**Chapter 19 Online Appendices: Fitting RSM to ROC data**

# Table of contents

1. Online Appendix 19A: RSM fits
2. Online Appendix 19.B: Bootstrap analysis: inter and intra correlations
3. Online Appendix 19.C: Sample size estimation
4. References

# Online Appendix 19A: RSM fits

Following is a partial listing of mainRsmFits.R that performs unconstrained RSM fitting to ROC datasets, computes RSM and PROPROC AUCs, and displays corresponding ROC plots and operating points.

## Online Appendix 19.A.1 Code Listing (partial)

# RSM fits vs. PROPROC and CBM fits;

# Windows proproc results must be saved to MRMCRuns directory

rm(list = ls()) # mainRSM.R

...

fileNames <- c("TONY", "VD", "FR", "FED", "JT", "MAG", "OPT", "PEN", "NICO",

"RUS", "DOB1", "DOB2", "DOB3", "FZR")

fileName <- fileNames[fileNames == "TONY"]

...

# retrieve PROPROC parameters

ret <- ProprocFits(paste0(fileName, "\_MRMC")) # this contains values generated by Windows DBM-MRMC

c1 <- ret$c1;da <- ret$da;aucProproc <- c1\*0;aucRsm <- c1\*0;aucCBM <- c1\*0

S <- rep(NA, I\*J);AucC <- rep(NA, I\*J)

s <- 1 # following line prints a header row

cat("i, j, mu, lambdaP, nuP, c, da, alpha, muCbm, aucRsm, aucProproc, aucCbm","\n")

retFileName <- paste0("ANALYZED/RSM/", "saveRetRoc", fileName)

if (!file.exists(retFileName)){

retSmRoc <- list()

for (i in 1:I){

for (j in 1:J){

# Metz and Pan Journal of Mathematical Psychology 43, 1?33 (1999)

rho2 <- -(1-c1[i,j]^2)/(1+c1[i,j]^2)

corr <- diag(2)

corr[lower.tri(corr)] <- rho2

corr[upper.tri(corr)] <- rho2

lower <- rep(-Inf,2)

upper <- c(-da[i,j]/sqrt(2),0)

mean <- rep(0,2)

aucProproc[i,j] <- pnorm(da[i,j]/sqrt(2))+2\*pmvnorm(lower, upper, mean, corr) #Eqn. 36 Metz

tempRoc <- FitRsmRoc(frocData, i, j) # fit to RSM

mu <- as.numeric(tempRoc$mu$mu)

lambdaP <- as.numeric(tempRoc$lambdaP$lambdaP)

nuP <- as.numeric(tempRoc$nuP$nuP)

aucRsm[i,j] <- tempRoc$AUC$AUC

empOp <- GetOperatingPoints(rocData, i, j, opChType = "ROC")

fpf <- empOp$FPF; tpf <- empOp$TPF

retCbm <- FitCbmRoc(rocData,i,j) # fit to CBM

compPlot <- PlotRsmPropCbm(

mu, lambdaP, nuP, nLesDistr, c1[i, j], da[i, j],

retSmRoc[[s]]$muCbm, retSmRoc[[s]]$alpha,

fpf, tpf, i, j, K1, K2, c(1, length(fpf)))

print(compPlot)

retSmRoc[[s]] <- as.list(

c(tempRoc,

list(

nLesDistr = nLesDistr, aucProp = aucProproc[i,j],

da = da[i, j], c1 = c1[i, j],

alpha=retCbm$alpha, muCbm=retCbm$mu, aucCbm=retCbm$AUC)))

s <- s + 1

}

}

stop("safety stop...") # comment this to write a new results file or overwrite an exitsting

save(retSmRoc, file = retFileName)

}else{

load(retFileName)

s <- 1

for (i in 1:I){

for (j in 1:J){

mu <- as.numeric(retSmRoc[[s]]$mu$mu)

lambdaP <- as.numeric(retSmRoc[[s]]$lambdaP$lambdaP)

nuP <- as.numeric(retSmRoc[[s]]$nuP$nuP)

nLesDistr <- retSmRoc[[s]]$nLesDistr

aucRsm[i, j] <- retSmRoc[[s]]$AUC$AUC

aucProproc[i, j] <- retSmRoc[[s]]$aucProp

empOp <- GetOperatingPoints(rocData, i, j, opChType = "ROC")

fpf <- empOp$FPF; tpf <- empOp$TPF

compPlot <- PlotRsmPropCbm(

mu, lambdaP, nuP, nLesDistr, c1[i, j], da[i, j],

retSmRoc[[s]]$muCbm, retSmRoc[[s]]$alpha,

fpf, tpf, i, j, K1, K2, c(1, length(fpf)))

print(compPlot)

cat(i, j, mu, lambdaP, nuP, c1[i,j], da[i,j],retSmRoc[[s]]$alpha, retSmRoc[[s]]$muCbm,

aucRsm[i,j], aucProproc[i,j], retSmRoc[[s]]$aucCbm, "\n")

s <- s + 1

}

}

}

There are 14 datasets with abbreviated names in line 20 - 21. Line 22 selects the TONY dataset. To save vertical real estate in the main file listing, the actual loading is done inside function loadDataFile() called at line 24, which returns an RJafroc dataset object frocData, which is converted to an ROC dataset object rocData using function FROC2HrROC() at line 25. Line 28 calculates the lesion distribution vector nLesDistr; recall that the RSM-predicted ROC-AUC depends on the number of lesion per diseased case, because inferred *TPF* depends on this information. Insert a break point at line 34 and click source.

Highlight nLesDistr and click on Run. Code snippet Online Appendix 19.A.2 shows that the dataset has 83 diseased cases with 1 lesion per case and 6 diseased cases with 2 lesions per case.

## Online Appendix 19.A.2: Code snippet

Browse[2]> nLesDistr

[,1] [,2]

[1,] 1 83

[2,] 2 6

Use the Environment panel to confirm that rocData is indeed an ROC dataset object with two modalities, 5 readers, 89 diseased cases and 185 – 89 = 96 non-diseased cases. The corresponding frocData from which it is derived has a maximum of 3 NL marks per case. Line 30 checks if the appropriate file with the extension .lrc exists, and if not it creates it; this little diversion is needed to run DBM-MRMC on the .lrc file with the PROPROC figure of merit selected. Since this has already been done for you, the file exists, and line 31 is skipped. Click the "*go into function*" button to enter the function ProprocFits()in debug mode.

## Online Appendix 19.A.3: Code Listing

ProprocFits <- function(filename) {

mrmcFile <- paste0("./MRMCRuns/", fileName, "\_MRMC proproc area pooled.csv")

if (!file.exists(mrmcFile)) stop("need to run proproc for this dataset")

proprocRet <- read.csv(mrmcFile)

c1 <- matrix(data = proprocRet$c, nrow = length(unique(proprocRet$T)), ncol = length(unique(proprocRet$R)), byrow = TRUE)

da <- matrix(data = proprocRet$d\_a, nrow = length(unique(proprocRet$T)), ncol = length(unique(proprocRet$R)), byrow = TRUE)

return (list(c1 = c1, da = da))

}

The code pointer should be at the first line of the listing in Online Appendix 19.A.3. On clicking Next the code pointer moves to line 2, which constructs the name of the output file created by *PROPROC*. Click Next: line 3 checks for the existence of the output file and if not present execution stops with a helpful message. Click Next: line 4 reads the output file created by *PROPROC* using the R function read.csv(). CSV stands for comma-separated variables. Lines 5 and 6 read the contents of this file and saves values of the extracted c1 (using c as the variable name leads to trouble as it clashes with the c() function of R) and da parameters. For example, the values print out as (highlight and Run):

## Online Appendix 19.A.4: Code Listing

Browse[3]> c1

[,1] [,2] [,3] [,4] [,5]

[1,] -0.13228036 -0.08696513 -0.1444419 0.08046016 0.2225588

[2,] -0.08174248 0.04976448 -0.1326126 0.11822263 0.0781033

Browse[3]> da

[,1] [,2] [,3] [,4] [,5]

[1,] 1.1972393 1.7711756 1.481935 1.513757 1.7401572

[2,] 0.6281251 0.9738786 1.155871 1.620176 0.8928816

These are the *PROPROC* parameters for the two modalities and five readers in the study. Click the "*get me out of this loop*" button and the code pointer should be at line 35 in the main file. The values of c1 and da are extracted from the returned list variable ret.

