Model Selection Using Stepwise Function

Pranav Gopalkrishna, 1940223

2021-09-02

# **Setting work directory**

setwd("~/Documents/Study/computerScience/programming/r/data/")

# Data set and purpose of using it

The data set used here does not have an apparent practical use, as its fields are simply one generic response variable with nine generic independent variables, i.e. y, x1, x2... x9.

The purpose of working with this data set is to figure out the best fitting model for the response y among every possible model that can be made using the available regressors. Hence, we need to decide which regressors must be selected, and how they must be fitted to the data i.e. how should y be modelled using these selected regressors. Note that ultimately, we are comparing the possible models, to find out which one is best.

myData = read.csv("justSomeData.csv")

head(myData)

## y x1 x2 x3 x4 x5 x6 x7 x8 x9  
## 1 25.9 4.9176 1 3.472 0.998 1 7 4 42 0  
## 2 29.5 5.0208 1 3.531 1.500 2 7 4 62 0  
## 3 27.9 4.5429 1 2.275 1.175 1 6 3 40 0  
## 4 25.9 4.5573 1 4.050 1.232 1 6 3 54 0  
## 5 29.9 5.0597 1 4.455 0.988 1 6 3 56 0  
## 6 30.9 5.8980 1 5.850 1.240 1 7 3 51 1

# Model with all regressors (raw model)

A full model with all the available regressors is to be created. The best fitting model will be created using this raw model as the source for drawing and studying the available regressors.

rawModel = lm(y~., data = myData)  
summary(rawModel)

##   
## Call:  
## lm(formula = y ~ ., data = myData)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.8504 -1.4017 0.0929 1.7541 3.7206   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 17.11351 5.88549 2.908 0.0131 \*  
## x1 2.39009 1.05740 2.260 0.0432 \*  
## x2 5.74422 4.35113 1.320 0.2114   
## x3 0.12998 0.52530 0.247 0.8087   
## x4 2.63623 4.34493 0.607 0.5553   
## x5 2.32382 1.46160 1.590 0.1378   
## x6 -1.62471 2.40137 -0.677 0.5115   
## x7 -0.09723 3.38794 -0.029 0.9776   
## x8 -0.04445 0.06212 -0.716 0.4879   
## x9 2.03656 1.97372 1.032 0.3225   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.841 on 12 degrees of freedom  
## Multiple R-squared: 0.8774, Adjusted R-squared: 0.7854   
## F-statistic: 9.539 on 9 and 12 DF, p-value: 0.0003125

# Step function

## Purpose

This function chooses the best regression model using the AIC stepwise model selection algorithm. Best in this context means the model that is has the regressors and coefficients that best explain or match the responses, given the data. Hence, it is not only best fitting for a given set of regressors, it is also best fitting among all possible models using the available regressors.

*(AIC stands for* ***Akaike’s information criterion****. It is a stepwise model selection method compares the quality of a set of statistical models to each other)*

## Usage

The following only presents the range of options we will be using for this function. There are more options, however.

step( object,  
 scope,  
 direction = c("both", "backward", "forward"))

**Argument “object”** is an object representing a model of an appropriate class (mainly “lm” and “glm”). This is used as the initial model in the stepwise search (variable selection) for the best regressors for modelling the given response. Initial model implies the model with the response and an initial set of regressors and coefficients on top of which more regressors will be added. Typically, it is a model with only the response, intercept and error term.

**Argument “scope”** defines the range of models examined in the stepwise search. It holds the model or models containing the different regressors that may be selected for the final model returned by the function. This option could contain a single model, or two models “lower” and “upper”, wherein the regressors in the lower model are a subset of the regressors in the upper model. In the case of “lower” and “upper” models, the step function performs a stepwise search for every model from the lower to the upper (and the models in between, with respect to rhe regressors used).

**Argument "direction"** the mode of stepwise search, can be one of "both", "backward", or "forward", with a default of "both". If the scope argument is missing the default for direction is "backward". Forward implies that we start with less regressors, and keep adding and fitting more.

