Multivariate TRT with 4 Dummies

Soudeh

December 1, 2017

## Install stremr Package Version: 0.8.99 and get Libraries

# Get Source Data from another Github Rasp

library(repmis)  
  
source\_data("https://github.com/Soudi00/Multi-Treatment-Causal-Modeling/blob/master/sampleAD.RData?raw=True")

## Downloading data from: https://github.com/Soudi00/Multi-Treatment-Causal-Modeling/blob/master/sampleAD.RData?raw=True

## SHA-1 hash of the downloaded data file is:  
## 89e884ef132014c520d5a0bd50ed7c05757ff6e0

## [1] "AD"

AD = as.data.table(AD, key= c(ID,SEQ))

# I have 4 different treatment , should I only use 3 of the dummies in the importData or I should include all of them?

# ----------------------------------------------------------------------  
# Define intervention (always OI/IUI with Oral Meds):  
# ----------------------------------------------------------------------  
AD[, ("zero.set1") := 0L]  
AD[, ("zero.set2") := 0L]  
AD[, ("zero.set3") := 0L]  
AD[, ("TRT1.set") := 1L]  
  
# ----------------------------------------------------------------------  
# Import Data in to stremr object  
# ----------------------------------------------------------------------  
OData.1 <- importData(AD, ID = "ID", t\_name = "SEQ",   
 covars = c("CAT\_VAR1","CAT\_VAR2","CONT\_VAR1"),   
 CENS = c("CNS","ADM\_CNS"),   
 TRT = c("TRT1","TRT2","TRT3","TRT4"),  
 MONITOR = NULL, OUTCOME = "STATUS",  
 weights = NULL, remove\_extra\_rows = TRUE,  
 verbose = getOption("stremr.verbose"))

## [1] "stremr will use the following options as defaults: "  
##   
## List of 7  
## $ fit\_method : chr "none"  
## $ fold\_column : NULL  
## $ lower\_bound\_zero\_Q: logi TRUE  
## $ skip\_update\_zero\_Q: logi TRUE  
## $ up\_trunc\_offset : num 20  
## $ low\_trunc\_offset : num -10  
## $ eps\_tol : num 1e-05

## ...converting the following factor(s) to binary dummies (and droping the first factor levels): TRTN,CAT\_VAR1,CAT\_VAR2

# ----------------------------------------------------------------------  
# Look at the input data object  
# ----------------------------------------------------------------------  
print(OData.1)

