Best subset variable selection based on Allen's PRESS-Statistic

Sebastian Schütz, Konstantin Thiel

09.07.2021

Introduction

Within this report variable selection based on **Allen's PRESS-statistic** is presented. Variable selection itself is a popular topic now more than ever. Due to increasing machine resources such variable selection algorithms, which are mostly already existing for a long time, are now possible to execute even on larger scale. The desire to find relations between variables based on algorithms has been present for some time too. It comes from the idea that selection based on algorithms is more objective and therefore in some way better. Another reason may be that one is getting the feeling that no connections were overlooked with such algorithms. It has already been found that variable selection based on algorithms is not automatically the best solution, nevertheless it can help to find a *good* set of variables to describe data. Although the basic procedures like **backward** or **forward elimination** exist for some time, there is still research done to improve or investigate these algorithms.

Mathematical background

Within this tutorial the **best subset selection**, a very common approach will be applied. This algorithm is often based on criteria like **AIC, BIC, R-squared**, ... Especially for this way of variable selection increasing machine resources are important. The procedure is as follows:

1. Define ...

- the criteria on which the selection process is based on.
- the set of variables selection should be done on.
- the range of subset sizes you want to find best subsets on (optional).

2. The algorithm finds the *best* subset of variables for each subset size according to the defined criteria.

The selection process shows why the computational effort should not be underestimated. For a working set of 10 variables, 2^{10} models have to be calculated in order to find the *best* according to a given criteria.

Before referring to **Allen's PRESS-statistic** we want to give a short overview of other common criteria:

- **AIC** (Akaike's information criteria): $AIC = -2l(\hat{\theta}_{ML}) + 2k$ with k denoting the number of explanatory variables in the model and $l(\hat{\theta}_{ML})$ describing the log-likelihood of the vector θ which includes k coefficients that result from the maximum-likelihood-estimation. In the case of linear models the AIC can be calculated directly as $AIC = n \cdot ln(\sigma^2) + 2k$. [1]
- **BIC** (Bayesian information criteria): $BIC = -2l(\hat{\theta}_{ML}) + k \cdot ln(n)$. AIC and BIC differ only regarding the penalty term. While AIC always adds $2k$, BIC considers sample size n. It can be concluded that for $n > 7$ ($ln(8) > 2$), BIC is larger than AIC. [1]

The idea behind these measures is to reward a good model fit and punish for model complexity. The difference to the **PRESS-statistc** is, that a good model fit is assessed different. Instead of maximum-likelihood-estimation, the goodness of prediction for every point in the data set is assessed in the following way:

$$PRESS = \sum_{i=1}^{n} (y_i - \hat{y}_{-i})^2$$

with $\hat{y_{-i}}$ describing the expected value excluding the i-th observation. In other words: For every point in the data set, the goodness of prediction of a model which is based on the remaining $n-1$ data points is calculated (as squared difference between the actual and predicted value). The sum of these predicted errors is then describing the PRESS-statistic which stands for **PRE**diction **S**um of **S**quares. [2]

Although this criterion is not one of the latest findings, there is a connection to a very actual topic: Artificial intelligence. Here, the PRESS-statistic is equivalent to a special case of cross validation, the *leave-one-out* cross validation. The standard procedure follows three steps:

1. Split the data set into n so-called *folds* (n-*fold* cross validation).
2. Combine $n-1$ folds to a training data set, build a model (in AI often a classifier or similar) on it and test it on the n-th fold.
3. Do this for all folds and average the error.

Now, leave-one-out cross validation also follows the described steps with the difference that one fold consists of only one data point. This method is also referred to in the literature as the *Jackknife* method. [3]

Especially for larger data sets calculation of PRESS-statistic is cumbersome. According to the description one would need to fit a separate model for every data point in order to estimate the outcome. However, this problem can be circumvented for linear models by using the hat matrix $H = X(X^TX)^{-1}X^T$ to calculate $\hat{y}_{-i}$ in the following way: If H_{ii} denotes the i-th diagonal entry of H,

$$\hat{y}_{-i} = \frac{\hat{y}_i - H_{ii}y_i}{1 - H_{ii}}$$

From there it follows, that

$$PRESS = \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{(1 - H_{ii})^2}$$

That means, that only one linear model has to be built for all n data points. This approach is also used in the following tutorial. A more detailed mathematical explanation can be found here [4][5].

