**Appendix: Sketch of R code**

The R code that was written for fitting the LR lasso DIF method and extracting all useful information is briefly sketched. Several functions were written to split the computational effort and return specific output. Recall that the R package *glmnet* (Friedman, Hastie, & Tibshirani, 2010) must be installed and loaded prior to performing LR lasso DIF analyses. Moreover, the data set must be either a matrix or data frame with five columns and specific column names (but the ordering of the columns is irrelevant). Each row of the data matrix corresponds to one particular item response. The five columns are: *Y* for the item response (coded as 0 for incorrect and 1 for correct); *PERS* for person indicator (integer values from 1 to ); *GROUP* for group membership (two distinct values, either numeric or character); *ITEM* for item number (from 1 to *J*, character names are also allowed); and *SCORE* for person test score (integer value, typically the raw score).

First, the function *lassoDIF* fits the lasso penalized logistic regression for a given data set (*data*) and returns all basic output from the *glmnet* function.

|  |
| --- |
| lassoDIF <- function(data){  if (!is.data.frame(data)) data <- as.data.frame(data)  y <- factor(data$Y)  J <- length(unique(data$ITEM))  x <- model.matrix(data$Y ~ -1 + factor(data$ITEM) + data$SCORE + factor(data$ITEM):factor(data$GROUP))  pen <- rep(0, 2\*J+1)  pen[(length(pen)-J+1):length(pen)] <- 1  prov <- glmnet(x, y, family = "binomial", alpha = 1, penalty.factor = pen, intercept = FALSE)  res <- glmnet(x, y, family = "binomial", alpha = 1, penalty.factor = pen,  lambda = c(prov$lambda,0), intercept = FALSE)  return(res)} |

The *lassoDIF.coef* extracts the estimated DIF parameters in a matrix format with one row per item and one column per value. Input to be provided consists of the output (*out*) of the *lassoDIF* function and the number of values (*nr.lambda*) to be considered (with a default value of 1000, the function internally defines a sequence from 0 to the maximal selected value). The output is a list with all values and the corresponding matrix of DIF parameters.

|  |
| --- |
| lassoDIF.coef <- function(out, nr.lambda = 1000){  J <- (nrow(out$beta)-1)/2  nr1 <- nrow(coef(out, s = 0)) – J + 1  nr2 <- nrow(coef(out, s = 0))  s <- seq(from = 0, to = max(out$lambda), length = nr.lambda)  mat <- coef(out, s = s)[nr1:nr2, ]  mat.names <- "Item1"  for (i in 2:J) mat.names <- c(mat.names, paste("Item", i, sep = ""))  rownames(mat) <- mat.names  return(list(lambda = s, pars = as.matrix(mat)))} |

The function lassoDIF.ABWIC determines the optimal value with either AIC, BIC or WIC criterion, and returns in addition the items that are flagged as DIF for this value (together with additional output such as DIF parameters of all items for that optimal value). Input to be provided consists of the output (*out*) of the *lassoDIF* function, the type (type*) of criterion to be used (either* “AIC”*,* “BIC” *or* “WIC”) and the total sample size (*N*), the latter not being used for AIC criterion.

|  |
| --- |
| lassoDIF.ABWIC <- function(out, type="AIC", N=NULL){  J <- (nrow(out$beta)-1)/2  if (type == "AIC" | type == "BIC"){  CRIT <- switch(type, AIC = deviance(out)+2\*out$df, BIC = deviance(out)+log(J\*N)\*out$df)  l.opt <- out$lambda[CRIT == min(CRIT)]  nr.opt <- (1:length(out$lambda))[abs(out$lambda-l.opt) == min(abs(out$lambda-l.opt))]  pr <- out$beta[, nr.opt]  }  if (type == "WIC"){  CRIT <- NULL  ppAIC <- deviance(out) + 2\*out$df  ppBIC <- deviance(out) + log(J\*N)\*out$df  s <- seq(from = 0, to = 1, length = nr.w)  l.seq <- NULL  for (i in 1:length(s)){  f <- s[i]\*ppAIC + (1-s[i])\*ppBIC  l.seq[i] <- out$lambda[f == min(f)]  }  l.opt <- median(unique(l.seq))  nr1 <- nrow(coef(out, s = 0)) – J + 1  nr2 <- nrow(coef(out, s = 0))  pr <- coef(out, s = l.opt)[nr1:nr2, 1]  }  IND <- (length(pr) – J + 1):(length(pr))  RES <- NULL  if (max(abs(pr[IND])) > 0) RES <- (1:J)[abs(pr[IND]) > 0]  mat <- cbind(pr[IND])  mat.names <- "Item1"  for (i in 2:J) mat.names <- c(mat.names, paste("Item", i, sep = ""))  rownames(mat) <- mat.names  return(list(DIFitems = RES, DIFpars = mat, crit.value = CRIT, crit.type = type, lambda = out$lambda, opt.lambda = l.opt))} |

Finally, the *lassoDIF.CV* function determines the optimal value by cross-validation and returns in addition the items that are flagged as DIF for this value (together with the estimated DIF parameters of all items for that optimal value). Input to be provided consists of the output (*out*) of the *lassoDIF* function, the original data set (*data*) and the number of folds (*nfold*) for cross-validation, with a default value of 3.

|  |
| --- |
| lassoDIF.CV <- function(out, data, nfold = 3){  if (!is.data.frame(data)) data <- as.data.frame(data)  y <- factor(data$Y)  J <- length(unique(data$ITEM))  x <- model.matrix(data$Y~-1+factor(data$ITEM)+data$SCORE+factor(data$ITEM):factor(data$GROUP))  pen<-rep(0, 2\*J+1)  pen[(length(pen)-J+1):length(pen)] <- 1  prov <- cv.glmnet(x, y, family = "binomial", nfolds = nfold, alpha = 1, type.measure = "deviance", penalty.factor = pen)  l.opt <- max(prov$lambda[prov$cvm == min(prov$cvm)])  nr.opt <- (1:length(out$lambda))[abs(out$lambda-l.opt) == min(abs(out$lambda-l.opt))]  pr <- out$beta[, nr.opt]  IND <- (length(pr)-J+1):(length(pr))  RES <- NULL  if (max(abs(pr[IND])) > 0) RES <- (1:J)[abs(pr[IND]) > 0]  return(list(DIFitems = RES, DIFpars = pr, opt.lambda = l.opt))} |