BADA

Richa Singh

11/9/2018

## BADA

Barycentric discriminant analysis (BADA) is a robust version of discriminant analysis that is used to assign, to pre-defined groups (also called categories), observations described by multiple variables.The goal of BADA is to combine the measurements to create new variables (called components or discriminant variables) that best separate the categories. These discriminant variables are also used to assign the original observations or ???new??? observations to the a-priori defined categories.Barycentric discriminant analysis is a robust version of discriminant analysis that is used when multiple measurements describe a set of observations in which each observation be- longs to one category (i.e., group) from a set of a priori defined categories. BADA combines the original variables to create new variables that best separate the groups and that can also be used to optimally assign old or new observations to these categories. The quality of the performance is evaluated by cross-validation techniques that estimate the performance of the classification model for new observations. BADA is a very versatile technique that can be declined in several different varieties that can handle, for example, qualitative data and data structured in blocks. This versatility make BADA particularly suited for the analysis of multi-modal and Big data.

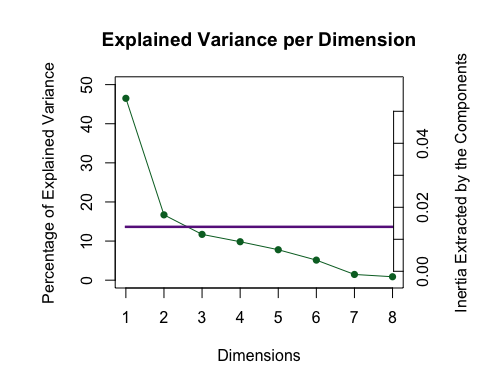
## Dataset

The dataset consists of 1233 observations and 32 variables describing the HR-IBM Employees. The variables: Quantitative Variables : Sub,Age,Monthly Income,Daily Rate,Hourly Rate, MonthlyRate, DistanceFromeHome,PercentSalaryHike, TotalWorkingYears,TrainingTimesLastYear,WorkLifeBalance,WorkLifeBalance,YearsAtCompany,YearsInCurrentRole,YearsSinceLastPromotion,YearsWithCurrManager Ordinal Variables : PerformanceRating, Education, JobInvolvement,Joblevel,StockOptionLevel,EnvironmentSatisfaction, JobSatisfaction,RelationshipSatisfaction Qualitative variable : Attrition,BusinessTravel,EducationField,Gender,JobRole,MaritalStatus,OverTime

datae <- data[,8]  
resBADA <- tepBADA(DATA = data1,  
 scale = 'SS1', center = TRUE,  
 DESIGN = datae,  
 make\_design\_nominal = TRUE,  
 group.masses = NULL,  
 weights = NULL, graphs = FALSE)

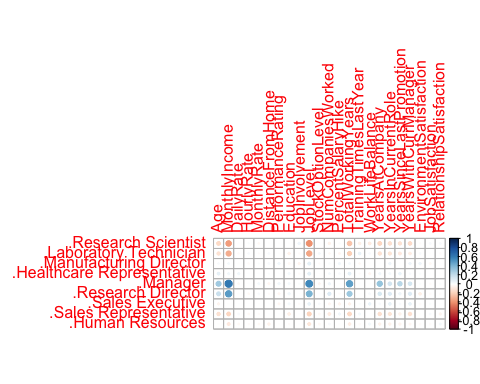
## ScreePlot

PlotScree(ev = resBADA$TExPosition.Data$eigs,  
 p.ev = NULL, max.ev = NULL, alpha = 0.05,  
 col.ns = "#006D2C", col.sig = "#54278F",  
 title = "Explained Variance per Dimension",plotKaiser = TRUE)



## HeatMap

color4Var <- prettyGraphs::prettyGraphsColorSelection(ncol(data1))  
data2 <- makeNominalData(as.matrix(data[,8]))  
corrplot::corrplot(cor(data2,data1))



Fk <- resBADA$TExPosition.Data$fi  
Fi <- resBADA$TExPosition.Data$fii  
Fj <- resBADA$TExPosition.Data$fj

## Factor Map J

col4Var <- prettyGraphsColorSelection(NCOL(data1))  
baseMap.j <- PTCA4CATA::createFactorMap(Fj,  
 col.points = col4Var,  
 alpha.points = .3,  
 col.labels = col4Var)  
# A graph for the J-set  
aggMap.j <- baseMap.j$zeMap\_background + # background layer  
 baseMap.j$zeMap\_dots + baseMap.j$zeMap\_text # dots & labels  
# We print this Map with the following code  
dev.new()  
print(aggMap.j)  
zeLines <- ggplot2::annotate("segment", x = c(0), y = c(0),  
 xend = Fj[,1],  
 yend = Fj[,2],  
 color = col4Var,  
 alpha = .5,  
 arrow = arrow(length = unit(.3, "cm") ) )  
# Create the map by adding background, labels, and arrows:  
aggMap.j.arrows <- baseMap.j$zeMap\_background +  
 zeLines + baseMap.j$zeMap\_text  
dev.new()  
print(aggMap.j.arrows)

