Online Appendix

Online appendix to accompany Pek, J. & Wu, H. (2015). Profile Likelihood-Based Confidence Intervals and Regions for Structural Equation Models. *Psychometrika*. doi = 10.1007/s11336-015-9461-1

OpenMx Code for Empirical Example on Latent Variable Mediation

This example involves a latent variable mediation model published by Schmitt, Branscombe, Kobrynowicz and Owen (2002). OpenMx code for computing the profile likelihood-based CI of the indirect effect of perceived discrimination (PD) on in-group identification (IGI) is provided below. Additionally, the profile likelihood-based CR for the total effect of PD on psychological well-being (PWB), consisting of the direct and indirect effect of PD on PWB is also provided.

```
source('http://openmx.psyc.virginia.edu/getOpenMx.R')
require(OpenMx)

#data

vechS <- rbind(.47,.2 ,.47,.1 , .01,.08, .16,-.01,-.04,-.06, .06, .14,
                   .27,.36,.17, .06,.23, .27,-.03,  0 ,-.06, .04, .05,
                       .29,.07, .03,.06, .04,-.14,-.07,-.11, .25, .12,
                           .05,-.08,.09,-.01,-.14,-.10,-.18, .13, .15,
                                .74,.72, .67, .14, .11, .09, .03,-.08,
                                    .63, .59, .2 , .13, .15, .08,-.07,
                                         .65, .13, .12, .08, .02,-.10,
                                              .19, .12, .17, .00,-.12,
                                                   .70, .57,-.45,-.46,
                                                        .50,-.52,-.65,
                                                            -.36,-.42,
                                                                  .53)
S <- diag(1,13)/2
S[lower.tri(S)] <- vechS
S <- S+t(S)
Sd <- diag(c(1.39,1.28,20.0,1.75,1.13,1.07,1.06,1.26,1.33,.91,.93,6.90,4.43))
S <- Sd%*%S%*%Sd
S <- (S+t(S))/2
varnames=paste("item",1:13,sep="")
dimnames(S) <- list(varnames,varnames)
n <- 220
```

```
#OpenMx matrices & fit model to data

lambda <- rbind(cbind(c(.72,.64,.38,.63),rep(0,4),rep(0,4)),
          cbind(rep(0,4),c(.89,.80,.81,.76),rep(0,4)),
          cbind(rep(0,5),rep(0,5),c(.78,.88,.62,-.61,-.71)))
lab <- L <- (lambda!=0)
lab[lab==1] <- paste("l",1:13,sep="")
Lambda <- mxMatrix(type="Full",nrow=13,ncol=3,free=L,values=lambda,
          label=lab,ubound=c(rep(.99,11),-0.01,-0.01),
          lbound=c(rep(0.01,11),-.99,-.99),name="Lambda")

#Effect of PD on IGI
PDIGI <- mxMatrix(type="Full",nrow=1,ncol=1,free=T,value=.17,
 label="pdigi",ubound=.99,lbound=-.99,name="PDIGI")

#Effect of IGI on PWB
IGIPWB <- mxMatrix(type="Full",nrow=1,ncol=1,free=T,value=.2,
          label="igipwb",ubound=.99,lbound=-.99,name="IGIPWB")

#Effect of PD on PWB
PDPWB <- mxMatrix(type="Full",nrow=1,ncol=1,free=T,value=-.18,
         label="pdpwb",ubound=.99,lbound=-.99,name="PDPWB")

#Indirect Effect
Med <- mxAlgebra(PDIGI*IGIPWB,name="Med")

#Total Effect
cov13 <- mxAlgebra(Med+PDPWB,name="cov13")
res3 <- mxAlgebra(1-PDPWB*cov13-IGIPWB*IGIPWB,name="res3")
cov23 <- mxAlgebra(IGIPWB+PDIGI*PDPWB,name="cov23")
Phi<-mxAlgebra(cbind(rbind(1,PDIGI,cov13),rbind(PDIGI,1,cov23),
     rbind(cov13,cov23,1)),name="Phi")
D <- mxMatrix(type="Full",nrow=13,ncol=1,free=T,value=Sd,
     label=paste("d",1:13,sep=""),lbound=.5,name="D")
Lambda0 <- mxAlgebra(vec2diag(D)%*%Lambda,name="Lambda0")
Sigma0 <- mxAlgebra(Lambda0%*%Phi%*%t(Lambda0),name="Sigma0")
Sigma <- mxAlgebra(Sigma0-vec2diag(diag2vec(Sigma0))+vec2diag(D*D),
```

```
         name="Sigma")

Model <- mxModel("Model",Lambda,PDIGI,IGIPWB,PDPWB,Med,Phi,D,Lambda0,
         Sigma0,Sigma,cov13,res3,cov23,mxMLObjective(covariance="Sigma",
         dimnames=varnames),mxData(observed=S,type="cov",numObs=n))

fit <- mxRun(Model)
est <- fit@output$estimate
varname <- names(est)
Model <- omxSetParameters(Model,labels=c("pdigi","igipwb","pdpwb"),
         lbound=-.5,ubound=.5)

### constraint in Equation 12

crit1 <- fit@output$minimum+qchisq(.95,df=1)
crit3 <- fit@output$minimum+qchisq(.95,df=3)

pvalue1 <- mxConstraint(Model.objective==crit1,name="pvalue1")
pvalue3 <- mxConstraint(Model.objective==crit3,name="pvalue3")

### Point-wise CI for effect of PD on IGI

PDIGIlb1 <- mxModel("PDIGIlb1",Model,mxAlgebraObjective("Model.PDIGI"),pvalue1)
fit.pdigilb1 <- mxRun(PDIGIlb1)
pdigi.lb1 <- fit.pdigilb1@output$minimum
path <- fit.pdigilb1@output$estimate[c("pdigi","igipwb","pdpwb")]
load <- fit.pdigilb1@output$estimate[paste("l",1:13,sep="")]
cat(all(path<.5),all(path>-.5),all(load[1:11]<.99),all(load[1:11]>.01),
    all(load[12:13]< -.01),all(load[12:13]> -.99),
    mxEval(c(Model.res3,Model.objective),fit.pdigilb1)-c(0,crit1),"\n")

negPDIGI <- mxAlgebra(-Model.PDIGI,name="negPDIGI")
PDIGIub1 <- mxModel("PDIGIub1",negPDIGI,Model,mxAlgebraObjective("negPDIGI"),
            pvalue1)
fit.pdigiub1 <- mxRun(PDIGIub1)
pdigi.ub1 <- -fit.pdigiub1@output$minimum
path <- fit.pdigiub1@output$estimate[c("pdigi","igipwb","pdpwb")]
load <- fit.pdigiub1@output$estimate[paste("l",1:13,sep="")]
```

```
cat(all(path<.5),all(path>-.5),all(load[1:11]<.99),all(load[1:11]>.01),
    all(load[12:13]< -.01),all(load[12:13]> -.99),
    mxEval(c(Model.res3,Model.objective),fit.pdigiub1)-c(0,crit1),"\n")

pdigi.lb1
pdigi.ub1
```

