R Script

FIT FUNCTIONS R SCRIPT

# various utility functions for parameter recovery routines

# density function for Wald (unshifted)

dwald = function(x,gamma,alpha){

return((alpha/(sqrt(2\*pi\*x^3)))\*exp(-(alpha-gamma\*x)^2/(2\*x)))

}

# function to generate random shifted Wald data

# adapted from pp. 79-80, Dagpunar, J. (1988). Principles of Random Variate Generation. Clarendon Press, Oxford.

# code modified from Heathcote (2004)

rwald = function(n, gamma, alpha, theta) {

y2 = rchisq(n,1)

y2onm = y2/gamma

u = runif(n)

r1 = (2\*alpha + y2onm - sqrt(y2onm\*(4\*alpha+y2onm)))/(2\*gamma)

r2 = (alpha/gamma)^2/r1

ifelse(u < alpha/(alpha+gamma\*r1), theta+r1, theta+r2)

}

# basic function that returns Root Mean Squared Error

rmse = function(error){

sqrt(mean(error^2))

}

# negative log likelihood for shifted Wald

nll.wald = function(par,dat){

return(-sum(log(dwald(dat-par[3], gamma=par[1], alpha=par[2]))))

}

# Hierach. negative log likelihood for shifted Wald

nll.wald.h = function(par,dat){

n\_subjects <- dim(dat)[1]

n\_trials <- dim(dat)[2]

par <- exp(par)

gamma\_group\_mean <- par[1]

gamma\_group\_sd <- par[2]

alpha\_group\_mean <- par[3]

alpha\_group\_sd <- par[4]

theta\_group\_mean <- par[5]

theta\_group\_sd <- par[6]

gamma\_ind <- par[6 + 1:(n\_subjects-1)]

alpha\_ind <- par[6 + (n\_subjects-1) + 1:(n\_subjects-1)]

theta\_ind <- par[6 + 2\*(n\_subjects-1) + 1:(n\_subjects-1)]

gamma\_ind <- c(gamma\_ind, n\_subjects\*gamma\_group\_mean - sum(gamma\_ind))

alpha\_ind <- c(alpha\_ind, n\_subjects\*alpha\_group\_mean - sum(alpha\_ind))

theta\_ind <- c(theta\_ind, n\_subjects\*theta\_group\_mean - sum(theta\_ind))

theta\_mat<- matrix(rep(theta\_ind, each=n\_trials), n\_subjects, n\_trials, byrow=TRUE)

if(any(dat-theta\_mat <0)) {return(Inf)}

nll <- 0

for (s in 1:n\_subjects){

nll <- nll - sum(log(dwald(dat[s,]-theta\_ind[s], gamma=gamma\_ind[s], alpha=alpha\_ind[s])))

}

nll <- nll -sum(dnorm(gamma\_ind, mean=gamma\_group\_mean, sd=gamma\_group\_sd), log=TRUE)

nll <- nll -sum(dnorm(alpha\_ind, mean=alpha\_group\_mean, sd=alpha\_group\_sd), log=TRUE)

nll <- nll -sum(dnorm(theta\_ind, mean=theta\_group\_mean, sd=theta\_group\_sd), log=TRUE)

return(nll)

}

# Start point estimate for SW, based on first two moments

# assumes s = p\*min(x), where x is a data vector

# from Heathcote (2004)

waldinit = function(x, p = 0.9) {

theta = p\*min(x)

x = x - theta

gamma = sqrt(mean(x)/var(x))

alpha = gamma\*mean(x)

return(c(gamma,alpha,theta))

}

waldinit.h = function(x, p = 0.9) {

#input:

#x: matrix of rt w rows indicating subject and columns indicating trial

n\_subjects <- dim(x)[1]

n\_trials <- dim(x)[2]

theta\_ind <- p\*apply(x,1, min) #vector of shift parameters

theta\_group\_mean <- mean(theta\_ind) #group mean shift parameter

theta\_group\_sd <- sd(theta\_ind) #SD of shift across group

theta\_mat<- matrix(rep(theta\_ind, each=n\_trials), n\_subjects, n\_trials, byrow=TRUE)