## Factor Map I

baseMap.i <- PTCA4CATA::createFactorMap(Fi,  
 col.points = resBADA$Plotting.Data$fii.col,  
 alpha.points = .3)  
# Plain map with color for the I-set  
aggMap.i <- baseMap.i$zeMap\_background + baseMap.i$zeMap\_dots  
#---------------------------------------------------------------------  
# print this Map  
dev.new()  
print(aggMap.i)

col4data <- resBADA$Plotting.Data$fii.col  
col4Means <- unique(col4data)  
# create the map for the means  
MapGroup <- PTCA4CATA::createFactorMap(Fk,  
 axis1 = 1, axis2 = 2,  
 constraints = baseMap.i$constraints,  
 title = NULL,  
 col.points = col4Means,  
 display.points = TRUE,  
 pch = 19, cex = 5,  
 display.labels = TRUE,  
 col.labels = col4Means,  
 text.cex = 4,  
 font.face = "bold",  
 font.family = "sans",  
 col.axes = "darkorchid",  
 alpha.axes = 0.2,  
 width.axes = 1.1,  
 col.background = adjustcolor("lavender",  
 alpha.f = 0.2),  
 force = 1, segment.size = 0)  
# The map with observations and group means  
aggMap.i.withMeans <- aggMap.i+  
 MapGroup$zeMap\_dots + MapGroup$zeMap\_text  
#---------------------------------------------------------------------  
# plot it!  
dev.new()  
print(aggMap.i.withMeans)

## Create 75% Tolerance interval polygons

GraphTI.Hull.90 <- MakeToleranceIntervals(Fi,  
 as.factor(datae),  
 names.of.factors = c("Dim1","Dim2"),  
 col = unique(col4data),  
 line.size = .5, line.type = 3,  
 alpha.ellipse = .2,  
 alpha.line = .4,  
 p.level = .75, # 75% TI  
 type = 'hull' #  
 # use 'hull' for convex hull  
)  
#---------------------------------------------------------------------  
# Create the map  
aggMap.i.withHull <- aggMap.i +  
 GraphTI.Hull.90 + MapGroup$zeMap\_dots +  
 MapGroup$zeMap\_text + MapGroup$zeMap\_dots  
#---------------------------------------------------------------------  
# Plot it!  
dev.new()  
print(aggMap.i.withHull)

## Inferences

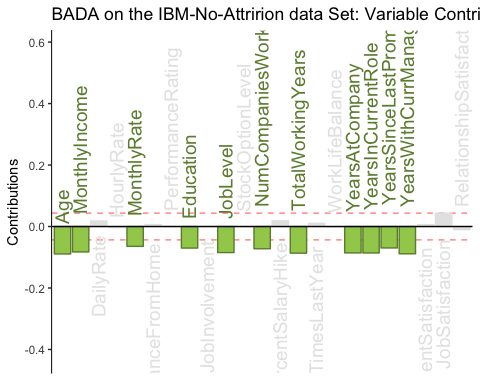
resBADA.inf <- tepBADA.inference.battery(DATA = data1,  
 scale = 'SS1', center = TRUE,  
 DESIGN = datae,  
 make\_design\_nominal = TRUE,  
 group.masses = NULL,  
 weights = NULL,  
 graphs = FALSE,  
 k = 2,  
 test.iters = 100,  
 critical.value = 2)

## [1] "It is estimated that your iterations will take 0.67 minutes."  
## [1] "R is not in interactive() mode. Resample-based tests will be conducted. Please take note of the progress bar."  
## ===========================================================================

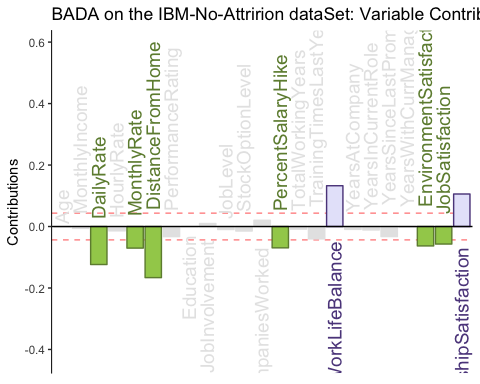
#---------------------------------------------------------------------  
# Confusion matrices  
# To be saved as table  
fixedCM <- resBADA.inf$Inference.Data$loo.data$fixed.confuse  
looedCM <- resBADA.inf$Inference.Data$loo.data$loo.confuse  
  
#---------------------------------------------------------------------  
# Create Confidence Interval Plots  
BootCube <- resBADA.inf$Inference.Data$boot.data$fi.boot.data$boots  
dimnames(BootCube)[[2]] <- c("Dimension 1","Dimension 2")  
# use function MakeCIEllipses from package PTCA4CATA  
GraphElli <- MakeCIEllipses(BootCube[,1:2,],  
 names.of.factors = c("Dimension 1","Dimension 2"),  
 col = col4Means,  
 p.level = .95  
)  
#---------------------------------------------------------------------  
# create the I-map with Observations, means and confidence intervals  
#  
aggMap.i.withCI <- aggMap.i + GraphElli + MapGroup$zeMap\_text  
#---------------------------------------------------------------------  
# plot it!  
dev.new()  
print(aggMap.i.withCI)