*The following lines are only executed if one forces entry into the for-loop beginning at line 40*. Lines 42 and 43 set up two for-loops to analyze each modality and each reader. Lines 44 – 52 implements Eqn. 36, the formula for the PROPROC-AUC, in the Metz and Pan paper1. Keep clicking on Next until the code pointer has advanced to line 53. Highlight proprocAuc[i,j] and click on Run; one should see 0.8014164, the first entry under PROPROC-AUC in Table 19.5.1. Line 53 fits the RSM model to the rocData dataset object for modality i and reader j. Click Next, after a brief pause the code pointer advances to line 54. The next few lines extract the search model parameters followed by the AUC. Line 58 uses the function GetOperatingPoints() to get the ROC operating points. Keep clicking on Next until the code pointer is at line 58. Print out aucRSM[i,j] one getting 0.8128219, the first entry under RSM-AUC in Table 19.5.1. Lines 61 - 65 plot the RSM, the PROPROC and the CBM-predicted ROC curves and the observed operating points. Click the "get me out of this loop" button in the console window. The code pointer should advance to line 74, and there is output in the window from lines 68 – 70, which was summarized in Table 19.5.1.

This completes the explanation of the code for the TONY dataset. The reader should experiment with all datasets in the distribution and confirm the tables and figures in §19.5.

# Online Appendix 19.B: Bootstrap analysis

The code to generate Table 19.5.5 is shown below, file mainBsRSMVsOthers.R. Line 13 initializes clParms, for cluster parameters, to NULL. This will be appended to to make a long list containing the results for each dataset, line 62 - 64. Lines 14 – 22 initializes arrays to hold needed quantities for each datasets. For example, avgAucPro will contain 14 values corresponding to modality and reader averaged PROPROC AUCs, one for each dataset; rhoNupRsmAlphaCbm will contain 14 correlation values , one for each dataset; avgSlopeCbmRsm will contain 14 constrained fit slopes, one for each dataset, between CBM AUCs and RSM AUCs and avgR2ProRsm will contain 14 R2 values, one for each dataset, between PROPROC AUCs and RSM AUCs.

Lines 23 – 68 retrieve values for each dataset and computes averages without bootstrapping. These are the values shown in the second to last row of Table 19.5.4. Lines 79 – 128 performs bootstrap sampling using cluster programming. Line 84 starts the foreach loop, corresponding to each thread; the returned values are saved to a varaible called bootStrapResults. Lines 85 – 88 initialize arrays with NAs to hold the 200 bootstrap values. For example, avgSlopeProRsm will contain 200 x 14 bootstrap values for the slope between PROPROC AUC and RSM AUC, 200 values for each dataset. It is filled in at line 101. Line 125 – 126 concatenates six variables; for example the first variable is mean(avgSlopeProRsm), which will return 200 values of the slope between PROPROC AUC and RSM AUC. Lines 130 – 135 extract the relevant values. For example, consider:

avgSlopeProRsm <- data.frame(value = bootStrapResults[ , 1])

This line extracts the average slopes as a dataframe, a construct useful in the histogram plotting routine that follows in lines 137 – 139. Line 140 computes the empirical 95% confidence interval and line 141 prints it.

## Online Appendix 19.C.1 Code Listing

rm(list = ls()) # mainBsRSMVsOthers.R # compares RSM to PROPROC and CBM

library(foreach)

library(RJafroc)

library(doRNG)

library(doParallel)

library(ggplot2)

source("loadDataFile.R")

pathName <- "../../../06 E Appendices/E23 Datasets/"

type <- "pearson"

fileNames <- c("TONY", "VD", "FR", "FED", "JT", "MAG", "OPT", "PEN", "NICO",

"RUS", "DOB1", "DOB2", "DOB3", "FZR")

clParms <- NULL # parameters passed to cluster

avgAucPro <- array(dim = length(fileNames))

avgAucRsm <- array(dim = length(fileNames))

avgAucCbm <- array(dim = length(fileNames))

avgSlopeProRsm <- array(dim = length(fileNames))

avgR2ProRsm <- array(dim = length(fileNames))

avgSlopeCbmRsm <- array(dim = length(fileNames))

avgR2CbmRsm <- array(dim = length(fileNames))

rhoMuRsmMuCbm <- array(dim = length(fileNames))

rhoNupRsmAlphaCbm <- array(dim = length(fileNames))

for (f in 1:length(fileNames)){

retFileName <- paste0("ANALYZED/RSM/", "saveRetRoc", fileNames[f])

if (file.exists(retFileName)){

load(retFileName)

frocData <- loadDataFile(fileNames[f], pathName)

I <- length(frocData$modalityID)

J <- length(frocData$readerID)

aucRsm <- array(dim = c(I, J));aucCbm <- array(dim = c(I, J));aucPro <- array(dim = c(I, J))

muRsm <- array(dim = c(I, J));muCbm <- array(dim = c(I, J))

nupRsm <- array(dim = c(I, J));alphaCbm <- array(dim = c(I, J))

s <- 1

for (i in 1:I){

for (j in 1:J){

aucRsm[i, j] <- as.numeric(retSmRoc[[s]]$AUC$AUC)

aucPro[i, j] <- as.numeric(retSmRoc[[s]]$aucProp)

aucCbm[i, j] <- retSmRoc[[s]]$aucCbm

muRsm[i, j] <- as.numeric(retSmRoc[[s]]$mu$mu)

nupRsm[i, j] <- as.numeric(retSmRoc[[s]]$nuP$nuP)

alphaCbm[i, j] <- retSmRoc[[s]]$alpha

muCbm[i, j] <- retSmRoc[[s]]$muCbm

s <- s + 1

}

}

avgAucRsm[f] <- mean(aucRsm);avgAucPro[f] <- mean(aucPro);avgAucCbm[f] <- mean(aucCbm)

rhoMuRsmMuCbm[f] <- cor(as.vector(muRsm), as.vector(muCbm),method = "pearson")

rhoNupRsmAlphaCbm[f]<- cor(as.vector(nupRsm), as.vector(alphaCbm),method = "pearson")

cat("rhoMuRsmMuCbm[f]=",rhoMuRsmMuCbm[f],

", rhoNupRsmAlphaCbm[f]=",rhoNupRsmAlphaCbm[f],"\n")

df <- data.frame(aucRsm = as.vector(aucRsm), aucPro = as.vector(aucPro))

m <- lm(aucPro ~ 0 + aucRsm, data = df)

avgSlopeProRsm[f] <- coef(m)

avgR2ProRsm[f] <- summary(m)$r.squared

df <- data.frame(aucRsm = as.vector(aucRsm), aucCbm = as.vector(aucCbm))

m <- lm(aucCbm ~ 0 + aucRsm, data = df)

avgSlopeCbmRsm[f] <- coef(m)

avgR2CbmRsm[f] <- summary(m)$r.squared

clParms <- c(clParms, list(list(aucRsm = aucRsm, aucCbm = aucCbm, aucPro = aucPro,

nupRsm = nupRsm, alphaCbm = alphaCbm,

muRsm = muRsm, muCbm = muCbm)))

}else{

stop("Results file does not exist. Must analyze all datasets before running this.")

