

Bayesian Network modelling for Cx Radiculopathy

Set this option to TRUE if you want to run the code

```
knitr::opts_chunk$set(eval = FALSE)
```

Load libraries

```
library(tidyverse)
library(bnlearn)
library(Rgraphviz)
library(doParallel)
library(corrplot)
```

Load data

```
rm(list = ls())
load("CxRadData_Final.RData")
```

Figure 1

```
descr = df.abs %>% # Group: 0 is control
  select (-Sex, -Age, -EQ5D, -csq_cop, -csq_adp) %>%
  mutate (Group = factor (Group, levels = c("0", "1"), labels = c("standard rehab", "structured rehab")))
  gather (c(NDI:csq_cat), key = var, value = val) %>%
  group_by(Group, Time, var) %>%
  summarize (Mean = mean (val, na.rm = T),
             Sd = sd (val, na.rm = T)) %>%
  ungroup ()

#start plotting

ggplot (data = descr, aes(x = as.factor (Time), y = Mean, ymin=Mean, ymax=Mean+Sd, fill = Group)) +
  geom_bar ( stat="identity", color="black",
            position=position_dodge()) +
  geom_errorbar(colour="black",
                width=.2,
                position=position_dodge(0.9)) +
  scale_fill_manual(values = c("blue", "red")) +
  facet_wrap(~var, ncol = 4, scales = "free") +
  ylab ("Mean values") +
```

```
xlab ("Follow up time (months)") +
ggtitle("Figure 1")
```

Create model of change scores

```
const = c(1:5)

change6 = df.abs[df.abs$Time == 6,-const] - df.abs[df.abs$Time == 0,-const] # 6 - 0 months
change12 = df.abs[df.abs$Time == 12,-const] - df.abs[df.abs$Time == 6,-const] # 12 - 6 months
constant = df.abs[df.abs$Time == 0, c("Sex", "Age", "ZUNG",
                                       "MSPQ", "SES", "TotROM",
                                       "Propriop", "HandStr", "NeckEndr",
                                       "NeckPain", "ArmPain")] # 7:18 is NDI to NeckStr
final = as.data.frame (df.abs[df.abs$Time == 12, c("NDI")] -
                      df.abs[df.abs$Time == 0, c("NDI")])

colnames (constant) = paste0 (colnames (constant) , "_base")
colnames (change6) = paste0 (colnames (change6) , "_d6")
colnames (change12) = paste0 (colnames (change12) , "_d12")
colnames (final) = paste0 (colnames (final) , "_final")

df.bn.orig = cbind (constant, change6, change12, final)

df.bn.orig = select (df.bn.orig, -c (NDI_d6, NDI_d12,
                                     EQ5D_d6, EQ5D_d12)) %>%
  select (-matches ("cop|adp"))

# Excludes baseline absolute values
base.ind = !grepl("_base", colnames (df.bn.orig))
base.ind[c(1,2)] = c(TRUE, TRUE)
df.bn = df.bn.orig[,base.ind] # remove all bases except sex and age, everyone
```

Correlation plot of different variables

```
M <-cor(df.bn[,-c(1,2)], use = "complete.obs")
corrplot(M, type="upper", order="hclust", tl.col="black", tl.srt=45)
```

Create blacklist

```
base.var = grep("_base", colnames (df.bn), value = TRUE)
d6.var = grep("_d6", colnames (df.bn), value = TRUE)
d12.var = grep("_d12", colnames (df.bn), value = TRUE)
```

```

final.var = grep("_final", colnames (df.bn), value = TRUE)

tiers_bl = list (base.var, #to
                  d6.var) #from
bl_1 = tiers2blacklist(tiers = tiers_bl)

tiers_bl = list (base.var, #to
                  d12.var) #from
bl_2 = tiers2blacklist(tiers = tiers_bl)

tiers_bl = list (d6.var, #to
                  d12.var) #from
bl_3 = tiers2blacklist(tiers = tiers_bl)

tiers_bl = list (d12.var, #to
                  final.var) #from
bl_4 = tiers2blacklist(tiers = tiers_bl)

tiers_bl = list (d6.var, #to
                  final.var) #from
bl_5 = tiers2blacklist(tiers = tiers_bl)

tiers_bl = list (base.var, #to
                  final.var) #from
bl_6 = tiers2blacklist(tiers = tiers_bl)

base.bl = expand.grid(base.var, base.var) %>%
  transmute (from = as.character(Var1),
             to = as.character(Var2))
rm.var = which (base.bl$from == base.bl$to)
base.bl = base.bl [-rm.var, ]

final.bl = expand.grid(final.var, final.var) %>%
  transmute (from = as.character(Var1),
             to = as.character(Var2))
rm.var = which (final.bl$from == final.bl$to)
final.bl = final.bl [-rm.var, ]

bl = rbind(bl_1,
            bl_2,
            bl_3,
            bl_4,
            bl_5,
            bl_6,
            base.bl,
            final.bl)

bl <- unique(bl[,c('from', 'to')])