Note that the cat line prints out indicators on whether the final estimates are within the boundaries set for parameters. These boundaries are set to help with the optimization, similar to how OpenMx matrices are defined. For a proper estimate, there should be no boundary solution. If a parameter is on the boundary, a more lenient boundary should be used. The second last number is the residual variance of PWB, which should be positive. The last number, which should be extremely small, is the difference between the -2 log-likelihood from the original model solution and the constrained model solution.

```
### CR for total effect

PDPWBlb3 <- mxModel("PDPWBlb3",Model,mxAlgebraObjective("Model.PDPWB"),pvalue3)
fit.pdpwblb3 <- mxRun(PDPWBlb3)
pdpwb.lb3 <- fit.pdpwblb3@output$minimum
pdpwb.lb3.est <- fit.pdpwblb3@output$estimate
path <- fit.pdpwblb3@output$estimate[c("pdigi","igipwb","pdpwb")]
load <- fit.pdpwblb3@output$estimate[paste("l",1:13,sep="")]
cat(all(path<.5),all(path>-.5),all(load[1:11]<.99),all(load[1:11]>.01),
    all(load[12:13]< -.01),all(load[12:13]> -.99),
    mxEval(c(Model.res3,Model.objective),fit.pdpwblb3)-c(0,crit3),"\n")

negPDPWB<-mxAlgebra(-Model.PDPWB,name="negPDPWB")
PDPWBub3 <- mxModel("PDPWBub3",negPDPWB,Model,mxAlgebraObjective("negPDPWB"),pvalue3)
fit.pdpwbub3 <- mxRun(PDPWBub3)
pdpwb.ub3<- -fit.pdpwbub3@output$minimum
pdpwb.ub3.est <- fit.pdpwbub3@output$estimate
path <- fit.pdpwbub3@output$estimate[c("pdigi","igipwb","pdpwb")]
load <- fit.pdpwbub3@output$estimate[paste("l",1:13,sep="")]
cat(all(path<.5),all(path>-.5),all(load[1:11]<.99),all(load[1:11]>.01),
    all(load[12:13]< -.01),all(load[12:13]> -.99),
    mxEval(c(Model.res3,Model.objective),fit.pdpwbub3)-c(0,crit3),"\n")
```