}

}

cat(

" avg aucRsm =", mean(avgAucRsm),

", avg aucPro =", mean(avgAucPro),

", avg aucCbm =", mean(avgAucCbm),

", avgSlopeCbmRsm =", mean(avgSlopeCbmRsm),", avg R2CbmRsm =", mean(avgR2CbmRsm),

", avgSlopeProRsm =", mean(avgSlopeProRsm),", avg R2ProRsm =", mean(avgR2ProRsm),

", avg rhoMuRsmMuCbm =", mean(rhoMuRsmMuCbm),", avg rhoNupRsmAlphaCbm =", mean(rhoNupRsmAlphaCbm),

"\n"

)

names(clParms) <- fileNames

cl <- makeCluster(detectCores())

registerDoParallel(cl)

B <- 200

seed <- 1

bootStrapResults <- foreach (b = 1:B, .options.RNG = seed, .combine = "rbind", .packages = "RJafroc") %dorng% {

avgSlopeCbmRsm <- rep(NA, length(fileNames));avgR2CbmRsm <- rep(NA, length(fileNames))

avgSlopeProRsm <- rep(NA, length(fileNames));avgR2ProRsm <- rep(NA, length(fileNames))

rhoMuRsmMuCbm <- rep(NA, length(fileNames));rhoNupRsmAlphaCbm <- rep(NA, length(fileNames))

for (f in 1:length(fileNames)){

retFileName <- paste0("ANALYZED/RSM/", "saveRetRoc", fileNames[f])

if (file.exists(retFileName)){

load(retFileName)

frocData <- loadDataFile(fileNames[f], pathName)

I <- length(frocData$modalityID);J <- length(frocData$readerID)

jBs <- ceiling(runif(J) \* J) # here is were we bootstap readers

# constrained fit thru origin; aucPro vs. aucRsm

df1 <- data.frame(aucRsm = as.vector(clParms[[fileNames[f]]]$aucRsm[ , jBs]),

aucPro = as.vector(clParms[[fileNames[f]]]$aucPro[, jBs]))

m <- lm(aucPro ~ 0 + aucRsm, data = df1)

avgSlopeProRsm[f] <- coef(m)

avgR2ProRsm[f] <- summary(m)$r.squared

# constrained fit thru origin; aucCbm vs. aucRsm

df2 <- data.frame(aucRsm = as.vector(clParms[[fileNames[f]]]$aucRsm[ , jBs]),

aucCbm = as.vector(clParms[[fileNames[f]]]$aucCbm[, jBs]))

m <- lm(aucCbm ~ 0 + aucRsm, data = df2)

avgSlopeCbmRsm[f] <- coef(m)

avgR2CbmRsm[f] <- summary(m)$r.squared

# correlation between muRsm and muCbm

df1 <- data.frame(muRsm = as.vector(clParms[[fileNames[f]]]$muRsm[ , jBs]),

muCbm = as.vector(clParms[[fileNames[f]]]$muCbm[, jBs]))

rhoMuRsmMuCbm[f] <- cor(df1$muRsm, df1$muCbm,method = "pearson")

# correlation between nupRsm and alphaCbm

df2 <- data.frame(nupRsm = as.vector(clParms[[fileNames[f]]]$nupRsm[ , jBs]),

alphaCbm = as.vector(clParms[[fileNames[f]]]$alphaCbm[, jBs]))

rhoNupRsmAlphaCbm[f] <- cor(df2$nupRsm, df2$alphaCbm,method = "pearson")

}else{

stop("Results file does not exist.")

}

}

c(mean(avgSlopeProRsm), mean(avgR2ProRsm),

mean(avgSlopeCbmRsm), mean(avgR2CbmRsm), mean(rhoMuRsmMuCbm), mean(rhoNupRsmAlphaCbm))

}

stopCluster(cl)

avgSlopeProRsm <- data.frame(value = bootStrapResults[ , 1])

avgR2ProRsm <- data.frame(value = bootStrapResults[ , 2])

avgSlopeCbmRsm <- data.frame(value = bootStrapResults[ , 3])

avgR2CbmRsm <- data.frame(value = bootStrapResults[ , 4])

rhoMuRsmMuCbm <- data.frame(value = bootStrapResults[ , 5])

rhoNupRsmAlphaCbm <- data.frame(value = bootStrapResults[ , 6])

histogram <- ggplot(avgSlopeProRsm, aes(x = value)) + geom\_histogram(color = "white") +

xlab("avgSlopeProRsm")

print(histogram)

ciAvgSlopeProRsm <- quantile(avgSlopeProRsm$value, c(0.025, 0.975), type = 1)

cat("The empirical 95% CI of avgSlopeProRsm is", paste(ciAvgSlopeProRsm, collapse = ", "), "\n")

histogram <- ggplot(avgSlopeCbmRsm, aes(x = value)) + geom\_histogram(color = "white") +

xlab("avgSlopeCbmRsm")

print(histogram)

ciAvgSlopeCbmRsm <- quantile(avgSlopeCbmRsm$value, c(0.025, 0.975), type = 1)

cat("The empirical 95% CI of avgSlopeCbmRsm is", paste(ciAvgSlopeCbmRsm, collapse = ", "), "\n")

histogram <- ggplot(rhoMuRsmMuCbm, aes(x = value)) + geom\_histogram(color = "white") +

xlab("rhoMuRsmMuCbm")

print(histogram)

ciRhoMuRsmMuCbm <- quantile(rhoMuRsmMuCbm$value, c(0.025, 0.975), type = 1)

cat("The empirical 95% CI of rhoMuRsmMuCbm is", paste(ciRhoMuRsmMuCbm, collapse = ", "), "\n")

histogram <- ggplot(rhoNupRsmAlphaCbm, aes(x = value)) + geom\_histogram(color = "white") +

xlab("rhoNupRsmAlphaCbm")

print(histogram)

ciRhoNupRsmAlphaCbm <- quantile(rhoNupRsmAlphaCbm$value, c(0.025, 0.975), type = 1)

cat("The empirical 95% CI of rhoNupRsmAlphaCbm is", paste(ciRhoNupRsmAlphaCbm, collapse = ", "), "\n")

# Online Appendix 19.B: Bootstrap analysis of AUC slopes

The code to generate Table 19.5.4 is shown below, file mainBsRSMVsOthers.R. Line 13 initializes clParms, for clust

# Online Appendix 19.C: Sample size estimation

The following is a listing of sample size estimation using both inferred ROC-AUC and wAFROC-AUC as figures of merit. The code listing of the relevant file mainwAFROCPowerDBMH.R follows.

### Online Appendix 19.C.1: Code listing

rm(list = ls()) #mainwAFROCPowerDBMH.R

library(ggplot2)

library(RJafroc)

source("loadDataFile.R")

# included datasets

fileName <- c("TONY", "VD", "FR", "FED", "JT", "MAG", "OPT", "PEN", "NICO", "RUS", "DOB1", "DOB2", "FZR" )

fileName <- fileName[4]

cat("fileName = ", fileName,"\n")

frocData <- loadDataFile(fileName)

retFileName <- paste0("ANALYZED/", "saveRetRoc", fileName)

if (!file.exists((retFileName))){

I <- length(frocData$modalityID)

J <- length(frocData$readerID)

s <- 1

for (i in 1:I){

for (j in 1:J){

cat("i = ", i, ", j = ", j, "\n")

CBM <- CBMFitR(rocData, i, j)

tempRoc <- FitRsmRocCurve(rocData, i, j, AUCCbm = CBM$AUC, zetaCbm = CBM$cutoffs, nLesDistr = nLesDistr)

retSmRoc[[s]] <- as.list(c(tempRoc,

list(nLesDistr = nLesDistr, CBMAUC = CBM$AUC, CBMmu = CBM$mu, CBMalpha = CBM$alpha, CBMcutoffs = CBM$cutoffs, i = i, j = j)))

s <- s + 1

}

}

save(retSmRoc, file = retFileName)

} else {

load(retFileName) # loads object retSmRoc, i.e., ROC data previously analyzed by RSM; this has parameter values

}

i1 <- 1;i2 <- 2 # FED data has 5 modalities; we choose to analyze the first two

cat("NH i1 = ", i1, "NH i2 = ", i2, "\n")

selectJ <- c(1, 2, 3, 4)

frocData <- ExtractDataset(frocData, trts = c(i1, i2), rdrs = selectJ)