#diff shift per person > create matrix

x = x - theta\_mat

gamma\_ind = sqrt(apply(x, 1, mean) / apply(x,1, var)) #gamma vector

gamma\_group\_mean <- mean(gamma\_ind)

gamma\_group\_sd <- sd(gamma\_ind)

alpha\_ind <- gamma\_ind \*apply(x, 1, mean) #alpha vector

alpha\_group\_mean <- mean(alpha\_ind) #group mean alpha

alpha\_group\_sd <- sd(alpha\_ind)

#to make sure you pull the right parameter, we have to be careful; pull group first

par <- c(gamma\_group\_mean, gamma\_group\_sd,

alpha\_group\_mean, alpha\_group\_sd,

theta\_group\_mean, theta\_group\_sd,

gamma\_ind[-n\_subjects], alpha\_ind[-n\_subjects], theta\_ind[-n\_subjects])

return(log(par))

}

CMLE R SCRIPT

#source("fitFunctions.R")

##############################

# define simulation parameters

##############################

nSub = 20

nTrials = 20

# parent distribution means and sds (from F, Vick, Bowman, 2018)

G = 3.91 # gamma

G.sd = 0.70

A = 0.92 # alpha

A.sd = 0.17

H = 0.32 # theta

H.sd = 0.05

# each simulation run starts HERE

# build simulated RT matrix

# rows = subjects, columns = trials

rts = matrix(0, nrow=nSub, ncol=nTrials)

# build matrix to store "target" parameters

# rows = subjects, columns = gamma, alpha, theta

targets = matrix(0, nrow=nSub, ncol=3)

# randomly draw target shifted Wald parameters from parent distribution

# (one unique parameter value for each subject)

for (i in 1:nSub){

targets[i,1] = rnorm(1, mean=G, sd=G.sd) # draw random gamma

targets[i,2] = rnorm(1, mean=A, sd=A.sd) # draw random alpha

targets[i,3] = rnorm(1, mean=H, sd=H.sd) # draw random theta

}

# from these target SW values, generate the distribution of observed RTs for each subject

for (i in 1:nSub){

rts[i,] = rwald(nTrials,

gamma = targets[i,1],

alpha = targets[i,2],

theta = targets[i,3])

}

# now let's use CMLE to fit shifted Wald distribution to each row of observed RTs

# first, we'll store these estimates in a matrix

predictions = matrix(0, nrow=nSub, ncol=3)

# now do the fits (one per subject)

for (i in 1:nSub){

fit = optim(waldinit(rts[i,]), nll.wald, dat=rts[i,])

predictions[i,] = fit$par

}

# now lets compare the predictions (fits) to the targets

# mean of parameter estimates

apply(predictions, 2, mean)

# RMSD

apply(predictions-targets, 2, rmse)

# mean bias

apply(predictions-targets, 2, mean)

# correlation

diag(cor(predictions,targets))

HMLE R SCRIPT

#source("fitFunctions.R")

nSub = 20

nTrials = 20

# parent distribution means and sds (from F, Vick, Bowman, 2018)

G = 3.91 # gamma

G.sd = 0.70

A = 0.92 # alpha

A.sd = 0.17

H = 0.32 # theta

H.sd = 0.05

#matrices

rmse\_mat = matrix(NA, 200, 3+(3\*nSub))

mb\_mat = matrix(NA, 200, 3+(3\*nSub))

ind\_means\_mat = matrix(NA, 200, (3\*nSub))

for (x in 1:200){

# each simulation run starts HERE

# build simulated RT matrix

# rows = subjects, columns = trials

rts = matrix(0, nrow=nSub, ncol=nTrials)

