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

Sebastian Schütz and Konstantin Thiel

14.07.2021

Sources available under https://github.com/Lime91/best-press.

Introduction

Variable selection is a typical task in many data analysis projects. The goal is to find relations between variables based on *selection algorithms*. These algorithms are considered to be objective, and therefore, preferable to pure manual selection. Another appealing characteristic is the conception that algorithms are precise and do not overlook relationships between variables. However, it has been found that variable selection based on algorithms does not automatically yield the best solution [7]. Nevertheless, it can help to find a *good* set of variables to describe data. Although the basic procedures like *backward* or *forward selection* exist for many years, there is still research to improve or investigate these algorithms today.

In this tutorial we present variable selection for linear models based on **Allen's PRESS-statistic**. Since powerful computers are widely used nowadays, we combine this criterion with the **best subset selection** algorithm, which is usually more expensive to evaluate that backward or forward selection. To the best of our knowledge, there is currently no R-package that implements the combination of PRESS with best subset selection. Thus, we contribute our own implementation of this technique.

Fundamentals

Best Subset Selection

Best subset selection uses a model *quality criterion* such as *AIC, BIC, R-squared* or, in our case, the PRESS statistic. The algorithm works as follows:

1. Define …

- the criterion on which the selection process is based on.
- the set of variables from which to select.
- the range of different subset sizes to investigate (optional).

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

The computational effort of this algorithm should not be underestimated. For a set of 10 variables, 2^{10} models have to be evaluated in order to find the best one according to the criterion, i.e., the number of models to be evaluated grows exponentially (!) in the variable set size. Therefore, we suggest this technique only for small to medium sets of variables and recommend users to perform a manual preselection (e.g., based on domain expert knowledge) beforehand.

Model Quality Criteria

Before introducing the PRESS-statistic we give a short overview of other common criteria:

- **AIC** (Akaike's information criteria): $AIC = -2l(\hat{\theta}_{ML}) + 2k$ where k denotes the number of explanatory variables in the model and $l(\hat{\theta}_{ML})$ describes 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 computed 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 with respect to 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 always larger than AIC [1].

The rationale behind these two measures is to reward a good model fit and punish for model complexity. The **PRESS-statistic** assesses model quality in a different way. Instead of maximum-likelihood-estimation, the prediction quality for every point in the dataset is estimated as follows:

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

where $\hat{y}_{-i}$ denotes the estimated response value for the i-th observation based on a model computed with this observation omitted. In other words, for every point in the dataset, the prediction quality of a model which is based on the remaining $n-1$ data points is evaluated (as squared difference between the actual and predicted value). The PRESS-statistic is defined as the sum of these errors. Its name 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 the currently trending topics of Machine Learning and AI. In these domains, the PRESS-statistic is equivalent to a special case of *cross validation*, a standard procedure that follows three steps:

1. Split the dataset into n so-called *folds* (*n-fold* cross validation).
2. Combine $n-1$ folds to a training dataset, 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.

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

For large datasets, the computation of PRESS according to the three steps listed above is cumbersome. According to the description we would need to fit a separate model for every data point. However, this problem can be circumvented for linear models by exploiting 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, we have:

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

From this it follows:

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

In other words, only one linear model has to be built to compute PRESS. This makes the combination of PRESS with best subset selection feasible and we use this approach also in our implementation in the subsequent section. A more detailed mathematical explanation can be found in [4][5].

Implementation in R

Framework/structure

Our implementation uses two R-packages:

- MPV for calculation of the PRESS-statistic
- dplyr for data handling

We define the following functions:

selectPredictors

Return the set of variable names (a subset of `predictors`) that are encoded in an integer `code`. Each bit in `code` represents a single variable. Hence, `code` must be in the interval `[0, 2^length(predictors) - 1]`.

```
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)
}
```

bitSum

Given an integer `code`, count all bits that are set to `1`. This function is used to compute the subset sizes.

```
bitSum <- function(code, nBits=32) {
  s <- 0
  while (nBits > 0) {
    if (code %% 2 == 1)  # check if current bit is set
      s <- s + 1
    code <- code %/% 2  # right shift to obtain next bit
    nBits <- nBits - 1
  }
  return(s)
}
```

computePress

Compute PRESS for a set of variables specified by the given `code`. `target` is the (string) name of the response variable and `predictors` is the vector of all explanatory variable names (including the ones that are not selected). `data` is the dataframe that contains measurements for `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))
}
```

bestPressDf

Select best model according to PRESS per subset size and outputs it as matrix. Note that the empty model is omitted.