J <- length(frocData$readerID)

K <- dim(frocData$NL)[3]

mu1 <- rep(NA, J);mu2 <- rep(NA, J);nu1 <- rep(NA, J);nu2 <- rep(NA, J);lambda1 <- rep(NA, J);lambda2 <- rep(NA, J)

S <- length(retSmRoc)

for (s in 1:S){

if (retSmRoc[[s]]$j %in% selectJ){

if (retSmRoc[[s]]$i == i1){

mu1[retSmRoc[[s]]$j == selectJ] <- as.numeric(retSmRoc[[s]]$mu$mu)

lambda1[retSmRoc[[s]]$j == selectJ] <- as.numeric(retSmRoc[[s]]$lambda$lambda)

nu1[retSmRoc[[s]]$j == selectJ] <- as.numeric(retSmRoc[[s]]$nu$nu)

}else if (retSmRoc[[s]]$i == i2){

mu2[retSmRoc[[s]]$j == selectJ] <- as.numeric(retSmRoc[[s]]$mu$mu)

lambda2[retSmRoc[[s]]$j == selectJ] <- as.numeric(retSmRoc[[s]]$lambda$lambda)

nu2[retSmRoc[[s]]$j == selectJ] <- as.numeric(retSmRoc[[s]]$nu$nu)

}

}

}

mu <- rbind(mu1, mu2);lambda <- rbind(lambda1, lambda2);nu <- rbind(nu1, nu2)

muMed <- median(mu) # instead of average, use median to get representative value over whole dataset

nuMed <- median(nu) # do:

lambdaMed <- median(lambda) # do:

# construct lesion weights, assuming equally weighted lesions

nLesDistr <- retSmRoc[[1]]$nLesDistr

lesionWeights <- matrix(-Inf, nrow = nrow(nLesDistr), ncol = nrow(nLesDistr))

for (l in 1:nrow(nLesDistr)){

nLes <- nLesDistr[l, 1]

lesionWeights[l, 1:nLes] <- 1/nLes

}

# calculate NH values for ROC-AUC and wAFROC-AUC

aucRocNH <- RsmOperatingCharacteristics(muMed, lambdaMed, nuMed,

lesionDistribution = nLesDistr, lesionWeights = lesionWeights, type = "ROC")$aucROC

aucAfrocNH <- RsmOperatingCharacteristics(muMed, lambdaMed, nuMed,

lesionDistribution = nLesDistr, lesionWeights = lesionWeights, type = "wAFROC")$aucwAFROC

# following code calculates ROC-ES and wAFROC-ES

deltaMu <- seq(0.01, 0.2, 0.01) # values of deltaMu to scan below

esRoc <- array(dim = length(deltaMu));eswAfroc <- array(dim = length(deltaMu))

for (i in 1:length(deltaMu)) {

esRoc[i] <- RsmOperatingCharacteristics(muMed + deltaMu[i], lambdaMed, nuMed, lesionDistribution =

nLesDistr, lesionWeights = lesionWeights, type = "ROC")$aucROC - aucRocNH

eswAfroc[i] <- RsmOperatingCharacteristics(muMed+ deltaMu[i], lambdaMed, nuMed, lesionDistribution =

nLesDistr, lesionWeights = lesionWeights, type = "wAFROC")$aucwAFROC - aucAfrocNH

#cat("ES ROC, wAFROC = ", esRoc[i], eswAfroc[i],"\n")

}

#cat("\n")

a<-lm(eswAfroc~-1+esRoc) # fit values to straight line thru origin

effectSizeROC <- seq(0.01, 0.1, 0.01)

effectSizewAFROC <- effectSizeROC\*a$coefficients[1]

JTest <- 5;KTest <- 100

varCompROC <- DBMHAnalysis(frocData, fom = "HrAuc", option = "RRRC")$varComp

varCompwAFROC <- DBMHAnalysis(frocData, fom = "wAFROC", option = "RRRC")$varComp

cat("JTest = ", JTest, "KTest = ", KTest, "\n")

powerROC <- array(dim = length(effectSizeROC));powerwAFROC <- array(dim = length(effectSizeROC))

for (i in 1:length(effectSizeROC)) {

varYTR <- varCompROC$varComp[3]

varYTC <- varCompROC$varComp[4]

varYEps <- varCompROC$varComp[6]

powerROC[i] <- SsPowerGivenJK(JTest, KTest, alpha = 0.05, effectSize = effectSizeROC[i], option = "RRRC",

method = "DBMH", varYTR = varYTR, varYTC = varYTC, varYEps = varYEps)

varYTR <- varCompwAFROC$varComp[3]

varYTC <- varCompwAFROC$varComp[4]

varYEps <- varCompwAFROC$varComp[6]

powerwAFROC[i] <- SsPowerGivenJK(JTest, KTest, alpha = 0.05, effectSize = effectSizewAFROC[i], option = "RRRC",

method = "DBMH", varYTR = varYTR, varYTC = varYTC, varYEps = varYEps)

cat("ROC effect size = ,", effectSizeROC[i], "wAFROC effect size = ,", effectSizewAFROC[i],

", Statistical power ROC, wAFROC:", powerROC[i], ",", powerwAFROC[i], "\n")

}

df <- data.frame(esRoc = esRoc, eswAfroc = eswAfroc)

p <- ggplot(data = df, aes(x = esRoc, y = eswAfroc)) +

geom\_smooth(method = "lm", se = FALSE, color = "black", formula = y ~ x) +

geom\_point()

print(p)

df <- data.frame(powerROC = powerROC, powerwAFROC = powerwAFROC)

p <- ggplot(mapping = aes(x = powerROC, y = powerwAFROC)) +

geom\_line(data = df, size = 0.5)

print(p)

Line 8 defines the dataset to be loaded, namely the "FED" dataset, which is a 5-modality 4-radiologist FROC dataset. Place a break point at line 32 and click Source. Line 34 selects all four radiologists in the dataset and line 35 uses the ExtractDataset() function to extract data corresponding to the first two modalities only, as these are the ones regarded as NH modalities. Keep clicking on Next until the code pointer advances to line 36. Examination of the Environment panel reveals that now frocData is a 2-modality 4-reader dataset with 100 non-diseased and 100 diseased cases. Line 39 assigns memory to hold the parameters  where the appendage "1" applies to modality 1 and "2" applies to modality 2. Keep clicking on Next until the code pointer advances to line 42. Confirm (highlight and click Run) that the value of S is 20, since retSmRoc is a list with 20 elements; this comes from the 5 modalities x 4 readers.

Click Next to bring the code-pointer to line 43. This line looks complicated: *carefully* highlight retSmRoc[[s]] and click Run, Online Appendix 19.C.2 (the author has compressed the listing be deleting blank lines) and an RSM predicted ROC plot and overlaid operating points for the first reader in the first modality is displayed in the Plots panel.

### Online Appendix 19.C.2: Code snippet

Browse[2]> retSmRoc[[s]]

$plots

$plots[[1]]

$mu

mu Modality Reader

1 3.67597138794971 1 1

$lambda

lambda Modality Reader

1 36.5574221504606 1 1

$lambdaP

lambdaP Modality Reader

1 9.94496917748065 1 1

$nu

nu Modality Reader

1 0.415773740358019 1 1

$nuP

nuP Modality Reader

1 0.783111606500295 1 1

$AUC

AUC Modality Reader

1 0.902039601221186 1 1

$zetas

zetaFwd1 zetaFwd2 zetaFwd3 zetaFwd4 zetaFwd5

1.500923 1.699539 2.102367 2.688416 3.698767

$gdnss

gdnss Modality Reader

1 <NA> 1 1

$nLesDistr

[,1] [,2]

[1,] 1 69

[2,] 2 20

[3,] 3 11

$CBMAUC

[,1]

[1,] 0.9020396

$CBMmu

[,1]

[1,] 3.063572

$CBMalpha

[,1]

[1,] 0.8291959

$CBMcutoffs

NULL

$i

[1] 1

$j

[1] 1

The object retSmRoc is list of 16 elements (for the curious, count the instances of $ in the above code snippet; or type str(retSmRoc[[s]]) at the Console prompt and view the last seven items of the ouput).

