Worked example with person level covariates for the R-package CIRT version 2.0

Rinke H. Klein Entink *

University of Twente

July 31, 2009

1 Introduction

This is a brief worked out example to illustrate the use and possibilities of the CIRT-package for the joint analysis of responses and response times on test items. For reading on the ideas behind the modeling framework, please check the references below. CIRT version 2.0 has been compiled and tested under R version 2.9.1. You can install the package in R via the "Packages" menu, selecting "Install from local zip-file" and then selecting the cirt package. Subsequently, load the package via the menu and you're all set to follow this example!

2 Example

2.1 Reading the data

In the zip-file the datasets Ycirt.txt, Tcirt.txt and XP.txt were provided, that contain the responses, response times and person level covariates, respectively. First, these have to be read in R:

$Y <- read.table("D:\\MyDocuments\\CIRT\\Ycirt.txt")$

$Time <- read.table("D:\\MyDocuments\\CIRT\\Tcirt.txt")$

$XP <- read.table("D:\\MyDocuments\\CIRT\\XP.txt")$

*E-mail: r.h.kleinentink@gw.utwente.nl. Website: www.kleinentink.eu.

Subsequently, make sure these are read as matrices:

$Y <- as.matrix(Y)$

$Time <- as.matrix(Time)$

$XP <- as.matrix(XP)$

Now the data are loaded. The number of persons who took the test is $N = 523$ and they answered $K = 23$ items: $N <- length(Y[,1])$

$K <- length(Y[1,])$

2.2 Estimating the first models

As a first step, we evaluate if the 1-parameter IRT model does the job or that the 2-parameter model gives a more appropriate description of the data. To do so, we fit both models and use the DIC criterium to select the best model. We run the algorithm for 6,000 iterations and use a burn-in of 1,000 iterations. So, the estimates are based on the last 5,000 iterations of the chain. (NOTE: I ignore convergence issues for the moment and restrict myself to the functionality of CIRT. I recommend to use the coda package to obtain an idea of the convergence of the MCMC algorithm.)

Specify:

$iter <- 6000$

$out1 <- estimate(Y, Time, N, K, iter, PL = 1)$, where $PL = 1$ restricts the discrimination parameters of the response model to be equal for all items.

$out2 <- estimate(Y, Time, N, K, iter)$, not specifying the value of PL will give you the 2-parameter response model, which is the default. (A similar option is available for the response time model (TM =1 and TM=2, respectively), see the help function of estimate by typing '?estimate' in the command prompt.)

2.3 Output list

The out1 and out2 objects that have been obtained are lists that contain the mcmc output of the algorithm. You can assess all the estimates provided by the programm directly from these lists. For example, we can make a plot of the estimated speed parameters against the estimated ability parameters. When you type '?estimate' you can see that these two parameters are stored in elements 26 and 27 in the list. We can obtain them via:

$theta <- matrix(out[[26]], nrow = N, ncol = 2)$, now $theta$ is a matrix with in the first column the EAP and in the second column the posterior standard deviations of the ability parameters. Similarly, we obtain the speed parameters by:

$zeta <- matrix(out[[27]], nrow = N, ncol = 2)$.

And to plot ability against speed we plot the EAP estimates of the parameters:

$plot(theta[, 1], zeta[, 1], xlab = \text{"Estimated Ability"}, ylab = \text{"Estimated Speed"})$.

2.4 Using the summarize function

But we were interested if the 1-parameter model would suffice or not. Therefore, we will analyze the output of the programm for which the function "summarize" is available. It has the output file and a burnin for the mcmc chain as required arguments. Just type "summarize(out1,1000)" in the command prompt. You get a report that provides you with the estimates for the item parameters of the response and response time model. Also, you can see that there is a small negative covariance between ability and speed (-.20). Lastly, an estimated DIC value is reported for the fitted model, which for the model with restriction $PL = 1$ gives us $DIC = 44526.81$. When you obtain the same report for the 2-parameter model, you get that the $DIC = 44457.18$. The latter value is somewhat lower and suggests that we should select the 2-parameter model for the IRT measurement model.

2.5 Model fit

There are some model fit analysis possible using the functions 'fitirt' and 'fitrt'. For the IRT model, there is a basic check on unidimensionality of the model and for conditional independence of the items. See the references to Sinharay (2005) and Sinharay, S., Johnson, M. S., and Stern, H. S. (2006) below for details and explanations. If you run the first function:

$fitirt(N, K, out2)$

You get two tables. The first is the Odds ratio statistic, the second the observed sum score statistic. Extreme p-values denote unlikely observations under the model, for instance, p-values lower than .025 or higher than .975.

The fitrt function plots the Bayesian residuals of the response time model for item k. If the RT-model fits well, these residuals will follow their expected values and you should get a roughly linear plot. For example,

for item 12 you can see that this is the case:
$fitrt(Y, N, K, out2, 12)$ gives you the plot.