## <DataStorageClass>  
## Public:  
## addYnode: function (YnodeVals, det.Y)   
## backup.savedGstarsDT: active binding  
## backupNodes: function (nodes)   
## check\_norows\_after\_event: function ()   
## clone: function (deep = FALSE)   
## convert.to.wide: function (bslcovars)   
## curr\_data\_A\_g0: TRUE  
## dat.sVar: active binding  
## define.stoch.nodes: function (NodeNames)   
## define\_CVfolds: function (nfolds = 5, fold\_column = "fold\_ID", seed = NULL)   
## det.Y: NULL  
## emptydat.sVar: active binding  
## eval\_follow\_rule: function (NodeName, gstar.NodeName)   
## eval\_follow\_rule\_each\_t: function (NodeName, gstar.NodeName)   
## eval\_uncensored: function ()   
## eval\_uncensored\_idx: function ()   
## evalsubst: function (subset\_vars, subset\_exprs = NULL)   
## fold\_column: NULL  
## follow\_rule: active binding  
## g\_holdout\_preds: NULL  
## g\_preds: NULL  
## get.dat.sVar: function (rowsubset = TRUE, covars)   
## get.outvar: function (rowsubset = TRUE, var)   
## get.sVar: function (name.sVar)   
## gstarNodes\_stoch: NULL  
## initialize: function (Odata, nodes, YnodeVals, det.Y, noCENScat, ...)   
## interventionNodes.g0: NULL  
## interventionNodes.gstar: NULL  
## IPWeights\_info: NULL  
## IPwts\_by\_regimen: active binding  
## is.sVar.CENS: function (name.sVar)   
## make\_origami\_fold\_from\_column: function (subset\_idx)   
## max.t: active binding  
## min.t: active binding  
## modelfit.gA: NULL  
## modelfit.gC: NULL  
## modelfit.gN: NULL  
## modelfits.g0: NULL  
## names.sVar: active binding  
## ncols.sVar: active binding  
## new.factor.names: list  
## nfolds: NULL  
## nobs: active binding  
## noCENScat: 0  
## nodes: active binding  
## noNA.Ynodevals: active binding  
## nuniqueIDs: active binding  
## nuniquets: active binding  
## Qlearn.fit: NULL  
## replaceNodesVals: function (subset\_idx, nodes\_to\_repl = intervened\_NODE, source\_for\_repl = NodeNames)   
## rescaleNodes: function (subset\_idx, nodes\_to\_rescale, delta)   
## restoreNodes: function (nodes)   
## set.sVar: function (name.sVar, new.sVarVal)   
## swapNodes: function (current, target)   
## type.sVar: function (var\_names, pcontinuous = 0.05)   
## uncensored: active binding  
## YnodeVals: NULL  
## Private:  
## .active.bin.sVar: NULL  
## .follow\_rule: NULL  
## .IPwts: NULL  
## .mat.bin.sVar: NULL  
## .mat.sVar: data.table, data.frame  
## .nodes: list  
## .ord.sVar: NULL  
## .protected.YnodeVals: NULL  
## .saveGstarsDT: NULL  
## .uncensored: NULL

# ----------------------------------------------------------------------  
# Access the input data  
# ----------------------------------------------------------------------  
get\_data(OData.1)  
  
# ----------------------------------------------------------------------  
# Model the Right Censroing and Adminstrative Censoring and Exposure  
# ----------------------------------------------------------------------  
gform\_CENS <- "CNS + ADM\_CNS ~ CAT\_VAR1 + CONT\_VAR1"  
gform\_TRT = "TRT1+TRT2+TRT3+TRT4 ~ CAT\_VAR1 + CAT\_VAR2 + CONT\_VAR1"  
  
# ----------------------------------------------------------------------  
# Fit Propensity Scores  
# ----------------------------------------------------------------------  
  
OData.1 <- fitPropensity(OData.1, gform\_CENS = gform\_CENS,ngform\_TRT = gform\_TRT )

## Using the default regression formula: TRT1+TRT2+TRT3+TRT4 ~ CAT\_VAR1\_2+CAT\_VAR1\_3+CAT\_VAR1\_4+CAT\_VAR1\_6+CAT\_VAR1\_7+CAT\_VAR2\_2+CAT\_VAR2\_3+CAT\_VAR2\_4+CAT\_VAR2\_5+CONT\_VAR1