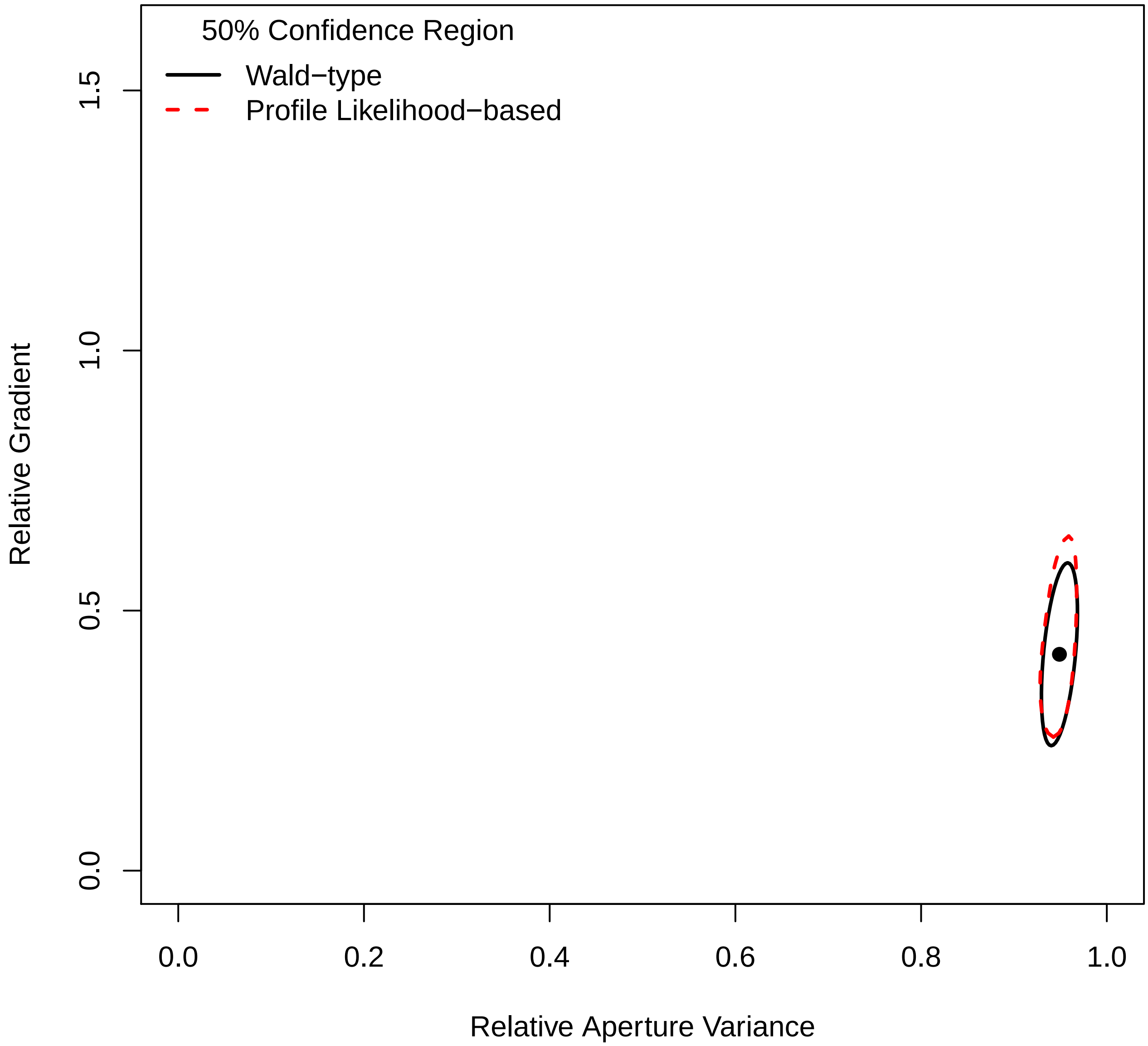
```
Medlb3 <- mxModel("Medlb3",Model,mxAlgebraObjective("Model.Med"),pvalue3)
fit.medlb3 <- mxRun(Medlb3)
med.lb3 <- fit.medlb3@output$minimum
path <- fit.medlb3@output$estimate[c("pdigi","igipwb","pdpwb")]
load <- fit.medlb3@output$estimate[paste("l",1:13,sep="")]
cat(all(path<.5),all(path>-.5),all(load[1:11]<.99),all(load[1:11]>.01),
    all(load[12:13]< -.01),all(load[12:13]> -.99),
    mxEval(c(Model.res3,Model.objective),fit.medlb3)-c(0,crit3),"\n")

negMed <- mxAlgebra(-Model.Med,name="negMed")
Medub3 <- mxModel("Medub3",negMed,Model,mxAlgebraObjective("negMed"),pvalue3)
fit.medub3 <- mxRun(Medub3)
med.ub3<- -fit.medub3@output$minimum
path <- fit.medub3@output$estimate[c("pdigi","igipwb","pdpwb")]
load <- fit.medub3@output$estimate[paste("l",1:13,sep="")]
cat(all(path<.5),all(path>-.5),all(load[1:11]<.99),all(load[1:11]>.01),
    all(load[12:13]< -.01),all(load[12:13]> -.99),
    mxEval(c(Model.res3,Model.objective),fit.medub3)-c(0,crit3),"\n")

Medlb3 <- omxSetParameters(Medlb3,"pdpwb",free=F)
Medub3 <- omxSetParameters(Medub3,"pdpwb",free=F)

# use result from last iteration as starting value

Med.CRlb3 <- Med.CRub3 <- Med.lb.error <- Med.ub.error <- rep(NA,51)
Med.CRlb3[1] <- Med.CRub3[1] <- pdpwb.lb3.est["pdigi"]*pdpwb.lb3.est["igipwb"]
Med.CRlb3[51] <- Med.CRub3[51] <- pdpwb.ub3.est["pdigi"]*pdpwb.ub3.est["igipwb"]

pdpwb.values <- seq(pdpwb.lb3,pdpwb.ub3,length.out=51)
est.lb <- est.ub <- pdpwb.lb3.est
varn <- names(est.lb)

for (i in 2:50)
{
    Medlb3 <- omxSetParameters(Medlb3,varn,values=est.lb[varn])
    Medlb3 <- omxSetParameters(Medlb3,"pdpwb",values=pdpwb.values[i])
    fit <- mxRun(Medlb3,silent=T)
```

```
    m <- Med.CRlb3[i]<-fit@output$minimum
    est.lb <- fit@output$estimate

    Model2 <- mxModel("Model2",Model,mxConstraint(Model.Med==m,name="ConstMed"),
                mxAlgebraObjective("Model.objective"));
    Model2 <- omxSetParameters(Model2,labels=varn,values=est.lb[varn])
    Model2 <- omxSetParameters(Model2,"pdpwb",values=pdpwb.values[i],free=F)