Implementation in R

Framework/structure

For implementation in R two packages were used:

- **MPV** for calculation of the PRESS-statistic
- **dplyr** for data preparation

Furthermore three functions were defined:

- *selectPredictors* selects a subset of the explanatory variables based on an integer

```
#' Extract predictor names from bitcode.
#'
#' @param code An integer in the interval [0, 2^length(predictors) - 1] that
#' encodes the selected predictors
#' @param predictors A vector of strings with predictor names
selectPredictors <- function(code, predictors) {
  selected <- c()
  for (i in 1:length(predictors)) {
    if (code %% 2 == 1)  # check if current predictor is encoded
      selected <- c(selected, predictors[i])
```

```
    code <- code %/% 2  # right shift to obtain next predictor
  }
  return(selected)
}
```

- *computePRESS* calculates the PRESS-statistic for the given explanatory variables (all combinations per subset size)

```
#' Compute PRESS for a set of predictors specified by the given code
#'
#' @param code An integer in the interval [0, 2^length(predictors) - 1] that
#' encodes the selected predictors
#' @param target Name of the target variable in the linear model
#' @param predictors A vector of strings with predictor names
#' @param data Dataframe that contains measurements for the target and all
#' predictors
computePRESS <- function(code, target, predictors, data) {
  p <- selectPredictors(code, predictors)
  if (is.null(p))  # empty model (iff code == 0)
    p <- "1"
  formula <- paste(target, "~", paste(p, collapse = "+"))
  model <- lm(formula, data)
  return(MPV::PRESS(model))
}
```

- *bestPRESS* selects best model (according to PRESS) per subset size and outputs it as matrix

```
#' @param target Name of the target variable in the linear model
#' @param predictors A vector of strings with predictor names
#' @param data Dataframe that contains measurements for the target and all

bestPRESS_df <- function(target, predictors, data) {
    nModels <- 2^length(predictors)
    modelCodes <- seq(0, nModels - 1)
    presses <- sapply(X = modelCodes, FUN = computePRESS, target, predictors, data)
    nrPredictors <- unlist(lapply(sapply(X = modelCodes, FUN = selectPredictors,
        predictors), FUN = length))
    presses_predictors_df = data.frame(modelCodes, presses, nrPredictors)
    minCode_nrPredictors <- presses_predictors_df %>%
        arrange(nrPredictors, presses) %>%
        distinct(nrPredictors, .keep_all = T)
    bestPredictors <- sapply(X = minCode_nrPredictors$modelCodes, FUN = selectPredictors,
        predictors)
    bestPredictorsdf <- data.frame(do.call(rbind, bestPredictors))
    bestPredictorsdf[upper.tri(bestPredictorsdf)] <- NA
    return(bestPredictorsdf)
}
```

Application on real life data

It follows a short demonstration based on a data set on bodyfat.

case	brozek	siri	density	age	weight	weight_kg	height	height_cm	neck	chest	abdomen	hip	thigh	knee	ankle	biceps	forearm	wrist
1	12.6	12.3	1.0708	23	154.25	70.1	67.75	172	36.2	93.1	85.2	94.5	59.0	37.3	21.9	32.0	27.4	17.1
2	6.9	6.1	1.0853	22	173.25	78.8	72.25	184	38.5	93.6	83.0	98.7	58.7	37.3	23.4	30.5	28.9	18.2
3	24.6	25.3	1.0414	22	154.00	70.0	66.25	168	34.0	95.8	87.9	99.2	59.6	38.9	24.0	28.8	25.2	16.6
4	10.9	10.4	1.0751	26	184.75	84.0	72.25	184	37.4	101.8	86.4	101.2	60.1	37.3	22.8	32.4	29.4	18.2
5	27.8	28.7	1.0340	24	184.25	83.8	71.25	181	34.4	97.3	100.0	101.9	63.2	42.2	24.0	32.2	27.7	17.7
6	20.6	20.9	1.0502	24	210.25	95.6	74.75	190	39.0	104.5	94.4	107.8	66.0	42.0	25.6	35.7	30.6	18.8

Applying the algorithm only consists of two steps:

1. Select dependent and independent variables.
2. Start the algorithm with the function *bestPRESS* and pass the selected variables and the data set to the function.

```
# dependent variable
target <- "siri"

# compute best subset of predictors for a lm after manual pre-selection
predictors <- c("age", "weight_kg", "height_cm", "neck", "chest", "abdomen", "hip",
    "thigh", "knee", "ankle", "biceps", "forearm", "wrist")

start.time = Sys.time()
best_subsets = bestPRESS_df(target, predictors, data)
end.time = Sys.time()
```

The result shows a matrix consisting the best model according to the PRESS-statistic for each subset size:

