forward-selection report

Xiao Liu

the motivation of using step wise selection

In real life there might be too much potential factors affect the responses we're interested,

for example, to model the accident rate on highway we might have predictors that describes the location of the highway, weather, status of the road and the speed limits.

or when we try to model the rate of having lung cancer, we might consider the predictors of air quality, frequency of smoking cigarette, genetic issue

Including all predictors will reduce the precision of our model, but ignore too much predictors might also makes the result of model untrustworthy, so selecting appropriate predictors is critical for setting up accurate model.

Step wise selection is a method that helps us select predictors that should be included in model.

One of the variations of step wise method is forward selection.

The idea of forward selection is starting at model consist only intercept, then consider all model consisting one additional regressor . compute the AIC score of each model and compare the AIC score of all models.

the formula of AIC is given by

$$AIC = nlog(\frac{RSS}{n}) + 2p$$

In this formula, RSS mean residual sum of square, and p stands for the number of regressors.

since we want model to be accurate, we want a model has low RSS value. we also want our model to be simple, so we want p as low as possible.

We keep the model that has the lowest AIC score, and repeat the previous step again, until there is no more additional regressor or our current model has the lowest AIC score.

we use Robey.txt as example

```r
rob = read.table("Robey.txt")
summary(rob)
```

```
##      region               tfr        contraceptors
##  Length:50          Min.   :1.700   Min.   : 4.00
##  Class :character   1st Qu.:3.600   1st Qu.:12.25
##  Mode  :character   Median :4.600   Median :41.00
##                     Mean   :4.688   Mean   :37.44
##                     3rd Qu.:5.975   3rd Qu.:55.00
##                     Max.   :7.300   Max.   :77.00
```

we use tfr as response, start at m1, which has only intercept

```r
m1 = lm(tfr ~ 1, data = rob)##current model
m2 = lm(tfr ~ 1 + contraceptors, data = rob)##intercept plus additional predictors contraceptors
m3 = lm(tfr ~ 1 + region, data = rob)##intercept plus additioanl predictors region
```

calculate the AIC value of each model,

```
AIC(m1, m2, m3)
```

```
##    df       AIC
## m1  2 182.27747
## m2  3  90.42072
## m3  5 160.34591
```

we keep m2, and repeat the previous step, since we have additional predictor region

```
m4 = lm(tfr ~ 1 + contraceptors + region, data = rob)
AIC(m2, m4)
```

```
##    df      AIC
## m2  3 90.42072
## m4  6 90.82472
```

so according to forward selection, the best model should be m2

the code for forward selection

Here are the two function that uses forward selection methods to find predictors.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
onestep_forward_selection = function(respons_e, star_t, en_d) {
  #in this function, respons_e is the response vector, star_t is the data frame that contains all
  #the predictors in the current model,
  #en_d is data.frame contains all the predictors that are not included in the current model
  original = star_t
  mstart = lm(respons_e ~ ., data = star_t)
  AIC_score = rep(NA, dim(en_d)[2])
  AIC_mstart = AIC(mstart)
  for (i in 1:dim(en_d)[2]) {
    newdata = star_t
    newdata$new_data = unlist(en_d[, i])
    names(newdata)[dim(newdata)[2]] = colnames(en_d[i])
    mtest = lm(respons_e ~., data = newdata)
    AIC_score[i] = AIC(mtest)
  }
  if (min(AIC_score) <= AIC_mstart) {
    index = which.min(AIC_score)
    mydata = star_t
    mydata$new_data = en_d[, index]
    names(mydata)[dim(mydata)[2]] = colnames(en_d[index])
    return(list(mydata, "go"))
  }
  else {
    return(list(original, "stop"))
  }
}
```

the onestep_forward_selection function returns the best model in each step. the output is the data frame that includes all the predictors in the best model our function chooses.

```
selection_function = function(respons, begin_data, full_data) {
  #the respons variable represent the responses in the model, the begin_data is the data frame
  #that includes the predictors in our initial model
  #full_data is the full data frame.
  dro = colnames(begin_data)#we extract the column names of all the predictors in our initial model
  en_dd = full_data %>%
    select(-one_of(dro))
  sta_t = begin_data
  while(TRUE) {
    result = onestep_forward_selection(respons_e = respons, star_t = sta_t, en_d = en_dd)
    if (result[[2]] == "go") {
      sta_t = result[[1]]
      dropp = colnames(sta_t)
      en_dd = full_data %>%
        select(-one_of(dropp))
      if (dim(en_dd)[2] == 0) {
        return(result[[1]])
        break
      }
    }
    else if (result[[2]] == "stop") {
      return(result[[1]])
      break
    }
  }
}
```

the selection_function run the one step forward selection function in each step. if the current model has the lowest AIC score, the function return the data frame contains the predictors in the current model. if we find any new model has lower AIC score than our current model, we uses that new model as our current model and repeat the previous step, until we have no more predictors left, so the output of selection_function contains all the predictors in the model that has lowest AIC score.

use our selection function to obtain the model from data Robey.txt

```
rob$region = as.factor(rob$region)
star_t = subset(rob, select = c(3))
respons_e = unlist(subset(rob, select = c(2)))
#en_d = subset(rob, select = c(1))
dat_a = rob
#a = onestep_forward_selection(respons_e, star_t, en_d)
selection_function(respons = respons_e, begin_data = star_t, full_data = dat_a)
```