...

...

..$ nLesDistr : num [1:3, 1:2] 1 2 3 69 20 11

..$ CBMAUC : num [1, 1] 0.879

..$ CBMmu : num [1, 1] 1.65

..$ CBMalpha : num [1, 1] 1

..$ CBMcutoffs: NULL

..$ i : int 5

..$ j : int 4

The first element is huge as it contains plotting data for 8000 predicted ROC data points (probably overkill, but it was done to ensure the plot looks continuous). It can be ignored for now. The list member named $nLesDistr gives the histogram of the lesion distribution. The member named $CBMAUC gives the CBM-AUC, which is the constraint. The member named $CBMalpha is the alpha parameter of the CBM distribution.

The list member named $j is the reader number and the statements in lines 44 – 52 will only execute if retSmRoc[[s]]$j is one of the four readers. Its current value is one (confirm by careful highlighting retSmRoc[[s]]$j and clicking on Run). Click on Next: the code block from lines 45 – 47 will only execute if retSmRoc[[s]]$i == i1, i.e., the modality index corresponds to 1, which happens to be true (confirm by highlighting etc.). Click on Next: the next three lines extract the values of for radiologist 1 in modality 1 and saves it in the positions of the length 4 array mu1, lambda1 and nu1 that evaluate to TRUE; to understand this jargon, on line 45 highlight retSmRoc[[s]]$j == selectJ and click on Run: one should see: TRUE FALSE FALSE FALSE (emphasis added). So the result of the right hand side, which is  = 3.675971, is saved to the *first* position of the mu1 array. The next two lines save the corresponding values (36.55742 and 0.4157737) to the first positions in the lambda1 and nu1 arrays. The three value correspond to *i* = 1 and *j* = 1. If one keep clicking on Next, the code pointer will return to line 45, but this time s = 2, corresponding to *i* = 1 and *j* = 2. This continues until all readers in the first modality have been exhausted, and the arrays mu1, lambda1 and nu1 are fully populated. On the next iterations of the for-loop retSmRoc[[s]]$i equals 2, which cause parameter results for the second modality to be extracted in lines 49 – 51 and saved to the correct positions in the mu2, lambda2 and nu2 arrays.

Exit debug mode (red-square button), remove any existing break point by clicking on it, insert a break point at line 62 and click Source. The RSM parameters are contained in mu, lambda and nu, which can be displayed by executing the command lines (the ones preceded with ">") in code snippet Online Appendix 19.C.3. Each parameter has two rows, corresponding to modalities 1 and 2 and 4 columns, corresponding to the four radiologists. Since some of the parameters can vary over large ranges, it is more prudent to take *medians* as representative of the 8 datasets, last three command lines in Online Appendix 19.C.3.

### Online Appendix 19.C.3: Code snippet

Browse[2]> mu

[,1] [,2] [,3] [,4]

mu1 3.675971 5.437591 2.617286 0.3358064

mu2 4.129106 2.959226 4.428728 1.2263501

Browse[2]> lambda

[,1] [,2] [,3] [,4]

lambda1 36.55742 2.098421 25.95178 0.04292726

lambda2 40.97380 1.342264 43.35359 0.64008142

Browse[2]> nu

[,1] [,2] [,3] [,4]

nu1 0.4157737 0.1324396 0.5027167 5.344361

nu2 0.2543416 0.3507453 0.1824401 1.892815

Browse[2]> muMed

[1] 3.317599

Browse[2]> lambdaMed

[1] 14.0251

Browse[2]> nuMed

[1] 0.3832595

Move the cursor to line 69 by clicking on Next many times or using the "*get me out of this loop*" button. Lines 65 – 66 calculate the lesion distribution vector nLesDistr and the lesion weights vector lesionWeights (this example assumes equal weighting).

### Online Appendix 19.C.4: Code snippet

Browse[2]> nLesDistr

[,1] [,2]

[1,] 1 69

[2,] 2 20

[3,] 3 11

Browse[2]> lesionWeights

[,1] [,2] [,3]

[1,] 1.0000000 -Inf -Inf

[2,] 0.5000000 0.5000000 -Inf

[3,] 0.3333333 0.3333333 0.3333333

This example tells us that there are 69 cases with 1 lesion, 20 with 2 lesions and 11 with 3 lesions. The corresponding weights are 1, 1/2 and 1/3, respectively. The –Inf's indicate lesions not present.

Keep clicking on Next to get past lines 69 – 71. These compute ROC-AUC and wAFROC-AUC. Use code snippet commands in Online Appendix 19.C.5 to print out the values (highlight and click Run).

### Online Appendix 19.C.5: Code snippet

Browse[2]> aucRocNH

[1] 0.8768796

Browse[2]> aucAfrocNH

[1] 0.6983678

Notice that the wAFROC FOM is smaller than the ROC FOM. This is usually the case unless the lesions are easily visible. Keep clicking on Next. Lines 75 – 83 increases muMed from 0.01 to 0.2 in steps of 0.01 (line 75). For each new value it calculates the ROC-AUC effect-size, esRoc, and the wAFROC-AUC effect-size, eswAfroc and saves the values in arrays dimensioned at line 76. An Excel file PowerComparison.xlsx in the online material showing that eswAfroc is a nearly linear function of esRoc (straight line fit constrained to run through the origin, slope = 2.023665, R2 = 0.9999844, contained in summary(a)$r.squared). In other words the wAFROC effect-size is about twice the ROC effect-size. Instead of going through Excel, line 86 calculates the regression (the ‑1 ensures the fit goes through the origin, as it must, because for zero , both FOMs must be at their null hypotheses values). Line 87 creates an effectSizeROC array that runs from 0.01 to 0.1 in steps of 0.01. Line 88 multiplies it by a factor 2.024, the value of a$coefficients[1], to create the corresponding effectSizeswAFROC array.

Line 90 sets the number of readers in the *pivotal* or test study as 5 and the corresponding number of cases to 100 (which is actually half that in the pilot study). Lines 91 – 92 perform DBMH analysis twice, once using fom = "HrAuc" and once using fom = "wAFROC", and extracts the two sets of pseudovalue variance components from the returned objects.

Exit debug mode, clear any existing break points and insert a new break point at line 100 and click Source. Lines 97 – 99 extract the pseudovalue treatment-reader (line 97), treatment-case (line 98) and error (line 99) variance components. Line 100 calculates power for the ROC method. Lines 103 – 106 repeat these steps for wAFROC analysis (fom = "wAFROC"). The function SSPowerGivenJK() computes statistical power for specified values of treatment-reader, treatment-case and error pseudovalue variance components and effect-size. Note the consistent usage of inferred ROC-AUC derived quantities in lines 97 – 101 and the consistent usage of wAFROC-AUC derived quantities in lines 103 – 107. All three variance components and effect-sizes are affected by the choice of FOM.

Remove all break points and click on Source. The results in a more readable format are in the Excel file PowerComparison.xlsx in the online material. The observed wAFROC effect-size was -0.00686, see red font near bottom of Online Appendix 19.C.8, which is too small to reach 80% power, even with the number of cases inflated to 200 and the number of readers reduced to 4 (the actual conditions of the study, Online Appendix 19.C.7). It is not surprising that the study did not find a significant difference between these two modalities.