## [1] "New 'ModelBinomial' regression defined:"  
## [1] "P(CNS|CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: ;\\ N: NA"  
## [1] "New 'ModelBinomial' regression defined:"  
## [1] "P(ADM\_CNS|CNS, CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: CNS == 0;\\ N: NA"  
## [1] "New 'ModelBinomial' regression defined:"  
## [1] "P(TRT1|CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CAT\_VAR2\_2, CAT\_VAR2\_3, CAT\_VAR2\_4, CAT\_VAR2\_5, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: ;\\ N: NA"  
## [1] "New 'ModelBinomial' regression defined:"  
## [1] "P(TRT2|TRT1, CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CAT\_VAR2\_2, CAT\_VAR2\_3, CAT\_VAR2\_4, CAT\_VAR2\_5, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: ;\\ N: NA"  
## [1] "New 'ModelBinomial' regression defined:"  
## [1] "P(TRT3|TRT1, TRT2, CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CAT\_VAR2\_2, CAT\_VAR2\_3, CAT\_VAR2\_4, CAT\_VAR2\_5, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: ;\\ N: NA"  
## [1] "New 'ModelBinomial' regression defined:"  
## [1] "P(TRT4|TRT1, TRT2, TRT3, CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CAT\_VAR2\_2, CAT\_VAR2\_3, CAT\_VAR2\_4, CAT\_VAR2\_5, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: ;\\ N: NA"  
## [1] "fitting the model: P(CNS|CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: ;\\ N: NA"  
## [1] "fitting the model: P(ADM\_CNS|CNS, CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: CNS == 0;\\ N: NA"  
## [1] "fitting the model: P(TRT1|CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CAT\_VAR2\_2, CAT\_VAR2\_3, CAT\_VAR2\_4, CAT\_VAR2\_5, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: ;\\ N: NA"  
## [1] "fitting the model: P(TRT2|TRT1, CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CAT\_VAR2\_2, CAT\_VAR2\_3, CAT\_VAR2\_4, CAT\_VAR2\_5, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: ;\\ N: NA"  
## [1] "fitting the model: P(TRT3|TRT1, TRT2, CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CAT\_VAR2\_2, CAT\_VAR2\_3, CAT\_VAR2\_4, CAT\_VAR2\_5, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: ;\\ N: NA"  
## [1] "fitting the model: P(TRT4|TRT1, TRT2, TRT3, CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CAT\_VAR2\_2, CAT\_VAR2\_3, CAT\_VAR2\_4, CAT\_VAR2\_5, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: ;\\ N: NA"

# What should be the dimension of the intervened\_TRT when we are using multiple dummy treatment

I have my own defined dynamic treatment patterns of interest (5 dummy variables for the 5 patterns). That is:

* Always TRT1 (PATH1)
* Always TRT2 (PATH2)
* Always TRT3 (PATH3)
* Start TRT1, switch at any time to TRT3 (PATH4)
* Start TRT2, switch at any time to TRT3 (PATH5)

# ----------------------------------------------------------------------  
# Error: length(intervened\_NODE) not equal to length(NodeNames)  
# ----------------------------------------------------------------------  
  
wts.DT.1 <- getIPWeights(OData = OData.1, intervened\_TRT="PATH1")

## Error: length(intervened\_NODE) not equal to length(NodeNames)

# ----------------------------------------------------------------------  
# Error in modelfit.g$getPsAsW.models()[[i]] : subscript out of bounds  
# ----------------------------------------------------------------------  
  
wts.DT.1 <- getIPWeights(OData = OData.1, intervened\_TRT=c("TRT1.set","zero.set1","zero.set2","zero.set3"))

## Error in modelfit.g$getPsAsW.models()[[i]]: subscript out of bounds

# ----------------------------------------------------------------------  
# useing diffrent intervened\_TRT didnt make a diffrence in the result  
# ----------------------------------------------------------------------  
  
wts.DT.1 <- getIPWeights(OData = OData.1, useonly\_t\_TRT="PATH1==1",rule\_name ="Only TRT1")  
wts.DT.1

