iForm Use

Kirk Gosik

March 30, 2016

Overview

The following are two small examples of using the iForm selection procedure. The procedure can handle ultra-high dimensional datasets but for demonstrationg purposes built in R datasets were used. Each of the iForm functions takes in two arguments. The first is the dataset being used and the second is the name of the variable from that data set used as the response.

mtcars

The the first dataset used is an R dataset called mtcars. For more information about this dataset you can see the R help menu. Each observation is a car and there are several variables measured on each car. I am using all the variables in the dataset to predict the horsepower(hp) of the car in the following examples. The first fit is uing the iForm procedure with a stong heredity principle. This assumes that both main effects need to be currently selected before considering and interaction effect. As you can see from the output that Number of cylinders (cyl) and Number of carburetors (carb) are in the model before the interaction effect of cyl.carb (Number of cylinders by Number of carburetors) was selected. You would interpret the coefficientst in the same fashion as any other linear model. For example, the coefficient for cyl is 7.83. Therefore for every 1 additional cylinder the horse power goes up by 7.833.

```
  # Sourcing iForm functions
source("iForm.R")

names(mtcars)  # Variable names
```

```
##  [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
## [11] "carb"
```

```
dim(mtcars)  # dimensions of the dataset
```

```
## [1] 32 11
```

```
sum(is.na(mtcars$hp)) # checking number of NA values
```

```
## [1] 0
```

```
help("mtcars") # more info on mtcars

  # Running iForm on R dataset mtcars using hourse power as response (hp)
iForm.fit1 <- iForm.model.matrix(mtcars, "hp")
iForm.fit1
```

```
##
## Call:
## lm(formula = y ~ . + 0, data = model)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.528 -13.851  -4.278  13.522  38.749
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## cyl         7.833      2.297   3.410  0.00199 **
## carb      -58.232     12.229  -4.762 5.32e-05 ***
## cyl.carb   10.317      1.557   6.627 3.45e-07 ***
## gear       18.211      4.563   3.991  0.00043 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.42 on 28 degrees of freedom
## Multiple R-squared:  0.9873, Adjusted R-squared:  0.9855
## F-statistic: 545.8 on 4 and 28 DF,  p-value: < 2.2e-16
```

iForm weak version

Here is the same dataset but with the iForm seleciton under weak heredity. This assumes that only one of the main effects needs to be previously selected for an interaction to be considered. The same two main effects were selected as above first, cyl then carb. After that however there is a differen interaction selected of carb.disp (Number of carburetors by Displacement). This less restrictive assumption improved the model accuracy by incresaing the adjusted R-square from 0.9855 under the strong assumption to 09864 under the weak assumption. The choice comes down to the practical search space. A question you may want to answer is, do you feel that both main effects need to be included before an interaction could be considered. If yes, use the strong case, if not use the weak case.

```
  # fitting the iForm procedure with the weak heredity assumption
iForm.weak.fit1 <- iForm.weak(mtcars, "hp")
iForm.weak.fit1
```

```
##
## Call:
## lm(formula = y ~ . + 0, data = model)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.197 -12.572   2.662  13.235  31.347
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## cyl       27.93912    3.59200   7.778 1.79e-08 ***
## carb      -7.49242    4.44952  -1.684 0.103324
## carb.disp  0.11937    0.01707   6.994 1.32e-07 ***
## cyl.wt    -4.31711    1.05287  -4.100 0.000321 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 18.81 on 28 degrees of freedom
## Multiple R-squared:  0.9881, Adjusted R-squared:  0.9864
## F-statistic: 582.4 on 4 and 28 DF,  p-value: < 2.2e-16
```

