

# Data Analysis Homework 4

Jimmy Hickey

11/10/2020

## 1

```
knitr::opts_chunk$set(echo = TRUE)
library(DynTxRegime)
```

```
## Loading required package: modelObj
```

```
library(rgenoud)
```

```
## ## rgenoud (Version 5.8-3.0, Build Date: 2019-01-22)
## ## See http://sekhon.berkeley.edu/rgenoud for additional documentation.
## ## Please cite software as:
## ## Walter Mebane, Jr. and Jasjeet S. Sekhon. 2011.
## ## ‘Genetic Optimization Using Derivatives: The rgenoud package for R.’
## ## Journal of Statistical Software, 42(11): 1-26.
## ##
```

```
library(modelObj)
library(rpart)
```

## a

```
ld1 = read.table("LDL.dat.txt", header=FALSE)
# remove ID column
ld1 = ld1[,-1]
names(ld1) = c("L1", "A1", "L2", "S2", "A2", "L3",
               "S3", "A3", "L4", "S4", "A4", "Y", "S5")
```

```
#####
# Decision 4
#####
```

```
fSet4 = function(S4){
  # can be (0,1) or (0)
```

```

# label them option S42 and S41
# these are A_k,2 and A_k,1 in problem statement
subsets = list( list("S42", c(0,1)),
                list("S41", c(0)))

txOpts = rep(x = NA, times = length(x = S4))

txOpts[ S4 == 0] = "S42"
txOpts[ S4 == 1] = "S41"

# need named list
return( list("subsets" = subsets, "txOpts" = txOpts))
}

moPropen_S42 = buildModelObjSubset(model = ~ L4,
                                   solver.method = "glm",
                                   solver.args = list("family" = "binomial"),
                                   predict.args = list("type" = "response"),
                                   subset = "S42",
                                   dp = 4L )

moPropen_S41 = buildModelObjSubset(model = ~ L4,
                                   solver.method = "glm",
                                   solver.args = list("family" = "binomial"),
                                   predict.args = list("type" = "response"),
                                   subset = "S41",
                                   dp = 4L )

moPropen_list4 = list(moPropen_S42, moPropen_S41)

regime = list("S42" = ~ L4, "S41" = ~ L4)

# BOWL STEP
# see Holloway slide 81
# note we take response -1 to minimize instead of maximize
# course website tried lambdas between 10^-4 and 10^-1
# was having issues with later lambdas, so I just chose 0.05
bowlObj4 = bowl(moPropen = moPropen_list4,
                data = ldl,
                response = -1 * ldl$Y,
                txName = "A4",
                regime = regime,
                BowlObj = NULL,
                lambdas = 0.05,
                kernel = list("S42" = "linear", "S41" = "linear"),
                kparam = NULL,
                fSet = fSet4,
                surrogate = 'hinge',
                verbose = FALSE

```

)

## NOTE: subset(s) S41 received tx not in accordance with specified feasible tx sets

```
#####  
# Decision 3  
#####  
  
fSet3 = function(S3){  
  
  # can be (0,1) or (0)  
  # label them option S32 and S31  
  # thse are A_k,2 and A_k,1 in problem statement  
  subsets = list( list("S32", c(0,1)),  
                  list("S31", c(0)))  
  
  txOpts = rep(x = NA, times = length(x = S3))  
  
  txOpts[ S3 == 0] = "S32"  
  txOpts[ S3 == 1] = "S31"  
  
  # need named list  
  return( list("subsets" = subsets, "txOpts" = txOpts))  
}  
  
moPropen_S32 = buildModelObjSubset(model = ~ L3,  
                                   solver.method = "glm",  
                                   solver.args = list("family" = "binomial"),  
                                   predict.args = list("type" = "response"),  
                                   subset = "S32",  
                                   dp = 3L )  
  
moPropen_S31 = buildModelObjSubset(model = ~ L3,  
                                   solver.method = "glm",  
                                   solver.args = list("family" = "binomial"),  
                                   predict.args = list("type" = "response"),  
                                   subset = "S31",  
                                   dp = 3L)  
  
moPropen_list3 = list(moPropen_S32, moPropen_S31)  
  
regime3 = list("S32" = ~ L3, "S31" = ~ L3)  
  
bowlObj3 = bowl(moPropen = moPropen_list3,  
                data = ld1,  
                response = -1 * ld1$Y,  
                txName = "A3",  
                regime = regime3,  
                BowlObj = bowlObj4,  
                lambdas = 0.05,  
                kernel = list("S32" = "linear", "S31" = "linear"),
```

```

kparam = NULL,
fSet = fSet3,
surrogate = 'hinge',
verbose = FALSE
)