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13
abdomen												
weight_kg	abdomen											
weight_kg	abdomen	wrist										
age	height_cm	abdomen	wrist									
age	height_cm	chest	abdomen	wrist								
age	height_cm	neck	abdomen	forearm	wrist							
age	height_cm	neck	chest	abdomen	forearm	wrist						
age	height_cm	neck	chest	abdomen	biceps	forearm	wrist					
age	height_cm	neck	chest	abdomen	hip	thigh	forearm	wrist				
age	height_cm	neck	chest	abdomen	hip	thigh	biceps	forearm	wrist			
age	height_cm	neck	chest	abdomen	hip	thigh	knee	biceps	forearm	wrist		
age	weight_kg	height_cm	neck	chest	abdomen	hip	thigh	knee	biceps	forearm	wrist	
age	weight_kg	height_cm	neck	chest	abdomen	hip	thigh	knee	ankle	biceps	forearm	wrist

Although, already using the more efficient calculation of PRESS by use of the hat matrix the execution for a data set with 251 rows and 13 explanatory variables takes 17.13 seconds.

Application on artificial data

In addition, the algorithm is applied on artificial data as well. The approach for data generation was taken from [6] and slightly modified.

```
set.seed(1234)
n.sample <- 100
error <- rnorm(n.sample, 0, 0.8)

x1 <- runif(n.sample, -2, 2)
x2 <- runif(n.sample, -1, 4)
```

```r
x3 <- sample(c(0,1),n.sample, replace=T)
x4 <- 0.8*x1 + rnorm(n.sample, 1, 0.5)
x5 <- 0.4*x1 + rnorm(n.sample, 2, 0.5)
x6 <- -0.5*x1 + rnorm(n.sample, 0, 0.5)
x7 <- rnorm(n.sample, 2, 0.5)
x8 <- 0.9*x2 + rnorm(n.sample, 0, 0.05)
X <- matrix(NA, n.sample, 9)
X[,1] <- x1
X[,2] <- x2
X[,3] <- x3
X[,4] <- x4
X[,5] <- x5
X[,6] <- x6
X[,7] <- x7
X[,8] <- x8
# true model
X[,9] <- 10 + 0.5*x1 - 1*x2 + 3*x3 + error
colnames(X) <- c("x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8","y")


fit1 <- lm(y~., data = as.data.frame(X))
summary(fit1)
```

```
##
## Call:
## lm(formula = y ~ ., data = as.data.frame(X))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5079 -0.5923 -0.1510  0.4340  2.1244
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.280510   0.510285  18.187   <2e-16 ***
## x1           0.229281   0.187220   1.225   0.2239
## x2          -2.511774   1.496098  -1.679   0.0966 .
## x3           3.256554   0.164204  19.832   <2e-16 ***
## x4           0.055209   0.180145   0.306   0.7599
## x5           0.186486   0.150517   1.239   0.2185
## x6          -0.008305   0.149419  -0.056   0.9558
## x7          -0.030037   0.179602  -0.167   0.8676
## x8           1.760410   1.667251   1.056   0.2938
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7962 on 91 degrees of freedom
## Multiple R-squared:  0.9015, Adjusted R-squared:  0.8928
## F-statistic: 104.1 on 8 and 91 DF,  p-value: < 2.2e-16
```

It can be seen, that the selection algorithm is choosing x1,x2 and x3 as *best* explanatory variables which reflects the true model.

```r
# dependent variable
target_art <- "y"
```

```
# compute best subset of predictors for a lm after manual pre-selection
predictors_art <- c("x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8")

best_subsets = bestPRESS_df(target_art, predictors_art, as.data.frame(X))
```

x3							
x2	x3						
x1	x2	x3					
x1	x2	x3	x5				
x1	x2	x3	x5	x8			
x1	x2	x3	x5	x7	x8		
x1	x2	x3	x4	x5	x7	x8	
x1	x2	x3	x4	x5	x6	x7	x8

Comparison to other algorithms

In order to test the performance of the selection algorithm based on PRESS, a comparison with other criteria within best subset selection is done in this chapter. For comparison the package *leaps* is used. Comparison is done on both data sets.

Leaps offers the possibility to find *nbest* models per subset size. Therefore the results can be compared easily for each size. Since information criteria (like AIC, BIC, DIC, ...) differ only in how model complexity is penalized, the *best* models found by the function *regsubsets* do not depend on which information criteria is used (because only models of same sizes are compared).

```
form = as.formula(paste(target, "~", paste(predictors, collapse = "+")))
models <- regsubsets(form, data = data, method = "exhaustive", nbest = 1,
    nvmax = length(predictors))
```