## Contribution

signed.ctrJ1 <- resBADA$TExPosition.Data$cj \* sign(resBADA$TExPosition.Data$fj)  
b003.ctrJ.s.11 <- PrettyBarPlot2(signed.ctrJ1[,1],  
 threshold = 1 / NROW(signed.ctrJ1),  
 font.size = 5,  
 # color4bar = gplots::col2hex(col4J.ibm), # we need hex code  
 main = 'BADA on the IBM-No-Attririon data Set: Variable Contributions (Signed)',  
 ylab = 'Contributions',  
 ylim = c(1.2\*min(signed.ctrJ1), 1.2\*max(signed.ctrJ1))  
)  
print(b003.ctrJ.s.11)

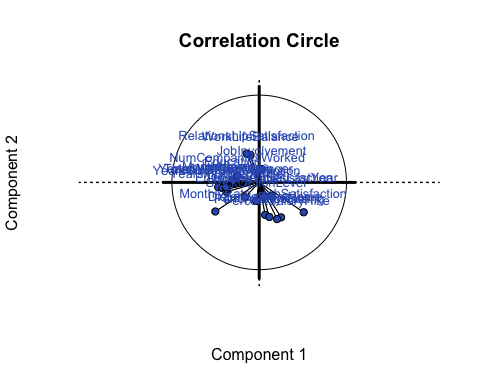


b004.ctrJ.s.21 <- PrettyBarPlot2(signed.ctrJ1[,2],  
 threshold = 1 / NROW(signed.ctrJ1),  
 font.size = 5,  
 # color4bar = gplots::col2hex(col4J.ibm), # we need hex code  
 main = 'BADA on the IBM-No-Attririon dataSet: Variable Contributions (Signed)',  
 ylab = 'Contributions',  
 ylim = c(1.2\*min(signed.ctrJ1), 1.2\*max(signed.ctrJ1))  
)  
print(b004.ctrJ.s.21)



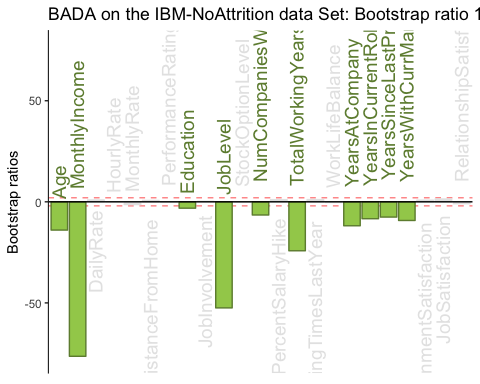
## Correlation Circle

correlationCircle <- correlationPlotter(data\_matrix = data1 , factor\_scores = resBADA$TExPosition.Data$lx , x\_axis = 1, y\_axis = 2, col = NULL ,pch = NULL, xlab = NULL, ylab = NULL, main = "Correlation Circle", asp = 1, dev.new = FALSE)

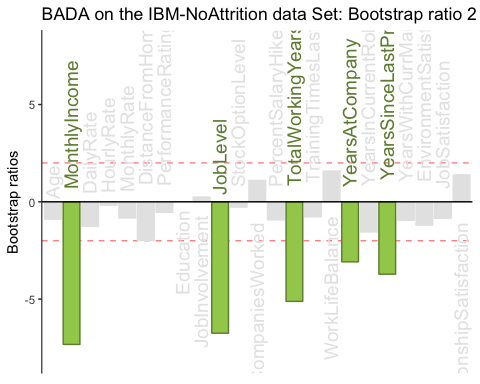


## Bootstrap Ratios

BR1 <- resBADA.inf$Inference.Data$boot.data$fj.boot.data$tests$boot.ratios  
laDim = 1  
ba001.BR11 <- PrettyBarPlot2(BR1[,laDim],  
 threshold = 2,  
 font.size = 5,  
 #color4bar = gplots::col2hex(col4J.ibm),  
 main = paste0( 'BADA on the IBM-NoAttrition data Set: Bootstrap ratio ',laDim),  
 ylab = 'Bootstrap ratios'  
 #ylim = c(1.2\*min(BR[,laDim]), 1.2\*max(BR[,laDim]))  
)  
print(ba001.BR11)



#  
laDim = 2  
ba002.BR21 <- PrettyBarPlot2(BR1[,laDim],  
 threshold = 2,  
 font.size = 5,  
 #color4bar = gplots::col2hex(col4J.ibm),  
 main = paste0(  
 'BADA on the IBM-NoAttrition data Set: Bootstrap ratio ',laDim),  
 ylab = 'Bootstrap ratios'  
)  
print(ba002.BR21)



## Random (LOO) confusion matrix

cm <- as.matrix(resBADA$TExPosition.Data$assign$confusion)  
correctly\_classified <- sum(diag(cm))  
Incorrectly\_classified <- sum(cm)-sum(diag(cm))  
print(sum(diag(cm))/sum(cm))

## [1] 0.1686942

## Summary