### Online Appendix 19.C.6: Code output

JTest = 4 KTest = 200

ROC effect-size = , 0.01 wAFROC effect-size = , 0.02023665 , Statistical power ROC, wAFROC: 0.06936532 , 0.4445016

ROC effect-size = , 0.02 wAFROC effect-size = , 0.04047331 , Statistical power ROC, wAFROC: 0.1288161 , 0.9535477

ROC effect-size = , 0.03 wAFROC effect-size = , 0.06070996 , Statistical power ROC, wAFROC: 0.2291864 , 0.9997675

ROC effect-size = , 0.04 wAFROC effect-size = , 0.08094661 , Statistical power ROC, wAFROC: 0.3642924 , 0.9999999

ROC effect-size = , 0.05 wAFROC effect-size = , 0.1011833 , Statistical power ROC, wAFROC: 0.517793 , 1

ROC effect-size = , 0.06 wAFROC effect-size = , 0.1214199 , Statistical power ROC, wAFROC: 0.6672359 , 1

ROC effect-size = , 0.07 wAFROC effect-size = , 0.1416566 , Statistical power ROC, wAFROC: 0.7927595 , 1

ROC effect-size = , 0.08 wAFROC effect-size = , 0.1618932 , Statistical power ROC, wAFROC: 0.8841596 , 1

ROC effect-size = , 0.09 wAFROC effect-size = , 0.1821299 , Statistical power ROC, wAFROC: 0.9420896 , 1

ROC effect-size = , 0.1 wAFROC effect-size = , 0.2023665 , Statistical power ROC, wAFROC: 0.9741633 , 1

To put the previous values in perspective the following lists the results of DBMH analysis of the extracted 2-modality dataset. The important result is the small observed effect-size, in red font. The choice of 0.03 for the postulated effect size is not unreasonably large as the lower limit of the 95% CI is -0.028.

### Online Appendix 19.C.7: Code output

> ret <- DBMHAnalysis(frocData, fom = "wAFROC", option = "RRRC")

> ret

$fomArray

Rdr - 1 Rdr - 3 Rdr - 4 Rdr - 5

Trt - 1 0.7792667 0.7248917 0.7036250 0.8050917

Trt - 2 0.7870000 0.7269000 0.7226167 0.8037833

$anovaY

Source SS DF MS

1 T 0.01880327 1 0.018803266

2 R 2.33024767 3 0.776749224

3 C 130.13175366 199 0.653928410

4 TR 0.02382003 3 0.007940011

5 TC 27.33988244 199 0.137386344

6 RC 73.93443040 597 0.123843267

7 TRC 57.74518265 597 0.096725599

8 Total 291.52412012 1599 NA

$anovaYi

Source DF 1 2

1 R 3 0.4420838 0.3426054

2 C 199 0.3754000 0.4159148

3 RC 597 0.1088409 0.1117280

$varComp

varComp

Var(R) 0.0018542289

Var(C) 0.0611780498

Var(T\*R) -0.0004439279

Var(T\*C) 0.0101651862

Var(R\*C) 0.0135588340

Var(Error) 0.0967255991

$fRRRC

[1] 0.3868925

$ddfRRRC

[1] 112.3997

$pRRRC

[1] 0.5351971

$ciDiffTrtRRRC

Treatment Estimate StdErr DF t Pr > t CI Lower CI Upper

1 1 - 2 -0.00685625 0.01102279 112.3997 -0.6220068 0.5351971 -0.02869565 0.01498315

$ciAvgRdrEachTrtRRRC

Treatment Area StdErr DF CI Lower CI Upper

1 1 0.7532187 0.02976245 7.708447 0.6841319 0.8223056

2 2 0.7600750 0.02843396 10.692084 0.6972717 0.8228783

This concludes the explanation of the sample size estimation procedure.

As an exercise, the reader should repeat the analysi for the Franken and Van Dyke datasets and compare them to the results obtained in Chapter "Sample Size".

# References

1. Metz CE, Pan X. Proper Binormal ROC Curves: Theory and Maximum-Likelihood Estimation. *J Math Psychol.* 1999;43(1):1-33.

2. Zanca F, Jacobs J, Van Ongeval C, et al. Evaluation of clinical image processing algorithms used in digital mammography. *Medical Physics.* 2009;36(3):765-775.

3. Hillis SL, Obuchowski NA, Berbaum KS. Power Estimation for Multireader ROC Methods: An Updated and Unified Approach. *Academic Radiology.* 2011;18(2):129-142.

# Bookmarks

Chapter "Preliminaries"

Chapter "Binary Paradigm"

Chapter "Modeling Binary Paradigm"

Chapter "Ratings Paradigm"

Chapter "Empirical AUC"

Chapter "Binormal Model"

Chapter "Sources of variability in AUC"

Chapter "Hypothesis Testing"

Chapter "DBMH Analysis"

Chapter "ORH Analysis"

Chapter "Sample Size"

Chapter "FROC Paradigm"

Chapter "Empirical plots from FROC data"

Chapter "Meanings of FROC FOMs"

Chapter "Visual Search Paradigms"

Chapter "Radiological Search Model (RSM)"

Chapter "Predictions of the RSM"

Chapter Analyzing FROC Data

Chapter "RSM Fitting"

Chapter "Other proper ROC models"

Chapter "Bivariate binormal model"

Chapter "CAD Evaluation"

Chapter "Validating CAD Analysis"