```

## NOTE: subset(s) S31 received tx not in accordance with specified feasible tx sets

```

#####
# Decision 2
#####
fSet2 = function(S2){

  # can be (0,1) or (0)
  # label them option S22 and S21
  # these are A_k,2 and A_k,1 in problem statement
  subsets = list( list("S22", c(0,1)),
                  list("S21", c(0)))

  txOpts = rep(x = NA, times = length(x = S2))

  txOpts[ S2 == 0] = "S22"
  txOpts[ S2 == 1] = "S21"

  # need named list
  return( list("subsets" = subsets, "txOpts" = txOpts))
}

moPropen_S22 = buildModelObjSubset(model = ~ L2,
                                   solver.method = "glm",
                                   solver.args = list("family" = "binomial"),
                                   predict.args = list("type" = "response"),
                                   subset = "S22",
                                   dp = 2L )

moPropen_S21 = buildModelObjSubset(model = ~ L2,
                                   solver.method = "glm",
                                   solver.args = list("family" = "binomial"),
                                   predict.args = list("type" = "response"),
                                   subset = "S21",
                                   dp = 2L )

moPropen_list2 = list(moPropen_S22, moPropen_S21)

regime2 = list("S22" = ~ L2, "S21" = ~ L2)

bowlObj2 = bowl(moPropen = moPropen_list2,
                data = ld1,
                response = -1 * ld1$Y,
                txName = "A2",

```

```

regime = regime2,
BowlObj = bowlObj3,
lambdas = 0.05,
kernel = list("S22" = "linear", "S21" = "linear"),
kparam = NULL,
fSet = fSet2,
surrogate = 'hinge',
verbose = FALSE
)

```

## NOTE: subset(s) S21 received tx not in accordance with specified feasible tx sets

```

#####
# Decision 1
#####
fSet1 = function(data){

  # no side effects at 1
  # so only make decision based on LDL
  subsets = list( list("S12", c(0,1)))

  txOpts = rep(x = "S12", times = dim(data)[1])

  # need named list
  return( list("subsets" = subsets, "txOpts" = txOpts))

}

moPropen_S12 = buildModelObjSubset(model = ~ L1,
                                   solver.method = "glm",
                                   solver.args = list("family" = "binomial"),
                                   predict.args = list("type" = "response"),
                                   subset = "S12",
                                   dp = 1L )

moPropen_list1 = list(moPropen_S12)

regime1 = list("S12" = ~ L1)

bowlObj1 = bowl(moPropen = moPropen_list1,
                data = ld1,
                response = -1 * ld1$Y,
                txName = "A1",
                regime = regime1,
                BowlObj = bowlObj2,
                lambdas = 0.05,
                kernel = list("S12" = "linear"),
                kparam = NULL,
                fSet = fSet1,
                surrogate = 'hinge',
                verbose = FALSE
                )

```

```
estimator(bowl0bj1)
```

```
## [1] -119.8855
```

**b**

Here we get a value  $\hat{\mathcal{V}}(d^{opt}) = 119.8855$ . This is larger of 103.6736 that I got from question 2 of homework 3.

**2**

```
smart = read.table("SMART.dat.txt", header=FALSE)
names(smart) = c("X11", "X12", "X13", "A1",
                 "X21", "X22", "R2", "A2", "Y")
```

**a**

**i**

**ii**

**iii**

**b**

**c**

**d**

**i**

**ii**

**e**

**f**