

Appendix

R code to generate artificial data according to the Rasch model

```
Rasch.data<-function(n,k,beta){ #1#
set.seed(1) #2#
theta<-rnorm(n,0,1) #3#
probs<-matrix(0,nrow=n,ncol=k) #4#
dat<-matrix(0,nrow=n,ncol=k) #5#
for(i in 1:n){ #6#
  for(j in 1:k){ #7#
    probs[i,j]<-exp(theta[i]-beta[j])/(1+exp(theta[i]-beta[j])) #8#
    dat[i,j]<-rbinom(1,1,probs[i,j]) #9#
  } #10#
} #11#
return(dat) #12#
} #13#
```

In #1# we assign the name Rasch.data specifying that it is a function of the number of persons n , the number of items k , and the real values of the β parameters. In #3# it is indicated that the values of the abilities are drawn from a normal distribution of zero mean and standard deviation 1. In #4# and #5# matrices are created to store the probabilities p_{ij} and the data matrix of zeros and ones, whose elements are the observations y_{ij} .

Once the function has been compiled in R, a call to generate data is as follows

```
data.set1=Rasch.data(500,11,seq(-2.5,2.5,0.5))
where data.set1 is the object in which the data are stored.
```

R-WinBUGS code to fit the models

We present the used code to fit the models presented in the paper. This code can easily be adapted to fit other models. For a detailed explanation of the R2WinBUGS package the reader is referred to Sturtz, Ligges, and Gelman (2005).

Preliminaries

1. Install WinBUGS if it is not already installed in your system. The program and all the information for installation can be found at <http://www.mrc-bsu.cam.ac.uk/bugs/>

2. Install R and the R2WinBUGS package if they are not already installed. R and all available packages can be found at <http://www.r-project.org/>

3. Write the WinBUGS code of the model in the file file.txt (see below)

4. Run the R script to call WinBUGS from R (see below)

WinBUGS code

Copy and paste the following WinBUGS code to create the Rasch Model rd.txt file.

```
model; #1#
{ #2#
for(i in 1:n){ #3#
theta[i]~dnorm(0,tau) #4#
for(j in 1:k){ #5#
p[i,j]<-exp(theta[i]-beta[j])/(1+exp(theta[i]-beta[j])) #6#
x[i,j]~dbern(p[i,j]) #7#
} #8#
} #9#
for(j in 1:k){ #10#
beta[j]~dnorm(0,0.001) #11#
} #12#
tau<-pow(sigma,-2) #13#
```

```
sigma~dunif(0,100) #14#  
sigma.theta<-1/tau #15#  
} #16#
```

From lines 3 to 9 the likelihood of the model is specified. From lines 10 to 12, the prior distribution for the β parameters is specified. From lines #13# to #15# the prior for the σ parameter is specified

```
Fitting the Rasch model using ltm to obtain MML estimates to be compared with WinBUGS results  
library(ltm)  
rasch.fit=rasch(data.set1, constraint = cbind(ncol(data.set1) + 1, 1))  
summary(rasch.fit)[1]
```

R code to call WinBUGS from R via the R2WinBUGS package

```
library(R2WinBUGS) #1#  
dat=read.table("data.set1.txt",header=F) #2#  
n=nrow(dat) #3#  
k=ncol(dat) #4#  
x=as.matrix(dat) #5#  
data=list("n","k","x") #6#  
inits=function(){ #7#  
list(theta=rnorm(n,0,1),beta=rnorm(11,0,1),sigma=runif(1)) #8#  
} #9#  
parameters=c("beta","sigma.theta") #10#  
mod1<-bugs (data, inits, parameters, "Rasch Model rd.txt",n.chains=3, n.iter=1000, #11#  
n.thin=1) #12#
```

In #1# the R2WinBUGS library which allows to call WinBUGS from R is loaded. In #2# the data set in file `data.set1.txt` is loaded. In #3# and #4# the number of rows and columns are specified. Note that these values coincide with the number of individuals and the number of items, respectively. From #7# to #9#, initial values for the algorithm are generated. In #10# it is specified which are the parameter we want to obtain samples from its posterior distributions. In #10# and #11# the bugs function makes call to WinBUGS, using the arguments previously described. The `n.chains` argument indicate that 3 parallel chains are being run, each with `n.iter=1000` iterations. By default, half of each chain is discarded as a burn-in