Hitters

Here is another example that has NA values in the dataset. This would need to be removed before running the procedure. The interpretation would be similar.

```
  # Using a slightly larger dataset from the package ISLR

  # If package isn't installed this code will install it
list.of.packages <- c("ISLR")
new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]
if(length(new.packages)) install.packages(new.packages)

library(ISLR)
names(Hitters)  # Variable names
```

```
##  [1] "AtBat"     "Hits"      "HmRun"     "Runs"      "RBI"
##  [6] "Walks"     "Years"     "CAtBat"    "CHits"     "CHmRun"
## [11] "CRuns"     "CRBI"      "CWalks"    "League"    "Division"
## [16] "PutOuts"   "Assists"   "Errors"    "Salary"    "NewLeague"
```

```
dim(Hitters)  # dimensions of the dataset
```

```
## [1] 322  20
```

```
sum(is.na(Hitters$Salary))  # number of NA values in the Salary variable
```

```
## [1] 59
```

```
Hitters <- na.omit(Hitters)  # removing all na values
dim(Hitters)  # rechecking dimension size after removal of NA's
```

```
## [1] 263  20
```

```
sum(is.na(Hitters))   # rechecking number of NA values
```

```
## [1] 0
```

```
help("Hitters") # more information on Hitters dataset

iForm.fit2 <- iForm.model.matrix(Hitters, "Salary")
iForm.fit2
```

```
##
## Call:
## lm(formula = y ~ . + 0, data = model)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -670.30 -127.28  -10.05  124.69 2092.57
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## Hits       -1.2821316  2.1147763  -0.606 0.544877
## CRBI        0.7136095  0.2740357   2.604 0.009757 **
## Hits.CRBI   0.0165156  0.0046434   3.557 0.000448 ***
## PutOuts     0.2976200  0.0667894   4.456 1.25e-05 ***
## Walks       1.8741779  1.1540083   1.624 0.105608
## AtBat       0.2038724  0.5875577   0.347 0.728893
## CRBI.AtBat -0.0031871  0.0013214  -2.412 0.016582 *
## CRuns       1.9290128  0.3711853   5.197 4.18e-07 ***
## CRBI.CRuns -0.0009795  0.0001507  -6.499 4.26e-10 ***
## CAtBat     -0.1814535  0.0529089  -3.430 0.000706 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 283 on 253 degrees of freedom
## Multiple R-squared:  0.8427, Adjusted R-squared:  0.8365
## F-statistic: 135.6 on 10 and 253 DF,  p-value: < 2.2e-16
```

C. Elegans

The following example is from the C Elegans dataset I analyzed in the paper from Briefings in Bioinformatics. Here is the first gene that should to have some interaction effects between the SNPs.

```
 # gene expression data
expression <- t(read.table("M_quantile.txt"))
dim(expression)
```

```
## [1]   208 20401
```

```
colnames(expression)[1:20] # first 20 gene expression names
```

```
##  [1] "A_12_P108045" "A_12_P108879" "A_12_P115540" "A_12_P101622"
##  [5] "A_12_P105685" "A_12_P117297" "A_12_P104093" "A_12_P120371"
##  [9] "A_12_P117675" "A_12_P114854" "A_12_P119331" "A_12_P101256"
## [13] "A_12_P119545" "A_12_P118044" "A_12_P120220" "A_12_P119449"
## [17] "A_12_P105363" "A_12_P107391" "A_12_P119234" "A_12_P106869"
```

```
  # SNP data
genotype <- read.table("genotype.txt")
dim(genotype)
```

```
## [1]  208 1454
```

```r
colnames(genotype)[1:20] # first 20 SNP names
```

```
##  [1] "X_350084"  "X_387180"  "X_478348"  "X_543068"  "X_547548"
##  [6] "X_596094"  "X_636850"  "X_718523"  "X_734972"  "X_789666"
## [11] "X_911871"  "X_982564"  "X_1001412" "X_1127311" "X_1182001"
## [16] "X_1215425" "X_1310698" "X_1358166" "X_1447260" "X_1447824"
```

```r
  # Transposing data to remove duplicated snp values
genotype.t <- t(genotype)
genotype <- t(unique(genotype.t)) # removing duplicate values and transposing back
genotype <- genotype - 0.5 # Centering genotypes

  # combining to 1 dataset to run analysis
analysisdata <- data.frame(A_12_P114854  = expression[,"A_12_P114854"], genotype)

# running iForm on dataset

iForm.fit3 <- iForm.model.matrix(analysisdata, "A_12_P114854")
iForm.fit3
```

```
##
## Call:
## lm(formula = y ~ . + 0, data = model)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.56601 -0.10758  0.00027  0.11948  0.76306
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## X_2924225           -3.43193    0.12650 -27.130   <2e-16 ***
## X_2877421            3.46139    0.12794  27.055   <2e-16 ***
## X_3015791           -0.05026    0.07686  -0.654    0.514
## X_2877421.X_3015791  6.94690    0.25706  27.025   <2e-16 ***
## X_2924225.X_3015791 -6.68014    0.25205 -26.503   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1697 on 203 degrees of freedom
## Multiple R-squared:  0.8873, Adjusted R-squared:  0.8846
## F-statistic: 319.8 on 5 and 203 DF,  p-value: < 2.2e-16
```