# Making the initial model i.e. the "object"

The initial model in our case is simply a model containing the response being explained by the intercept and its associated error term. No regressors are added yet.

initialModel = lm(y~1, data = myData)  
# 1 as the regressor implies the constant term, whose coefficient is the intercept.  
summary(initialModel)

##   
## Call:  
## lm(formula = y ~ 1, data = myData)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.095 -5.071 1.405 3.655 10.805   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 34.995 1.308 26.77 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.133 on 21 degrees of freedom

# Performing the stepwise search for the best model

step(initialModel, direction = "forward", scope = formula(rawModel))

## Start: AIC=80.78  
## y ~ 1  
##   
## Df Sum of Sq RSS AIC  
## + x1 1 616.67 173.16 49.390  
## + x2 1 386.12 403.71 68.012  
## + x4 1 385.20 404.63 68.062  
## + x3 1 339.51 450.32 70.416  
## + x6 1 208.00 581.83 76.053  
## + x5 1 199.58 590.25 76.369  
## + x8 1 132.88 656.95 78.725  
## <none> 789.83 80.777  
## + x7 1 57.89 731.94 81.103  
## + x9 1 43.18 746.65 81.540  
##   
## Step: AIC=49.39  
## y ~ x1  
##   
## Df Sum of Sq RSS AIC  
## + x2 1 22.9619 150.20 48.260  
## + x9 1 15.9875 157.17 49.259  
## <none> 173.16 49.390  
## + x4 1 7.8167 165.34 50.374  
## + x5 1 5.6693 167.49 50.657  
## + x6 1 3.5583 169.60 50.933  
## + x3 1 3.2496 169.91 50.973  
## + x7 1 2.4360 170.72 51.078  
## + x8 1 1.7536 171.41 51.166  
##   
## Step: AIC=48.26  
## y ~ x1 + x2  
##   
## Df Sum of Sq RSS AIC  
## <none> 150.20 48.260  
## + x9 1 11.3028 138.90 48.539  
## + x7 1 8.3644 141.83 49.000  
## + x5 1 7.6678 142.53 49.107  
## + x6 1 6.5813 143.62 49.274  
## + x8 1 6.2771 143.92 49.321  
## + x3 1 4.2757 145.92 49.625  
## + x4 1 0.3311 149.87 50.212

##   
## Call:  
## lm(formula = y ~ x1 + x2, data = myData)  
##   
## Coefficients:  
## (Intercept) x1 x2   
## 9.321 2.923 5.550

From the function’s results, we see that x1 and x1 are the best regressors for our response y, with the given coefficients leading to the best fitting model possible for the data and available regressors.

# Final model (best model)

cookedModel = lm(y~x1+x2, data = myData)  
cookedModel

## Call:  
## lm(formula = y ~ x1 + x2, data = myData)  
## Coefficients:  
## (Intercept) x1 x2  
## 9.321 2.923 5.550   
summary(cookedModel)

##   
## Call:  
## lm(formula = y ~ x1 + x2, data = myData)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.8510 -2.0998 0.0266 1.3604 5.1215   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 9.3207 3.1373 2.971 0.00785 \*\*   
## x1 2.9232 0.5162 5.663 1.85e-05 \*\*\*  
## x2 5.5497 3.2563 1.704 0.10462   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.812 on 19 degrees of freedom  
## Multiple R-squared: 0.8098, Adjusted R-squared: 0.7898   
## F-statistic: 40.46 on 2 and 19 DF, p-value: 1.418e-07

# Conclusions

Hence, we see that the best fitting model i.e. the most explanatory model for the response variable is given by

*y = 9.321 + 2.923x1 + 5.550x2*

Based on the summary, the intercept and x1 are significant given a 0.05 significance level, but not x2. However, adding x2 seemingly makes the best fitting model. We also see that the R-squared value is 80.98 (and the adjusted R-squared is 78.98%), meaning that around 75-80% of the variation in the response in the sample is explained by the model.