rep[i,9]+rep[i,j], 0)

pattern.10.11[i,j] <- equals( rep[i,10]+rep[i,j], 2)
pattern.10.01[i,j] <- equals( rep[i,10]-rep[i,j],-1)
pattern.10.10[i,j] <- equals(-rep[i,10]+rep[i,j],-1)
pattern.10.00[i,j] <- equals( rep[i,10]+rep[i,j], 0)

pattern.11.11[i,j] <- equals( rep[i,11]+rep[i,j], 2)
pattern.11.01[i,j] <- equals( rep[i,11]-rep[i,j],-1)
pattern.11.10[i,j] <- equals(-rep[i,11]+rep[i,j],-1)
pattern.11.00[i,j] <- equals( rep[i,11]+rep[i,j], 0)

pattern.12.11[i,j] <- equals( rep[i,12]+rep[i,j], 2)
pattern.12.01[i,j] <- equals( rep[i,12]-rep[i,j],-1)
pattern.12.10[i,j] <- equals(-rep[i,12]+rep[i,j],-1)
pattern.12.00[i,j] <- equals( rep[i,12]+rep[i,j], 0)

pattern.13.11[i,j] <- equals( rep[i,13]+rep[i,j], 2)
```

```

    pattern.13.01[i,j] <- equals( rep[i,13]-rep[i,j],-1)
    pattern.13.10[i,j] <- equals(-rep[i,13]+rep[i,j],-1)
    pattern.13.00[i,j] <- equals( rep[i,13]+rep[i,j], 0)

    pattern.14.11[i,j] <- equals( rep[i,14]+rep[i,j], 2)
    pattern.14.01[i,j] <- equals( rep[i,14]-rep[i,j],-1)
    pattern.14.10[i,j] <- equals(-rep[i,14]+rep[i,j],-1)
    pattern.14.00[i,j] <- equals( rep[i,14]+rep[i,j], 0)

    pattern.15.11[i,j] <- equals( rep[i,15]+rep[i,j], 2)
    pattern.15.01[i,j] <- equals( rep[i,15]-rep[i,j],-1)
    pattern.15.10[i,j] <- equals(-rep[i,15]+rep[i,j],-1)
    pattern.15.00[i,j] <- equals( rep[i,15]+rep[i,j], 0)
  }
}

for (j in 2:n)
{
  n.1.11[j-1] <- sum(pattern.1.11[,j])
  n.1.01[j-1] <- sum(pattern.1.01[,j])
  n.1.10[j-1] <- sum(pattern.1.10[,j])
  n.1.00[j-1] <- sum(pattern.1.00[,j])
  odds1[j-1] <- (n.1.11[j-1] * n.1.00[j-1]) / (n.1.01[j-1] * n.1.10[j-1])
  or1[j-1] <- step(odds1[j-1]-orig.odds1[j-1])
}

for (j in 3:n)
{
  n.2.11[j-2] <- sum(pattern.2.11[,j])
  n.2.01[j-2] <- sum(pattern.2.01[,j])
  n.2.10[j-2] <- sum(pattern.2.10[,j])
  n.2.00[j-2] <- sum(pattern.2.00[,j])
  odds2[j-2] <- (n.2.11[j-2] * n.2.00[j-2]) / (n.2.01[j-2] * n.2.10[j-2])
  or2[j-2] <- step(odds2[j-2]-orig.odds2[j-2])
}

for (j in 4:n)
{
  n.3.11[j-3] <- sum(pattern.3.11[,j])
  n.3.01[j-3] <- sum(pattern.3.01[,j])
  n.3.10[j-3] <- sum(pattern.3.10[,j])
  n.3.00[j-3] <- sum(pattern.3.00[,j])
  odds3[j-3] <- (n.3.11[j-3] * n.3.00[j-3]) / (n.3.01[j-3] * n.3.10[j-3])
  or3[j-3] <- step(odds3[j-3]-orig.odds3[j-3])
}

for (j in 5:n)
{
  n.4.11[j-4] <- sum(pattern.4.11[,j])
  n.4.01[j-4] <- sum(pattern.4.01[,j])
  n.4.10[j-4] <- sum(pattern.4.10[,j])
  n.4.00[j-4] <- sum(pattern.4.00[,j])
  odds4[j-4] <- (n.4.11[j-4] * n.4.00[j-4]) / (n.4.01[j-4] * n.4.10[j-4])
  or4[j-4] <- step(odds4[j-4]-orig.odds4[j-4])
}

for (j in 6:n)

```

```

{
  n.5.11[j-5] <- sum(pattern.5.11[,j])
  n.5.01[j-5] <- sum(pattern.5.01[,j])
  n.5.10[j-5] <- sum(pattern.5.10[,j])
  n.5.00[j-5] <- sum(pattern.5.00[,j])
  odds5[j-5] <- (n.5.11[j-5] * n.5.00[j-5]) / (n.5.01[j-5] * n.5.10[j-5])
  or5[j-5] <- step(odds5[j-5]-orig.odds5[j-5])
}

for (j in 7:n)
{
  n.6.11[j-6] <- sum(pattern.6.11[,j])
  n.6.01[j-6] <- sum(pattern.6.01[,j])
  n.6.10[j-6] <- sum(pattern.6.10[,j])
  n.6.00[j-6] <- sum(pattern.6.00[,j])
  odds6[j-6] <- (n.6.11[j-6] * n.6.00[j-6]) / (n.6.01[j-6] * n.6.10[j-6])
  or6[j-6] <- step(odds6[j-6]-orig.odds6[j-6])
}