# build matrix to store "target" parameters

# rows = subjects, columns = gamma, alpha, theta

targets = matrix(0, nrow=nSub, ncol=3)

# randomly draw target shifted Wald parameters from parent distribution

# (one unique parameter value for each subject)

for (i in 1:nSub){

targets[i,1] = rnorm(1, mean=G, sd=G.sd) # draw random gamma

targets[i,2] = rnorm(1, mean=A, sd=A.sd) # draw random alpha

targets[i,3] = rnorm(1, mean=H, sd=H.sd) # draw random theta

}

# from these target SW values, generate the distribution of observed RTs for each subject

for (i in 1:nSub){

rts[i,] = rwald(nTrials,

gamma = targets[i,1],

alpha = targets[i,2],

theta = targets[i,3])

}

# now let's use CMLE to fit shifted Wald distribution to each row of observed RTs

# first, we'll store these estimates in a matrix

#predictions = matrix(0, nrow=nSub, ncol=3)

# now do the fits (one per subject)

# mean of parameter estimates

#apply(predictions, 2, mean)

# RMSD

#apply(predictions-targets, 2, rmse)

# mean bias

#apply(predictions-targets, 2, mean)

# correlation

#diag(cor(predictions,targets))

#HIERARCHICAL MODEL FIT

fit.h = optim(waldinit.h(rts), nll.wald.h, dat=rts)

# now lets compare the predictions (fits) to the targets

#will need to do

par <- exp(fit.h$par)

gamma\_group\_mean <- par[1]

gamma\_group\_sd <- par[2]

alpha\_group\_mean <- par[3]

alpha\_group\_sd <- par[4]

theta\_group\_mean <- par[5]

theta\_group\_sd <- par[6]

gamma\_ind <- par[6 + 1:(nSub-1)]

alpha\_ind <- par[6 + (nSub-1) + 1:(nSub-1)]

theta\_ind <- par[6 + 2\*(nSub-1) + 1:(nSub-1)]

gamma\_ind <- c(gamma\_ind, nSub\*gamma\_group\_mean - sum(gamma\_ind))

alpha\_ind <- c(alpha\_ind, nSub\*alpha\_group\_mean - sum(alpha\_ind))

theta\_ind <- c(theta\_ind, nSub\*theta\_group\_mean - sum(theta\_ind))

##RMSE

#g level error

gg\_rmse = sqrt((G- gamma\_group\_mean)^2)

ga\_rmse = sqrt((A- alpha\_group\_mean)^2)

gh\_rmse = sqrt((H- theta\_group\_mean)^2)

#i level error

ig\_rmse = sqrt((targets[,1]-gamma\_ind)^2)

ia\_rmse = sqrt((targets[,2]-alpha\_ind)^2)

ih\_rmse = sqrt((targets[,3]-theta\_ind)^2)

rmse\_mat[x,] = c(gg\_rmse,ga\_rmse, gh\_rmse, ig\_rmse, ia\_rmse, ih\_rmse)

##MEAN BIAS

#g level mb

gg\_mb = -1\*(G- gamma\_group\_mean)

ga\_mb = -1\*(A- alpha\_group\_mean)

gh\_mb = -1\*(H- theta\_group\_mean)

#i level mb

ig\_mb = -1\*(targets[,1]-gamma\_ind)

ia\_mb = -1\*(targets[,2]-alpha\_ind)

ih\_mb = -1\*(targets[,3]-theta\_ind)

mb\_mat[x,] = c(gg\_mb,ga\_mb, gh\_mb, ig\_mb, ia\_mb, ih\_mb)

#MEANS (individual level)

ind\_means\_mat[x,] = c(gamma\_ind, alpha\_ind, theta\_ind)

}

write.csv(rmse\_mat, file= "hrmse.csv")

write.csv(mb\_mat, file= "hmb.csv")

write.csv(ind\_means\_mat, file = "MeansMat.csv")