```

Create whitelists

```
##### white listing #####

wl <- matrix(c("HandStr_d6", "HandStr_d12"), nrow = 1, ncol = 2, byrow = TRUE, dimnames = list(NULL, c("HandStr_d6", "HandStr_d12")))

wl.1 <- rbind(wl,
  c("TotROM_d6", "TotROM_d12" ),
  c("Propriop_d6", "Propriop_d12"),
  c("NeckEndr_d6", "NeckEndr_d12"),
  c("csq_cat_d6", "csq_cat_d12"),
  c("NeckPain_d6", "NeckPain_d12"),
  c("ArmPain_d6", "ArmPain_d12"),
  c("ZUNG_d6", "ZUNG_d12"),
  c("SES_d6", "SES_d12"),
  c("MSPQ_d6", "MSPQ_d12"))

wl <- wl.1
```

Do BN analysis

```
doParallel::registerDoParallel(7)
n_boot = 200

df.bn $id = 1:nrow (df.bn )

data_id = df.bn$id
train = df.bn %>%
  sample_frac(0.9)

train_id = train$id
test_id = data_id [-train_id]
test = df.bn [test_id,]

train = train %>% select (-id) %>% as.data.frame()
test = test %>% select (-id) %>% as.data.frame()

#####

boot <- foreach (B = 1: n_boot) %dopar% {
  boot.sample = train[sample(nrow(train),
                             nrow(train), replace = TRUE), ]
  bnlearn::structural.em(boot.sample, impute = "bayes-lw", max.iter = 3,
    maximize.args = list(blacklist = bl,
                          whitelist = wl,
                          k = log(nrow(boot.sample))))
}

#####
```

Fit the data

```
bootstr = custom.strength(boot, nodes = names(train))
avg = averaged.network(bootstr, threshold = 0.7)
fit = bn.fit (avg, df, method = "mle")
imp.data = impute (fit, data = df, method = "bayes-lw")
strength.plot(avg, bootstr, shape = "ellipse")
```

Figure 2

```
g = strength.plot(avg,
                  bootstr,
                  shape = "rectangle",
                  highlight = list (arcs = w1),
                  main = "Figure 2")

graph::nodeRenderInfo(g) = list(fontsize=30)
Rgraphviz::renderGraph(g)
```

Table 1

```
#imp.test <- impute (fit, data = test, method = "bayes-lw")
inames <- names (imp.test) [-c(1:2)]
corr <- structure(numeric(length (inames)), names = inames)

for (var in inames) {
  corr[var] = cor(predict(fit, data = imp.test, var, method = "bayes-lw"), imp.test[, var])
}

corr.df <- data.frame (variables = names (corr),
                      correlation = round (corr, 2))

colorder = c("ZUNG_d6", "MSPQ_d6", "SES_d6", "csq_cat_d6",
             "TotROM_d6", "Propriop_d6", "HandStr_d6", "NeckEndr_d6",
             "ArmPain_d6", "NeckPain_d6",
             "ZUNG_d12", "MSPQ_d12", "SES_d12", "csq_cat_d12",
             "TotROM_d12", "Propriop_d12", "HandStr_d12", "NeckEndr_d12",
             "ArmPain_d12", "NeckPain_d12",
             "NDI_final")

corr.df = corr.df %>%
  mutate (strength = ifelse (abs (correlation) <= 0.3, "negligible",
                            ifelse (abs (correlation) > 0.3 & abs (correlation) <= 0.5 , "low",
                            ifelse (abs (correlation) > 0.5 & abs (correlation) <= 0.7 , "moderate",
                            ifelse (abs (correlation) > 0.7 & abs (correlation) <= 0.9 , "high", "very high"))))
```

```

arrange (match (variables, colorder))
write.table(corr.df, file = "../paper/table_1.txt", sep = "\t", row.names = F)

```

Use the model to answer questions

Does an improvement in ses result in reduction in disability?

```

set.seed (123)

sim = cpdist(fit, nodes = c("SES_d6", "NDI_final"), n = 10^4,
             evidence = (TRUE))

summary (lm(NDI_final ~ SES_d6, data = sim))

tab = table(More_eff = sim$SES_d6 > 0,
            More_abled = sim$NDI_final < quantile(imp.data$NDI_final , 0.5) )

round(prop.table(tab), 2)

plot(sim, col = "grey")
abline(v = 0, col = 2, lty = 2, lwd = 2)
abline(h = 0, col = 2, lty = 2, lwd = 2)
abline(coef(lm(NDI_final ~ SES_d6, data = sim)), lwd = 2)

```

Does an increase in SES at 6 months reduce neck pain at 6 months?

```

set.seed (123)

sim = cpdist(fit, nodes = c("NeckPain_d6", "SES_d6"), n = 10^4,
             evidence = (TRUE))

summary (lm(NeckPain_d6 ~ SES_d6, data = sim))

tab = table(More_eff = sim$SES_d6 > 0,
            Less_pain = sim$NeckPain_d6 < 0)

round(prop.table(tab), 2)

plot(sim, col = "grey")
abline(v = 0, col = 2, lty = 2, lwd = 2)
abline(h = 0, col = 2, lty = 2, lwd = 2)
abline(coef(lm(NeckPain_d6 ~ SES_d6, data = sim)), lwd = 2)