Fig. 1.3.2

Appendix 1.A

Figure 1.A.4

Figure 1.A.5.1

Figure 1.A.5.2

Figure 1.A.5.1.2

2.9.2

Table 2.2.1 Truth Table

Table 2.2.2

Table 2.5

Appendix 2.A

Fig. 3.6

Fig. 3.6.1

Fig. 3.6.2

Fig. 3.C.1

Fig. 3.9

Fig. 3.9.5

Fig. 3.11.1 Beam

Fig. 3.A.2

Fig. 3.C.1

Fig. 3.G.2

Fig. 3.G.3

Fig. 3.C.4

Fig. 3.E.1

Fig. 3.G.2

Fig. 3.G.3

Fig. 4.G.4

Fig. 3.H.2.1

Fig. 3.H.2.2

Fig. 3.I.2

3.10

Table 3.7.6

Table 3.11.1 Beam

Appendix 3.A: R code demonstration of concepts

Appendix 3.B: Calculating a confidence interval

Appendix 3.C: Introduction to R/RStudio, part II

Appendix 3.D: Plotting in R

Appendix 3.E: Getting help in R – Part I

Appendix 3.F: Getting help in R – Part II

Appendix 3.G: What to do if a package is missing

Appendix 3.H: Showing shaded distributions in R

Appendix 3.I: Numerical integration in R

Chapter 4

Fig. 4.3.1

Fig 4.6.1

Fig. 4.7.1

Table 4.2.1

Table 4.3.1

Table 4.3.2

Table 4.3.3

Table 4.10.1

Appendix 4.A

Chapter 5

Table 5.2.1

Table 5.3.1

Fig. 5.3.1

Fig. 5.4.1

Fig. 5.6.1

5.6

Appendix 5.A

Fig. 5.A.2.1

Fig. 5.A.2.2

Fig. 5.A.2.3

Fig. 5.A.2.4

Fig. 5.A.2.5

Appendix 6.A

Appendix 6.B

Appendix 6.C

Appendix 6.D

Appendix 6.E

Appendix 6.F

6.4.2

Fig. 6.1

Fig. 6.2.6

Fig. 6.2.1

Fig. 6.2.3

Fig. 6.2.8

Fig. 6.3.2

Table 6.1

Table 6.2

Table 6.4.4

Fig. 6.A.2.1

Fig. 6.A.2.2

Fig. 6.A.3

Fig. 6.A.4

7.3.2

Table 7.4.1

Table 7.4.2

Table 7.6

Fig. 7.9

Appendix 7.A

Appendix 7.B

Appendix 7.C

Appendix 7.D

Appendix 7.E

Table 8.5.1

Fig. 8.5.1

9

Table 9.7

9.10.2.2

Fig. 9.7.3

Table 9.8

Table 9.9

Table 9.10.2.1

Table 9.10.2.2

Table 9.10.2.3

Table 9.10.2.4

Table 9.11.1

Fig. 9.10.1.1

Fig. 9.10.1.2

Fig. 9.11.1

Fig. 9.11.2

9.13

Table 9.13.1

Table 9.13.3

Fig. 9.A.2

Fig. 9.A.3.1

Table 9.E.1

Table 9.H.1

Table 09.C.1

Online Appendix 9.A

Online Appendix 9.B

Online Appendix 9.C

Online Appendix 9.D

Online Appendix 9.E

Online Appendix 9.F

Online Appendix 9.G

10.2.5.1

10.2 1RMultipleT

10.5.1

10.7 1TMultipleR

Fig. 10.5.1

10.2.4 Meaning of Cov matrix

Appendix 10.A

Appendix 10.B

Appendix 10.C

Appendix 10.D

Appendix 10.E

Appendix 10.F

Table 11.2.1 2 types of errors

Fig. 11.3.1

11.7.1

11.7.2

Fig. 11.11.1

Fig. 11.11.2

Fig. 11.11.1

Fig. 11.11.2

11.11

Table 11.7.2.1

Table 11.8.1

Online Appendix 11.A

Online Appendix 11.B

Online Appendix 11.C

Online Appendix 11.D

Chapter 12

Fig. 12.2.1

Fig. 12.5.1

Fig. 12.5.2

Appendix 12.B

Fig. 12.A.1

Fig. 12.B.1

Fig. 12.B.2

Table 12.4.1

12.A.1

12.6

Chapter 13

Fig. 13.4.2

Fig. 13.10.1

Fig. 13.10.4

Fig. 13.13.1

Fig. 13.14.1

Fig. 13.15.1

Fig. 13.16.1

Fig. 13.16.2

Fig. 13.16.3

13.5

13.6

13.7

13.4.1

13.8.1

13.10.1

13.10.3.1

Fig. 13.10.4.1

13.10.5.1

13.10.5.2

13.11

13.13.1

13.12.1

13.12.2

13.12.3

13.14.1

13.15.1

Appendix 13.A.1

Table 13.2.1

Table 13.12.3

Table 13.14

Table 13.15

Table 13.16

Table 14.C.1

Fig. 14.4.1

Fig. 14.5.3.1

Fig. 14.5.3.2

Fig. 14.5.3

Fig. 14.5.4.1

Fig. 14.5.4.2

14.A

Table 4.5.2.1

Table 14.5.2.2

14.5.1

14.5.2

14.2

Appendix 14.A

Appendix 14.B

Appendix 14.C

Appendix 14.D

Fig. 15.2.1

Fig. 15.2.2

Fig. 15.4.2

Fig. 15.6.1

Fig. 15.7.1

Fig. 15.6.2

Fig. 15.A.1

Appendix 15A

Appendix 15B

Appendix 15C

Appendix 15D

Table 16.2.3.1

16.4

Appendix 16.A.1

Appendix 16.B.1

Appendix 16.C.1

17.3

Fig. 17.5.2.1

Fig. 17.5.2.2

Fig. 17.6.1

Fig. 17.8.1

Fig. 17.7.1

Fig. 17.7.4.1

Fig. 17.10.1

Fig. 17.10.2

Fig. 17.11.2

Fig. 17.11.3.1.1

Fig. 17.11.3.1.2

Table. 17.11.3.1.2

Table 17.11.3.2

17.5.1

17.5.2

17.6.2

17.7.2

17.8 Search performance

17.9 Class. Performance

17.10 IsFrocGood

17.10.1 Clinical relevance

17.11.2.1 Eng Code

17.11

demo

17.11.4

Table 17.4.6.2

Table 17.11.2

Online Appendix 17.A

Online Appendix 17.B

Online Appendix 17.C

Online Appendix 17.D.1

Online Appendix 17.D.2

Online Appendix 17.E

Online Appendix 17.F

Online Appendix 17.F.1

Online Appendix 17.F.2

Online Appendix 17.F.3

Online Appendix 17.G

18

Fig. 18.3.1

Fig. 18.3.2

Table 18.2.1

Table 18.2.2

Table 18.2.3

19

19.8

19.5

Fig. 19.3.1

Fig. 19.3.2

Table 19.5.1

Table 19.5.2

Table 19.5.3

Table 19.5.4

Table 19.5.5

Fig. 19.7.1

Fig. 19.7.2

Table 19.7.1

Table 19.7.2

Table 19.8.2.1

Fig. 19.8.2.1

Fig. 19.5.1

Fig. 19.5.2

Fig. 19.5.3

Fig. 19.5.4

Fig. 19.5.5

Online Appendix 19A

Online Appendix 19.A.1

Online Appendix 19.A.2

Online Appendix 19.A.3

Online Appendix 19.A.4

Online Appendix 19.B

Online Appendix 19.C

Online Appendix 19.C.1

Online Appendix 19.C.2

Online Appendix 19.C.3

Online Appendix 19.C.4

Online Appendix 19.C.5

Online Appendix 19.C.6

Online Appendix 19.C.7

Online Appendix 19.C.8

20.7.2.1

Fig. 20.4.1

Fig. 20.4.2

Fig. 20.5.3

Fig. 20.6.1

Fig. 20.7.1

Fig. 20.8.1

Fig. 20.8.2

Fig. 20.8.3

Fig. 20.9.1

Fig. 20.9.2

Table 20.5.1

Table 20.7.1.1

Online Appendix 20.A

Online Appendix 20.B

Online Appendix 20.B.2

Online Appendix 20.B.4

Online Appendix 20.D

Fig. 21.4.1

Fig. 21.6.1

Table 21.6.1

Fig. 21.7.1

Fig. 21.7.2

Fig. 21.B.1

Fig. 21.D.1

Fig. 21.D.2

Fig. 21.D.3

Fig. 21.D.4

Fig. 21.D.5

Fig. 21.D.6

Online Appendix 21.A

Online Appendix 21.B

Online Appendix 21.C

Online Appendix 21.D

Online Appendix 21.E

22

Table 22.2.1

Online Appendix 22.A

Online Appendix 22.B

22.2.1.1

22.2.1.2

22.2.1.3

22.2.1.4

Fig. 22.2.1

Fig. 22.A.1

Table 22.3.1

22.4

22.5

Table 22.5.1

23.A.1

Online Appendix Datasets

# Cross-references

Chapter "Preliminaries"

Chapter "Binary Paradigm"

Chapter "Modeling Binary Paradigm"

Chapter "Ratings Paradigm"

Chapter "Empirical AUC"

Chapter "Binormal Model"

Chapter "Sources of variability in AUC"

Chapter "Hypothesis Testing"

Chapter "DBMH Analysis"

Chapter "ORH Analysis"

Chapter "Sample Size"

Chapter "FROC Paradigm"

Chapter "Empirical plots from FROC data"

Chapter "Meanings of FROC FOMs"

Chapter "Visual Search Paradigms"

Chapter "Radiological Search Model (RSM)"

Chapter "Predictions of the RSM"

Chapter Analyzing FROC Data

Chapter "RSM Fitting"

Chapter "Other proper ROC models"

Chapter "Bivariate binormal model"

Chapter "CAD Evaluation"

Chapter "Validating CAD Analysis"

