Fitting the m-AFC eyewitness data to UV-SDT model

Kym McCormick

16 April 2019

Forced choice accuracy is given by

.

Step one is to find the best fitting UV-SDT model parameters to the empirical data. I adaped the following code from Kellen Supplemental script (2015)

#Empirical data for lineup size k = {2,3,...,7,8}  
  
BMIDataCID <- c(0.8862, 0.7616, 0.6737, 0.6491, 0.5606, 0.5498, 0.50276)  
  
data\_kafc <- c(rbind(BMIDataCID,1-BMIDataCID))\*300  
  
SDT\_kafc <- function(Q, data, param.names, n.params, tmp.env, lower.bound, upper.bound){  
   
 e<-vector(  
 mode = "numeric",  
 length = 4  
 )  
  
 mu <- Q[1]  
 ss <- Q[2]  
   
 rank <- function(i,k,mu=1,ss=1){  
 f1 <- function(x,i,k,mu,ss) {  
 choose(k -1, i-1)\*dnorm(x,mu,ss)\*pnorm(x)\*\*(k-i)\*(1-pnorm(x))\*\*(i-1)   
 }   
 tmp <- vector(  
 mode = "numeric",  
 length = length(i)  
 )  
 for(ii in 1:length(i))   
 tmp[ii] <- integrate(  
 f = f1,  
 lower = -Inf,   
 upper = Inf,   
 i = i[ii],  
 k = k,  
 mu = mu,  
 ss = ss  
 ) $value  
 return(tmp)  
 }  
   
  
 e[1:2] <- rank(1:2,2,mu=mu,ss=ss)   
 e[3:4] <- c(rank(1,3,mu=mu,ss=ss), 1- rank(1,3,mu=mu,ss=ss))  
 e[5:6] <- c(rank(1,4,mu=mu,ss=ss), 1- rank(1,4,mu=mu,ss=ss) )  
 e[7:8] <- c(rank(1,5,mu=mu,ss=ss), 1- rank(1,5,mu=mu,ss=ss))  
 e[9:10] <- c(rank(1,6,mu=mu,ss=ss), 1- rank(1,6,mu=mu,ss=ss) )  
 e[11:12] <- c(rank(1,7,mu=mu,ss=ss), 1- rank(1,7,mu=mu,ss=ss) )  
 e[13:14] <- c(rank(1,8,mu=mu,ss=ss), 1- rank(1,8,mu=mu,ss=ss) )   
 #add this last line if you have k=8 kAFC data  
  
   
 LL <- -sum(data[data!=0]\*log(e[data!=0]))  
 return(LL)  
}  
  
  
fit\_kafc <- fit.mptinr(  
 data = data\_kafc,   
 objective = SDT\_kafc,   
 param.names = c("mu", "sigma"),   
 categories.per.type = c(2,2,2,2,2,2,2),   
 lower.bound = c(0,0.1),   
 upper.bound = Inf,   
 n.optim = 5,  
 show.messages = FALSE  
 )  
  
fit\_kafc

## $goodness.of.fit  
## Log.Likelihood G.Squared df p.value  
## 1 -1276.376 2.608118 5 0.7601314  
##   
## $information.criteria  
## AIC BIC  
## 1 6.608118 17.9075  
##   
## $model.info  
## rank.fisher n.parameters n.independent.categories  
## 1 2 2 7  
##   
## $parameters  
## estimates lower.conf upper.conf  
## mu 1.3318149 1.2426560 1.4209739  
## sigma 0.6336813 0.3190277 0.9483348  
##   
## $data  
## $data$observed  
## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]  
## [1,] 265.86 34.14 228.48 71.52 202.11 97.89 194.73 105.27 168.18 131.82  
## [,11] [,12] [,13] [,14]  
## [1,] 164.94 135.06 150.828 149.172  
##   
## $data$predicted  
## list()  
##   
##   
## $fitting.runs  
## Min. 1st Qu. Median Mean 3rd Qu. Max.  
## [1,] -1276.376 -1276.376 -1276.376 -1276.376 -1276.376 -1276.376

Below is the code to simulate CIDs from the best fitting UV-SDT model parameters

mu=1.3318149  
ss=0.6336813  
simulated <- vector("numeric")  
rank <- function(i,k,mu=1.3318149,ss=0.6336813){  
 f1 <- function(x,i,k,mu,ss) {  
 choose(k -1, i-1)\*dnorm(x,mu,ss)\*pnorm(x)\*\*(k-i)\*(1-pnorm(x))\*\*(i-1)   
 }   
 tmp <- vector(  
 mode = "numeric",  
 length = length(i)  
 )  
 for(ii in 1:length(i))   
 tmp[ii] <- integrate(  
 f = f1,  
 lower = -Inf,   
 upper = Inf,   
 i = i[ii],  
 k = k,  
 mu = mu,  
 ss = ss  
 ) $value  
 return(tmp)  
}  
  
simulated[1] <- rank(1,2,mu=mu,ss=ss)   
simulated[2] <- rank(1,3,mu=mu,ss=ss)  
simulated[3] <- rank(1,4,mu=mu,ss=ss)  
simulated[4] <- rank(1,5,mu=mu,ss=ss)  
simulated[5] <- rank(1,6,mu=mu,ss=ss)  
simulated[6] <- rank(1,7,mu=mu,ss=ss)  
simulated[7] <- rank(1,7,mu=mu,ss=ss)  
simulated

## [1] 0.8696982 0.7716377 0.6945789 0.6321235 0.5803091 0.5365280 0.5365280

I realised then that I need to add in a G squared test of some sort (not sure if it is correct). I don’t really understand how this differs from the loglikelihood test used in the fitting….

sim\_kafc <- c(rbind(simulated,1-simulated))\*300  
matriz <- cbind(sim\_kafc,data\_kafc)  
matriz

## sim\_kafc data\_kafc  
## [1,] 260.90946 265.860  
## [2,] 39.09054 34.140  
## [3,] 231.49132 228.480  
## [4,] 68.50868 71.520  
## [5,] 208.37366 202.110  
## [6,] 91.62634 97.890  
## [7,] 189.63704 194.730  
## [8,] 110.36296 105.270  
## [9,] 174.09274 168.180  
## [10,] 125.90726 131.820  
## [11,] 160.95840 164.940  
## [12,] 139.04160 135.060  
## [13,] 160.95840 150.828  
## [14,] 139.04160 149.172

GTest(matriz)

##   
## Log likelihood ratio (G-test) test of independence without  
## correction  
##   
## data: matriz  
## G = 1.986, X-squared df = 13, p-value = 0.9998