for (j in 8:n)
{
  n.7.11[j-7] <- sum(pattern.7.11[,j])
  n.7.01[j-7] <- sum(pattern.7.01[,j])
  n.7.10[j-7] <- sum(pattern.7.10[,j])
  n.7.00[j-7] <- sum(pattern.7.00[,j])
  odds7[j-7] <- (n.7.11[j-7] * n.7.00[j-7]) / (n.7.01[j-7] * n.7.10[j-7])
  or7[j-7] <- step(odds7[j-7]-orig.odds7[j-7])
}

for (j in 9:n)
{
  n.8.11[j-8] <- sum(pattern.8.11[,j])
  n.8.01[j-8] <- sum(pattern.8.01[,j])
  n.8.10[j-8] <- sum(pattern.8.10[,j])
  n.8.00[j-8] <- sum(pattern.8.00[,j])
  odds8[j-8] <- (n.8.11[j-8] * n.8.00[j-8]) / (n.8.01[j-8] * n.8.10[j-8])
  or8[j-8] <- step(odds8[j-8]-orig.odds8[j-8])
}

for (j in 10:n)
{
  n.9.11[j-9] <- sum(pattern.9.11[,j])
  n.9.01[j-9] <- sum(pattern.9.01[,j])
  n.9.10[j-9] <- sum(pattern.9.10[,j])
  n.9.00[j-9] <- sum(pattern.9.00[,j])
  odds9[j-9] <- (n.9.11[j-9] * n.9.00[j-9]) / (n.9.01[j-9] * n.9.10[j-9])
  or9[j-9] <- step(odds9[j-9]-orig.odds9[j-9])
}

for (j in 11:n)
{
  n.10.11[j-10] <- sum(pattern.10.11[,j])
  n.10.01[j-10] <- sum(pattern.10.01[,j])
  n.10.10[j-10] <- sum(pattern.10.10[,j])
  n.10.00[j-10] <- sum(pattern.10.00[,j])
  odds10[j-10] <- (n.10.11[j-10]*n.10.00[j-10]) / (n.10.01[j-
10]*n.10.10[j-10])
}

```

```

    or10[j-10] <- step(odds10[j-10]-orig.odds10[j-10])
  }

  for (j in 12:n)
  {
    n.11.11[j-11] <- sum(pattern.11.11[,j])
    n.11.01[j-11] <- sum(pattern.11.01[,j])
    n.11.10[j-11] <- sum(pattern.11.10[,j])
    n.11.00[j-11] <- sum(pattern.11.00[,j])
    odds11[j-11] <- (n.11.11[j-11]*n.11.00[j-11]) / (n.11.01[j-
11]*n.11.10[j-11])
    or11[j-11] <- step(odds11[j-11]-orig.odds11[j-11])
  }

  for (j in 13:n)
  {
    n.12.11[j-12] <- sum(pattern.12.11[,j])
    n.12.01[j-12] <- sum(pattern.12.01[,j])
    n.12.10[j-12] <- sum(pattern.12.10[,j])
    n.12.00[j-12] <- sum(pattern.12.00[,j])
    odds12[j-12] <- (n.12.11[j-12]*n.12.00[j-12]) / (n.12.01[j-
12]*n.12.10[j-12])
    or12[j-12] <- step(odds12[j-12]-orig.odds12[j-12])
  }

  for (j in 14:n)
  {
    n.13.11[j-13] <- sum(pattern.13.11[,j])
    n.13.01[j-13] <- sum(pattern.13.01[,j])
    n.13.10[j-13] <- sum(pattern.13.10[,j])
    n.13.00[j-13] <- sum(pattern.13.00[,j])
    odds13[j-13] <- (n.13.11[j-13]*n.13.00[j-13]) / (n.13.01[j-
13]*n.13.10[j-13])
    or13[j-13] <- step(odds13[j-13]-orig.odds13[j-13])
  }

  for (j in 15:n)
  {
    n.14.11[j-14] <- sum(pattern.14.11[,j])
    n.14.01[j-14] <- sum(pattern.14.01[,j])
    n.14.10[j-14] <- sum(pattern.14.10[,j])
    n.14.00[j-14] <- sum(pattern.14.00[,j])
    odds14[j-14] <- (n.14.11[j-14]*n.14.00[j-14]) / (n.14.01[j-
14]*n.14.10[j-14])
    or14[j-14] <- step(odds14[j-14]-orig.odds14[j-14])
  }

  for(i in 1:N)
  {
    score0[i] <- equals(total[i],0)
    score1[i] <- equals(total[i],1)
    score2[i] <- equals(total[i],2)
    score3[i] <- equals(total[i],3)
    score4[i] <- equals(total[i],4)
    score5[i] <- equals(total[i],5)
    score6[i] <- equals(total[i],6)
    score7[i] <- equals(total[i],7)
  }

```

```
    score8[i] <- equals(total[i],8)
    score9[i] <- equals(total[i],9)
    score10[i] <- equals(total[i],10)
    score11[i] <- equals(total[i],11)
    score12[i] <- equals(total[i],12)
    score13[i] <- equals(total[i],13)
    score14[i] <- equals(total[i],14)
    score15[i] <- equals(total[i],15)
  }

  ts.result[1] <- sum(score0[])
  ts.result[2] <- sum(score1[])
  ts.result[3] <- sum(score2[])
  ts.result[4] <- sum(score3[])
  ts.result[5] <- sum(score4[])
  ts.result[6] <- sum(score5[])
  ts.result[7] <- sum(score6[])
  ts.result[8] <- sum(score7[])
  ts.result[9] <- sum(score8[])
  ts.result[10] <- sum(score9[])
  ts.result[11] <- sum(score10[])
  ts.result[12] <- sum(score11[])
  ts.result[13] <- sum(score12[])
  ts.result[14] <- sum(score13[])
  ts.result[15] <- sum(score14[])
  ts.result[16] <- sum(score15[])
}
```