```
bestPressDf <- function(target, predictors, data) {
    nPred <- length(predictors)
    nModels <- 2^nPred
    modelCodes <- seq(1, nModels - 1)  # skip empty model
    modelSizes <- sapply(modelCodes, bitSum, nPred)
    presses <- sapply(modelCodes, computePress, target, predictors, data)
    bestCodes <- data.frame(modelCodes, modelSizes, presses) %>%
        arrange(modelSizes, presses) %>%
        distinct(modelSizes, .keep_all = T) %>%
        dplyr::select(modelCodes)  # namespace only needed for RMarkdown
    bestPred <- apply(bestCodes, 1, selectPredictors, predictors)  # sizes differ
    bestDf <- lapply(bestPred, function(v) c(v, rep(NA, nPred - length(v)))) %>%
        as.data.frame(row.names = paste("X", seq(nPred), sep = "")) %>%
        t()
    rownames(bestDf) <- seq(nPred)
    return(bestDf)
}
```

Application to Real-World Data

We provide a short demonstration based on a dataset on body fat introduced in [8].

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 target and explanatory variables.
2. Start the algorithm with the function `bestPressDf` and pass the selected variables and the dataset to the function.

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

# compute best subset of predictors for a lm after manual pre-selection
start.time = Sys.time()
best_subsets = bestPressDf(target, predictors, data)
end.time = Sys.time()
```

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

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 we already compute PRESS efficiently by exploiting the hat matrix, the execution for the provided dataset with 251 rows and 13 explanatory variables takes 18.14 seconds on the machine that is used to compile this tutorial.

Application to Artificial Data

We apply the technique to artificially generated data. The data generation was taken from [6] and slightly modified for our purposes.

```r
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)
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")

# full model
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
```

A full model that includes all available variables yields that only one of the true predictor variables is significant at $\alpha = 0.05$. Thus, we apply our selection technique to investigate whether we are able to recover the true model.

```
target_art <- "y"
predictors_art <- c("x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8")
best_subsets = bestPressDf(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

The selection algorithm chooses `x1`, `x2` and `x3` as *best* explanatory variables for a model of size 3. This reflects the true model.

Comparison to other algorithms

In order to test the performance of the selection algorithm based on PRESS, we perform a comparison with other criteria in this chapter. We use the package `leaps` that provides a best subset selection technique based on likelihood information criteria. Our comparison is performed on both the body fat and the artificial dataset. Note that the various information criteria (AIC, BIC, DIC, ...) differ only in how model complexity is penalized. Thus, if we fix the model size, the *best* model is independent of the choice of a specific information criterion.

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

NB: `method = "exhaustive"` corresponds to best subset selection; `nbest = 1` means that only one model for each subset size is reported; `nvmax = length(predictors)` determines that all possible subset sizes are investigated.

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

We implemented an algorithm that uses PRESS as criterion for variable selection in linear models. This required only a few functions with `bestPressDf` providing a simple API for users. The function suggests the best set of variables for each possible model size. Users must only provide a dataset (`data`) along with the information of which variables are explanatory (`predictors`) and which is the dependent one (`target`).

On the bodyfat dataset, we could show that PRESS selected similar subsets in comparison to a selection based on a likelihood criterion (AIC, BIC, . . .). Only 3/13 models suggested by both algorithms were different. However, it cannot be stated how close any of the suggested models is to the true model since there is no information about a true model for this dataset.

The results for the artificial dataset were even more similar to each other. Only 1/8 models was different between PRESS and the likelihood criteria. This increase in similarity might be attributed to the rather low complexity of the artificial dataset.

The main goal of this tutorial was to provide a working **best subset selection** algorithm based on **PRESS** as selection criterion, which was achieved. For a more detailed comparison with other selection algorithms we recommend to conduct further investigations.

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 = bestPressDf(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 datasets. 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.

References

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
7. Heinze, G. and Dunkler D. (2017). Five myths about variable selection. Transplant International (30), 6-10.
8. Heinze, G., Wallisch, C., and Dunkler D. (2017). Variable selection – A review and recommendations for the practicing statistician. Biometrical Journal (60), 431-449.