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13
abdomen												
weight_kg	abdomen											
weight_kg	abdomen	wrist										
age	height_cm	abdomen	wrist									
age	height_cm	chest	abdomen	wrist								
age	height_cm	chest	abdomen	biceps	wrist							
age	height_cm	neck	chest	abdomen	forearm	wrist						
age	height_cm	neck	chest	abdomen	biceps	forearm	wrist					
age	height_cm	neck	chest	abdomen	hip	thigh	forearm	wrist				
age	height_cm	neck	chest	abdomen	hip	thigh	biceps	forearm	wrist			
age	height_cm	neck	chest	abdomen	hip	thigh	ankle	biceps	forearm	wrist		
age	weight_kg	height_cm	neck	chest	abdomen	hip	thigh	ankle	biceps	forearm	wrist	
age	weight_kg	height_cm	neck	chest	abdomen	hip	thigh	knee	ankle	biceps	forearm	wrist

```
form = as.formula(paste(target_art, "~", paste(predictors_art, collapse = "+")))
models <- regsubsets(form, data = as.data.frame(X), method = "exhaustive", nbest = 1,
    nvmax = length(predictors))
```

x3							
x2	x3						
x1	x2	x3					
x1	x2	x3	x5				
x1	x2	x3	x5	x8			
x1	x2	x3	x4	x5	x8		
x1	x2	x3	x4	x5	x7	x8	
x1	x2	x3	x4	x5	x6	x7	x8

Conclusion

The whole algorithm for using PRESS as criteria is implemented by three functions, where only the *bestPRESS* function is needed for execution. This guarantees an easy handling for all sorts of applications. The user only has to provide the data set (*data*) along with the information which variables are explanatory (*predictors*) and which is the dependent one (*target*).

For the bodyfat data set it could be shown that overall similar variables were selected in comparison to best subset selection with AIC/BIC/... Only 3/13 models were different. However, it cannot be validated how well variable selection performance was in general on this data set because no information about the true model was available.

The results for the artificial data set were even more similar to each other. Only 1/8 models were different between PRESS and AIC/BIC/... The increase on similarity could also be attributed to the rather low complexity of the data set.

The main goal of this tutorial was to provide a working **best subset selection** algorithm with **PRESS** as selection criteria, which was done. For more detailed comparison with other selection algorithms further investigations would be needed.

Limitations

- Interaction terms

It has not yet been discussed how to deal with interaction terms. Although it needs some additional preparation, it is possible to add interactions within the *predictors*-vector by just combining the variable names with **:** as separator. For few interaction terms this can be done by hand. At some point *paste*-function can be useful to adapt the *predictors*-vector.

```
# example
predictors <- c("age", "weight_kg", "height_cm", "neck", "chest", "abdomen",
                "hip", "thigh", "knee", "ankle", "biceps", "forearm", "wrist","age:knee")

best_subsets = bestPRESS_df(target, predictors, data)
```

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14
abdomen													
weight_kg	abdomen												
weight_kg	abdomen	wrist											
height_cm	abdomen	wrist	age:knee										
height_cm	chest	abdomen	wrist	age:knee									
height_cm	chest	abdomen	forearm	wrist	age:knee								
height_cm	neck	chest	abdomen	forearm	wrist	age:knee							
height_cm	neck	chest	abdomen	biceps	forearm	wrist	age:knee						
height_cm	neck	chest	abdomen	hip	thigh	forearm	wrist	age:knee					
height_cm	neck	chest	abdomen	hip	thigh	biceps	forearm	wrist	age:knee				
height_cm	neck	chest	abdomen	hip	thigh	knee	biceps	forearm	wrist	age:knee			
age	height_cm	neck	chest	abdomen	hip	thigh	knee	biceps	forearm	wrist	age:knee		
age	weight_kg	height_cm	neck	chest	abdomen	hip	thigh	knee	biceps	forearm	wrist	age:knee	
age	weight_kg	height_cm	neck	chest	abdomen	hip	thigh	knee	ankle	biceps	forearm	wrist	age:knee

- Algorithm runtime

Variable selection is often performed in larger data sets. It was shown that with 13 explanatory variables the algorithm already takes some time. Since the number of models increases exponentially, it has to be checked if the usage of the algorithm is still feasible at some point.

Sources

1. Sachs, L., & Hedderich, J. (2006). Angewandte Statistik: Methodensammlung mit R. Springer-Verlag.

2. Allen, D. M. (1974). The relationship between variable selection and data agumentation and a method for prediction. technometrics, 16(1), 125-127.
3. Kruse, R., Borgelt, C., Braune, C., Mostaghim, S., Steinbrecher, M., Klawonn, F., & Moewes, C. (2011). Computational intelligence. Vieweg+ Teubner Verlag.
4. https://statisticaloddsandends.wordpress.com/2018/07/30/the-press-statistic-for-linear-regression/
5. http://statweb.stanford.edu/~owen/courses/305a/305MinNotesMarked.pdf
6. https://www.rpubs.com/lictof/480764