```
##                    contraceptors tfr
## Botswana                      35 4.8
## Burundi                        9 6.5
## Cameroon                      16 5.9
## Ghana                      13 6.1
## Kenya                      27 6.5
## Liberia                     6 6.4
## Mali                        5 6.8
## Mauitius                   75 2.2
## Niger                       4 7.3
## Nigeria                     6 5.7
## Senegal                    12 6.4
## Sudan                       9 4.8
## Swaziland                  21 5.0
## Tanzania                   10 6.1
## Togo                       12 6.1
## Uganda                      5 7.2
## Zambia                     15 6.3
## Zimbabwe                   45 5.3
## Bangladesh                 40 5.5
## China                      72 2.5
## India                      45 4.3
## Indonesia                  50 3.0
## Korea.Rep.of               77 1.7
## Pakistan                   12 5.2
## Philippines                34 4.3
## Sri.Lanka                  62 2.7
## Thailand                   68 2.3
## Vietnam                    53 3.9
## Belize                     47 4.5
## Bolivia                    32 4.9
## Brazil                     66 3.6
## Columbia                   66 2.8
## Costa.Rica                 70 3.6
## Dom.Republic               56 3.3
## Ecuador                    53 3.8
## El.Salvador                47 4.6
## Guatemala                  23 5.6
## Haiti                      10 6.0
## Jamaica                    55 2.9
## Mexico                     55 4.0
## Panama                     58 4.0
## Paraguay                   48 4.6
## Peru                       59 3.5
## Trinidad.Tobago            54 3.1
## Egypt                      40 4.6
## Jordan                     35 5.5
## Morocco                    42 4.0
## Tunisia                    51 4.3
## Turkey                     60 3.4
## Yemen                       7 7.0
```

the result shows that contraceptors should be the only predictor we need to use. this result is consistent with the result we got before.

simulated data

in this section we will set up a simulated data, and then use our method to find the best model based on the simulated data. the parameter is given below

```
simulat = data.frame("x1" = c(1:20), "x2" = runif(20, 0, 10), "x3" = rbinom(20, 1, .5),
                     "x4" = rnorm(20))
#the data frame has 4 columns x1, x2, x3, x4.

b0 = 17 #parameter for intercept
b1 = 0.5 #parameter for x1
b2 = 0.3 #parameter for x2
b3 = -5.2 #parameter for x3
sigma = 1.4
eps = rnorm(simulat$x1, 0, sigma) #error
y = b0 + b1*simulat$x1 + b2*simulat$x2 + b3*simulat$x3 + eps #the response y
```

now we will use the forward selection function to help us find the predictors for y.

```
inital = subset(simulat, select = c(1))
interes = y
en = simulat %>%
  select(-one_of(colnames(inital)))
l = selection_function(respons = interes, begin_data = inital, full_data = simulat)
#k = onestep_forward_selection(respons_e = interes, star_t = inital, en_d = en)
#k[[2]]
l
```

```
##    x1 x3         x2
## 1   1  0 8.22843018
## 2   2  0 7.46100578
## 3   3  0 6.69202902
## 4   4  0 0.70941839
## 5   5  0 2.84515006
## 6   6  0 5.14489330
## 7   7  1 8.42233966
## 8   8  0 3.67410766
## 9   9  1 0.02987189
## 10 10  0 6.57529730
## 11 11  1 4.48179621
## 12 12  1 4.60883059
## 13 13  0 5.41617113
## 14 14  0 1.68698269
## 15 15  0 4.71396258
## 16 16  1 6.83130448
## 17 17  1 3.64993327
## 18 18  0 8.85683463
## 19 19  0 7.77958960
## 20 20  0 9.90355481
```

the result of the function shows that our model should contains all the predictors in our data except x4. this is what we expect since x4 is not related with our response y

based on the result of our forward selection function, our final model would be(y ~ x1 + x2 + x3)

```
mfinal = lm(y ~ x1 + x2 + x3, data = simulat)
summary(mfinal)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3, data = simulat)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.53462 -0.70666 -0.05646  0.60215  2.55578
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.13552    0.78879  21.724 2.66e-13 ***
## x1           0.44929    0.05147   8.729 1.76e-07 ***
## x2           0.38333    0.11086   3.458  0.00324 **
## x3          -4.66404    0.64501  -7.231 2.01e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.272 on 16 degrees of freedom
## Multiple R-squared:  0.9045, Adjusted R-squared:  0.8866
## F-statistic: 50.51 on 3 and 16 DF,  p-value: 2.211e-08
```

```
interval = data.frame(confint(mfinal))
interval$realvalue = c(17, 0.5, 0.3, -5.2)
interval
```

```
##                 X2.5..    X97.5.. realvalue
## (Intercept) 15.4633694 18.8076759      17.0
## x1           0.3401805  0.5584060       0.5
## x2           0.1483152  0.6183393       0.3
## x3          -6.0314093 -3.2966755      -5.2
```

no predictors have high p value. this shows that all predictors help explain the change of the response. the confidence data frame shows that every interval contains the true parameter value.