## ID SEQ STATUS g0.A g0.C g0.N g0.CAN gstar.C gstar.A  
## 1: 1 1 0 0.4054166 0.9674897 1 0.39223636 1 0.4054166  
## 2: 1 2 0 0.3585701 0.9590887 1 0.34390048 1 0.3585701  
## 3: 1 3 0 0.4708409 0.9764564 1 0.45975562 1 0.4708409  
## 4: 1 4 1 0.3425051 0.9557057 1 0.32733405 1 0.3425051  
## 5: 2 1 0 0.6816217 0.7165851 1 0.48843991 1 0.6816217  
## ---   
## 302: 100 1 0 0.2799980 0.8564319 1 0.23979923 1 0.2799980  
## 303: 100 2 0 0.2955584 0.8757783 1 0.25884365 1 0.2955584  
## 304: 100 3 0 0.5260467 0.8543336 1 0.44941938 1 0.5260467  
## 305: 100 4 0 0.4804644 0.8784357 1 0.42205709 1 0.4804644  
## 306: 100 5 0 0.5419484 0.1156405 1 0.06267119 0 0.5419484  
## gstar.N gstar.CAN wt.by.t cum.IPAW N.follow.rule cum.stab.P  
## 1: 1 0.4054166 1.033603 1.033603 81 0.81  
## 2: 1 0.3585701 1.042656 1.077693 63 0.63  
## 3: 1 0.4708409 1.024111 1.103677 50 0.50  
## 4: 1 0.3425051 1.046347 1.154829 35 0.35  
## 5: 1 0.6816217 1.395508 1.395508 81 0.81  
## ---   
## 302: 1 0.2799980 1.167635 1.167635 81 0.81  
## 303: 1 0.2955584 1.141841 1.333254 63 0.63  
## 304: 1 0.5260467 1.170503 1.560578 50 0.50  
## 305: 1 0.4804644 1.138387 1.776542 35 0.35  
## 306: 1 0.0000000 0.000000 0.000000 25 0.25  
## rule.name  
## 1: Only TRT1  
## 2: Only TRT1  
## 3: Only TRT1  
## 4: Only TRT1  
## 5: Only TRT1  
## ---   
## 302: Only TRT1  
## 303: Only TRT1  
## 304: Only TRT1  
## 305: Only TRT1  
## 306: Only TRT1

wts.DT.2 <- getIPWeights(OData = OData.1, useonly\_t\_TRT="PATH2==1", rule\_name = "Only TRT2")  
wts.DT.2

## ID SEQ STATUS g0.A g0.C g0.N g0.CAN gstar.C gstar.A  
## 1: 1 1 0 0.4054166 0.9674897 1 0.39223636 1 0.4054166  
## 2: 1 2 0 0.3585701 0.9590887 1 0.34390048 1 0.3585701  
## 3: 1 3 0 0.4708409 0.9764564 1 0.45975562 1 0.4708409  
## 4: 1 4 1 0.3425051 0.9557057 1 0.32733405 1 0.3425051  
## 5: 2 1 0 0.6816217 0.7165851 1 0.48843991 1 0.6816217  
## ---   
## 302: 100 1 0 0.2799980 0.8564319 1 0.23979923 1 0.2799980  
## 303: 100 2 0 0.2955584 0.8757783 1 0.25884365 1 0.2955584  
## 304: 100 3 0 0.5260467 0.8543336 1 0.44941938 1 0.5260467  
## 305: 100 4 0 0.4804644 0.8784357 1 0.42205709 1 0.4804644  
## 306: 100 5 0 0.5419484 0.1156405 1 0.06267119 0 0.5419484  
## gstar.N gstar.CAN wt.by.t cum.IPAW N.follow.rule cum.stab.P  
## 1: 1 0.4054166 1.033603 1.033603 81 0.81  
## 2: 1 0.3585701 1.042656 1.077693 63 0.63  
## 3: 1 0.4708409 1.024111 1.103677 50 0.50  
## 4: 1 0.3425051 1.046347 1.154829 35 0.35  
## 5: 1 0.6816217 1.395508 1.395508 81 0.81  
## ---   
## 302: 1 0.2799980 1.167635 1.167635 81 0.81  
## 303: 1 0.2955584 1.141841 1.333254 63 0.63  
## 304: 1 0.5260467 1.170503 1.560578 50 0.50  
## 305: 1 0.4804644 1.138387 1.776542 35 0.35  
## 306: 1 0.0000000 0.000000 0.000000 25 0.25  
## rule.name  
## 1: Only TRT2  
## 2: Only TRT2  
## 3: Only TRT2  
## 4: Only TRT2  
## 5: Only TRT2  
## ---   
## 302: Only TRT2  
## 303: Only TRT2  
## 304: Only TRT2  
## 305: Only TRT2  
## 306: Only TRT2