Appendix 1.A

Fig. 1.3.2

Figure 1.A.4

Figure 1.A.5.1

Figure 1.A.5.2

Figure 1.A.5.2

Figure 1.A.5.1.2

2.9.2 PPV NPV

Table 2.2.1

Table 2.2.2

Table 2.5

Appendix 2.A

Fig. 3.6

Fig. 3.6.1

Fig. 3.6.2

Fig. 3.A.2

Fig. 3.H.2.1

Fig. 3.H.2.2

Fig. 3.E.1

Fig. 3.9

Fig. 3.9.5

Fig. 3.11.1 beam et al study

Fig. 3.G.3

Fig. 3.C.4

Fig. 3.A.2

Fig. 3.C.1

Fig. 3.E.1

Fig. 3.G.2

Fig. 3.G.3

Fig. 4.G.4

Fig. 3.H.2.1

Fig. 3.I.2

3.10

Table 3.7.6

Table 3.11.1 beam et al study

Appendix 3.A

Appendix 3.B

Appendix 3.C

Appendix 3.D

Appendix 3.E Getting help in R-I

Appendix 3.F Getting help in R-II

Appendix 3.G

Appendix 3.H

Appendix 3.I

Fig. 4.3.1 equal var vs uneq var

Fig 4.6.1

Fig. 4.7.1  
Table 4.2.1 Roc counts table

Table 4.3.1

Table 4.3.2

Table 4.3.3

Table 4.10.1

Appendix 4.A

Table 5.2.1

Table 5.3.1

5.6

Appendix 5.A

Fig. 5.A.2.1 Debug buttons

Fig. 5.A.2.2 Debug buttons

Fig. 5.A.2.3 Debug buttons

Fig. 5.A.2.4 Debug buttons

Fig. 5.A.2.5 Debug buttons

Appendix 6.A

Appendix 6.B Eng Java

Appendix 6.C MLE

Appendix 6.D Goodness of fit

Appendix 6.E

Appendix 6.F Partial/full Az

6.4.2

Fig. 6.2.6

Fig. 6.2.8

Fig. 6.2.1

Fig. 6.2.3

Fig. 6.3.2

Table 6.2

Table 6.4.4

Fig. 6.A.2.1

Fig. 6.A.2.2

Fig. 6.A.3

Fig. 6.A.4

7.3.2

7.3.2

Table 7.4.1

Table 7.4.2

Table 7.6

Fig. 7.9

Appendix 7.A

Appendix 7.B

Appendix 7.C

Appendix 7.D

Appendix 7.E

Table 8.5.1

Table 8.5.1

Fig. 8.5.1

Table 9.7

9.10.2.2

Fig. 9.7.3

Table 9.8

Table 9.9

Table 9.10.2.2

Table 9.10.2.3

Table 9.10.2.4

Table 9.11.1

Fig. 9.10.1.1

Fig. 9.10.1.2

Fig. 9.11.1

Fig. 9.11.2

Table 9.10.2.1

9.13: Meaning of pseudovalues

Table 9.13.1

Table 9.13.3

Fig. 9.A.2

Fig. 9.A.3.1

Table 09.C.1

Table 9.E.1

Table 9.H.1

Online Appendix 19A

Online Appendix 9.B

Online Appendix 9.C

Online Appendix 9.D

Online Appendix 9.E

Online Appendix 9.F

Online Appendix 9.G

10.2.5.1

10.2 1RIT

10.5.1

10.7 1TMultipleR

Fig. 10.5.1

10.5.1

10.7 1TMultipleR

10.2.4 Meaning of Cov matrix

Appendix 10.A

Appendix 10.B

Appendix 10.C

Appendix 10.D

Appendix 10.E

Appendix 10.F

Table 11.2.1 2 types of errors

Fig. 11.3.1

11.7.1

11.7.2

Fig. 11.11.1

Fig. 11.11.2

Fig. 11.11.1

Fig. 11.11.2

11.11

Table 11.7.2.1

Table 11.8.1

Online Appendix 11.A

Online Appendix 11.B

Online Appendix 11.C

Online Appendix 11.D

Appendix 12.B

Fig. 12.B.2

12.6

Table 9.H.1

Table 13.2.1 FROC notation

Table 13.12.3

Table 13.14

Table 13.15

Table 13.16

Fig. 13.4.2

Fig. 13.10.1

Fig. 13.10.4

Fig. 13.13.1

Fig. 13.14.1

Fig. 13.15.1

Fig. 13.16.1

Fig. 13.16.2

Fig. 13.16.3

13.6

13.7

13.4.1

13.8.1

13.10.1

13.10.3.1

Fig. 13.10.4.1

13.10.5.1

13.10.5.2

13.11

Appendix 13.A.1

13.12.1

13.12.2

13.12.3

13.13.1

13.14.1

13.15.1

Table 14.C.1

Fig. 14.4.1

Fig. 14.5.3.1

Fig. 14.5.3.2

Fig. 14.5.4.1

Fig. 14.5.4.2

14.5.1

14.5.2

14.2

Appendix 14.A

Appendix 14.B

Appendix 14.C

Appendix 14.D

Table 14.5.2.1

Table 14.5.2.2

Fig. 15.2.1

Fig. 15.2.2

Fig. 15.4.2

Fig. 15.6.1

Fig. 15.6.2

Fig. 15.7.1

Fig. 15.A.1

Appendix 15A

Appendix 15B

Appendix 15C

Appendix 15D

Table 16.2.3.1

16.4

Appendix 16.A.1

Appendix 16.B.1

Appendix 16.C.1

17.3

Fig. 17.5.2.1

Fig. 17.5.2.2

17.5.1

17.5.2

Fig. 17.6.1

Fig. 17.7.1

Fig. 17.8.1

Fig. 17.7.4.1

Fig. 17.10.1

Fig. 17.10.2

Fig. 17.11.2

Fig. 17.11.3.1.1

Fig. 17.11.3.1.2

Table. 17.11.3.1.2

Table 17.11.3.2

17.5.1

17.5.2

17.6.2

17.7.2

17.8

17.9

17.10

17.10.1

17.11

17.11.2.1

17.11.4

Table 17.4.6.2

Table 17.11.2

Online Appendix 17.A

Online Appendix 17.B

Online Appendix 17.C

Online Appendix 17.D.1

Online Appendix 17.D.2

Online Appendix 17.F

Online Appendix 17.F.1

Online Appendix 17.F.2

Online Appendix 17.F.3

Online Appendix 17.F.3

Online Appendix 17.G

18

Fig. 18.3.1

Fig. 18.3.2

Table 18.2.1

Table 18.2.2

Table 18.2.3

19

19.8 Sample size

19.5

Table 19.5.1

Table 19.5.2

Table 19.5.3

Table 19.5.4

Table 19.5.5

Fig. 19.7.1

Fig. 19.7.2

Table 19.8.2.1

Table 19.7.1

Table 19.7.2

Fig. 19.8.2.1

Fig. 19.3.1

Fig. 19.3.2

Fig. 19.5.1

Fig. 19.5.2

Fig. 19.5.3

Fig. 19.5.4

Fig. 19.5.5

Online Appendix 19A

Online Appendix 19.A.1

Online Appendix 19.A.2

Online Appendix 19.A.3

Online Appendix 19.A.4

Online Appendix 19.C

Online Appendix 19.C.1

Online Appendix 19.C.2

Online Appendix 19.C.3

Online Appendix 19.C.4

Online Appendix 19.C.5

Online Appendix 19.C.6

Online Appendix 19.C.7

Online Appendix 19.C.8

20.7.2.1

Fig. 20.4.1

Fig. 20.4.2

Fig. 20.5.3

Fig. 20.6.1

Fig. 20.7.1

Fig. 20.8.1

Fig. 20.8.2

Fig. 20.8.3

Fig. 20.9.1

Fig. 20.9.2

Table 20.5.1

Table 20.7.1.1

Online Appendix 20.A

Online Appendix 20.B

Online Appendix 20.B.2

Online Appendix 20.B.4

Online Appendix 20.D

Fig. 21.4.1

Fig. 21.6.1

Table 21.6.1

Fig. 21.7.1

Fig. 21.7.2

Fig. 21.B.1

Fig. 21.D.1

Fig. 21.D.2

Fig. 21.D.3

Fig. 21.D.4

Fig. 21.D.5

Fig. 21.D.6

Online Appendix 21.A

Online Appendix 21.B

Online Appendix 21.C

Online Appendix 21.D

Online Appendix 21.E

Table 22.2.1

Online Appendix 22.A

Online Appendix 22.B

22.2.1.1

22.2.1.2

22.2.1.3

22.2.1.4

Fig. 22.2.1

Fig. 22.A.1

Table 22.3.1

22.4

22.5

Table 22.5.1

23.A.1

Online Appendix Datasets

# Book last updated:

3/17/17 5:14:20 PM