```

Does a reduction in neck pain at 6 months reduce disability at final

```
set.seed (123)

sim = cpdist(fit, nodes = c("NeckPain_d6", "NDI_final"), n = 10^4,
             evidence = (TRUE))

summary (lm(NDI_final ~ NeckPain_d6, data = sim))

tab = table(Less_pain = sim$NeckPain_d6 < 0,
            More_abled = sim$NDI_final < quantile(imp.data$NDI_final , 0.5) )

round(prop.table(tab), 2)

plot(sim, col = "grey")
abline(v = 0, col = 2, lty = 2, lwd = 2)
abline(h = 0, col = 2, lty = 2, lwd = 2)
abline(coef(lm(NDI_final ~ NeckPain_d6, data = sim)), lwd = 2)
```

Does a reduction in neck pain at 6 months reduce arm pain at 6 months

```
set.seed (123)

sim = cpdist(fit, nodes = c("NeckPain_d6", "ArmPain_d6"), n = 10^4,
             evidence = (TRUE))

summary (lm(ArmPain_d6 ~ NeckPain_d6, data = sim))

tab = table(Less_pain = sim$NeckPain_d6 < 0,
            Less_neck_pain = sim$ArmPain_d6 < quantile(imp.data$ArmPain_d6 , 0.5) )

round(prop.table(tab), 2)

plot(sim, col = "grey")
abline(v = 0, col = 2, lty = 2, lwd = 2)
abline(h = 0, col = 2, lty = 2, lwd = 2)
abline(coef(lm(ArmPain_d6 ~ NeckPain_d6, data = sim)), lwd = 2)
```

Let's mutilate neck pain at 6 months and check the direct effect of ses on ndi

```
set.seed (123)

avg.mutilated = mutilated(avg, evidence = list(NeckPain_d6 = 0))
strength.plot(avg.mutilated, bootstr)
```

```

fitted.mutilated = bn.fit (avg.mutilated , df, method = "mle")
fitted.mutilated$NeckPain_d6 = list(coef = c("(Intercept)" = 0), sd = 0)

sim = cpdist(fitted.mutilated, nodes = c("SES_d6", "NDI_final"), n = 10^4,
             evidence = (TRUE))

summary (lm(NDI_final ~ SES_d6, data = sim))

tab = table(More_eff = sim$SES_d6 > 0,
            More_abled = sim$NDI_final < quantile(imp.data$NDI_final , 0.5) )

round(prop.table(tab), 2)

plot(sim, col = "grey")
abline(v = 0, col = 2, lty = 2, lwd = 2)
abline(h = 0, col = 2, lty = 2, lwd = 2)
abline(coef(lm(NDI_final ~ SES_d6, data = sim)), lwd = 2)

```

Let's mutilate arm pain at 12 months, and check direct effect of neck pain on ndi

```

set.seed (123)

avg.mutilated = mutilated(avg.mutilated, evidence = list(ArmPain_d12 = 0))
strength.plot(avg.mutilated, bootstr)

fitted.mutilated = bn.fit (avg.mutilated , df, method = "mle")
fitted.mutilated$ArmPain_d12 = list(coef = c("(Intercept)" = 0), sd = 0)

sim = cpdist(fitted.mutilated, nodes = c("NeckPain_d6", "NDI_final"), n = 10^4,
             evidence = (TRUE))

summary (lm(NDI_final ~ NeckPain_d6, data = sim))

plot(sim, col = "grey")
abline(v = 0, col = 2, lty = 2, lwd = 2)
abline(h = 0, col = 2, lty = 2, lwd = 2)
abline(coef(lm(NDI_final ~ NeckPain_d6, data = sim)), lwd = 2)

```

Let's mutilate arm pain at 12 months, and check effect of ses_d6 on ndi

```

set.seed (123)

```



```

avg.mutilated = mutilated(avg.mutilated, evidence = list(ArmPain_d12 = 0))
strength.plot(avg.mutilated, bootstr)

fitted.mutilated = bn.fit (avg.mutilated , df, method = "mle")
fitted.mutilated$ArmPain_d12 = list(coef = c("(Intercept)" = 0), sd = 0)

sim = cpdist(fitted.mutilated, nodes = c("SES_d6", "NDI_final"), n = 10^4,
             evidence = (TRUE))

summary (lm(NDI_final ~ SES_d6, data = sim))

plot(sim, col = "grey")
abline(v = 0, col = 2, lty = 2, lwd = 2)
abline(h = 0, col = 2, lty = 2, lwd = 2)
abline(coef(lm(NDI_final ~ SES_d6, data = sim)), lwd = 2)

```