2.6 Regressing the person level covariates on ability and speed

From the model fit statistics you can see that the model fits well. Now we can proceed with the next step and try to explain some of the variance in the ability and speed levels of the candidates using the covariate matrix XP. The covariate matrix XP contains three columns, of which the first is a column of 1s for the intercept. If we only provide XP, it is assumed that the covariates for ability and speed are the same. To fit the model, we use the function:
$out3 <- estimateXP(Y, Time, N, K, iter, XP)$
iter was still 6,000, and we will specify a burnin of the first 1,000 iterations and use the remaining 5,000 iterations to obtain our estimates of the regression effects. The following function does exactly that:
$summarizeXP(out3, 1000)$.
Now you have a report that gives the EAPs and posterior standard deviations of the regression effects for ability and speed. Note that the mean is fixed to zero for both ability and speed, which results from the identification restrictions of the model.

Two tables are reported for the estimated regression effects. The first is for ability, the second for speed. Also reported are estimates of a Bayesian measure of explained variance in the ability/speed by the covariates (see Gelman and Pardoe, 2006). We see that the proportion explained variance in ability is approximately 40 % and in speed approximately 27 %. The effect of the first covariate on ability is positive, the $EAP = .68$ and $SD = 0.05$. The second one, $X2$, has a negative effect. Looking at the regression of X1 and X2 on Speed, however, we see that the estimate for X1 is almost zero ($EAP = -0.06, SD = 0.05$). This suggests that this effect might not explain much variance in speed.

A simple test is to evaluate if zero is contained in the .95 Highest Posterior Density region of the estimated effect. To obtain the .95 HPD region, we will use the *coda* package that is available for *R*. For the following, you should have this package installed:
$library(coda)$
$gamma21 <- matrix(out3[[45]], nrow = iter)[, 5]$ *Obtain the iter mcmc draws for this covariate.*
$gamma21 <- as.mcmc(gamma21[1001 : iter])$ *Make an mcmc object of it, using the last 5,000 draws.*
$HPDinterval(gamma21, .95)$

The HPDinterval-function returns now the interval $[-.15, 0.03]$, which clearly contains the value 0. Therefore, we should reject the hypothesis that X1 explains any variance in the speed level of the persons.

We can kick out X1 of the regression for speed, by specifying a new covariate matrix for the speed parameters that only contains the intercept term and X2:

$XP2 <- XP1[, c(1, 3)]$

Now, using the optional statement $XP2 = XP2$ we can allow for different covariate matrices for ability and speed:

$out4 <- estimateXP(Y, Time, N, K, iter, XP, XP2 = XP2)$

Now we have our final estimates of the effects:

$summarizeXP(out4, 1000)$. Note that the covariate for speed is labeled $X1$ in the output again. The covariates are just numbered, that's why.

3 Some last remarks

Of course, the data provided by this example is a simulated one. In practice, my experience is that you need more iterations than 5,000 to obtain good estimates. You should be aware that using MCMC algorithms requires some expertise in analyzing the convergence of the chain, before you can rely on the estimates as they are provided by the cirt programm. I recommend making some traceplots first and using the coda (or boa) package to analyze the mcmc output for convergence, before you start making inferences from the model.

Furthermore, not all possibilities were discussed in this example. Use the R-help files of the package to see what options there are and how to obtain parameter estimates from the output list.

References

[1] A. Gelman and I. Pardoe (2006). Bayesian Measures of Explained Variance and Pooling in Multilevel (Hierarchical) Models, Technometrics, 2006, 48, 241-251.

[2] R. H. Klein Entink (2009). Statistical Models for Responses and Response Times. University of Twente, Unpublished Doctoral thesis.

[3] R. H. Klein Entink, J.-T. Kuhn, L. F. Hornke and J.-P. Fox (2009). Evaluating cognitive theory: A joint modeling approach using responses and response times, Psychological Methods, 14, 54-75.

[4] R.H. Klein Entink, J.-P. Fox, W.J. van der Linden (2009). A multivariate multilevel approach to the modeling of accuracy and speed of test takers, Psychometrika, 74, 21-48.

[5] J.-P. Fox, R.H. Klein Entink, W.J. van der Linden, (2007), Modeling of response times and patterns: With a package CIRT, Journal of Statistical Software, 20, issue 7, http://www.jstatsoft.org .

[6] W. J. van der Linden (2007). A hierarchical framework for modeling speed and accuracy on test items, Psychometrika, 72, 287-308.

[7] Sinharay, S. (2005). Assessing fit of unidimensional iterm response theory models using a Bayesian approach. Journal of Educational Measurement, 42, 375-394.

[8] Sinharay, S., Johnson, M. S., and Stern, H. S. (2006). Posterior predictive assessment of item repsonse theory models. Applied Psychological Measurement, 30, 298-321.