    fit <- mxRun(Model2,silent=T)
    Med.lb.error[i] <- fit@output$minimum-crit3
}

for (i in 2:50)
{
    Medub3 <- omxSetParameters(Medub3,varn,values=est.ub[varn])
    Medub3 <- omxSetParameters(Medub3,"pdpwb",values=pdpwb.values[i])
    fit <- mxRun(Medub3,silent=T)
    m <- Med.CRub3[i] <- -fit@output$minimum
    est.ub <- fit@output$estimate

    Model2 <- mxModel("Model2",Model,mxConstraint(Model.Med==m,name="ConstMed"),
                mxAlgebraObjective("Model.objective"))
    Model2 <- omxSetParameters(Model2,labels=varn,values=est.ub[varn])
    Model2 <- omxSetParameters(Model2,"pdpwb",values=pdpwb.values[i],free=F)
    fit <- mxRun(Model2,silent=T)
    Med.ub.error[i] <- fit@output$minimum-crit3
}

# Plot of 95% profile likelihood-based CR

plot(c(Med.CRlb3,Med.CRub3),rep(pdpwb.values,2),
     xlab="Direct Effect of PD on PWB", ylab='Indirect Effect of PD on PWB')

cbind(Med.lb.error,Med.ub.error)
```

OpenMx error codes have been suppressed; error codes of 6 in this context may not accurately indicate an error. Instead, extremely small numbers in the difference between the -2 log-likelihood from the initial and constrained model solutions (the last line of code) assures proper estimates.

Confidence Regions for Empirical Example on Latent Curve Model

The second example involves a latent growth curve model published in Hancock and Choi (2006). Wald-type and profile likelihood-based CRs of varying confidence levels are displayed below for the relative gradient (RG) and the relative aperture variance (RAV). With increasing levels of coverage, the Wald-type CR strikingly departs from the profile likelihood-based CR.



References

Hancock, G. R., & Choi, J. (2006). A vernacular for linear latent growth models. *Structural Equation Modeling*, *13*, 352–377.

Schmitt, M. T., Branscombe, N. R., Kobrynowicz, D., & Owen, S. (2002). Perceiving discrimination agaist one's gender group has different implications for well-being in women and men. *Personality and Social Psychology Bulletin*, *28*, 197–210.