Vignette: Weighted BACON algorithms

Tobias Schoch

University of Applied Sciences Northwestern Switzerland FHNW
School of Business, Riggenbachstrasse 16, CH-4600 Olten
`tobias.schoch@fhnw.ch`

March 1, 2021

1 Introduction

The package `wbacon` implements a weighted variant of the BACON algorithms (Billor, Hadi, and Vellemann, 2000) for multivariate outlier detection and robust linear regression. The extension of the BACON algorithm for outlier detection to allow for weighting is due to Béguin and Hulliger (2008).

Additional information on the BACON algorithms and the inplementation can be found in the documents:

- `methods.pdf`: A mathematical description of the algorithms and their implementation;
- `doc_c_functions.pdf`: A documentation of the `C` functions.

Both documents can be found in the package folder `inst/doc/`.

Organization of this document

Section 2 gives instructions how to install and load the package. In Section 3, we illustrate the application of the `wbacon` algorithm for multivariate outlier detection in two case studies (`bushfire` and `philips` data). In Section 4, we study the robust regression estimator `wbacon_reg`.

2 Installation

Make sure that the package `devtools` is installed.[1] Then, the `wbacon` package can be pulled and installed from www.github.com/tobiasschoch/wbacon using

```
> devtools::install_github("tobiasschoch/wbacon")
```

[1] The `devtools` package can be installed from CRAN by `install.packages("devtools")`.

The package contains C code that needs to be compiled. Users of Microsoft Windows need an installation of the R tool chain bundle rtools40 to build the package.

Once the package has been installed, it can be loaded and attached to the current R session by

```
> library(wbacon)
```

3 Multivariate outlier detection

In this section, we study multivariate outlier detection for the two datasets

- bushfire data (with sampling weights),
- philips data (without sampling weights).

3.1 Bushfire data

The bushfire dataset is on satellite remote sensing. These data were used by Campbell (1984)[2] to locate bushfire scars. The data are radiometer readings from polar-orbiting satellites of the National Oceanic and Atmospheric Administration (NOAA) which have been collected continuously since 1981. The measurements are taken on five frequency bands or channels. In the near infrared band, it is possible to distinguish vegetation types from burned surface. At visible wavelengths, the vegetation spectra are similar to burned surface. The spatial resolution is rather low (1.1 km per pixel).

Data preparation

The bushfire data contain radiometer readings for 38 pixels and have been studied in Maronna and Yohai (1995), Béguin and Hulliger (2002), Béguin and Hulliger (2008), and Hulliger and Schoch (2009). The data can be obtained from the R package modi (Hulliger and Sterchi, 2020).[3]

```
> data(bushfire, package = "modi")
```

The first 6 readings on the five frequency bands (variables) are

```
> head(bushfire)
```

```
   X1  X2  X3  X4  X5
1 111 145 188 190 260
2 113 147 187 190 259
3 113 150 195 192 259
4 110 147 211 195 262
5 101 136 240 200 266
6  93 125 262 203 271
```

[2]Campbell, N.A. (1989). Bushfire Mapping using NOAA AVHRR Data. Technical Report. Commonwealth Scientific and Industrial Research Organisation, North Ryde.

[3]The data are also distributed with the R package robustbase (Mächler et al., 2020).

Béguin and Hulliger (2008) generated a set of sampling weights. The weights can be attached to the current session by

```
> data(bushfire.weights, package = "modi")
```

Outlier detection

```
> fit <- wBACON(bushfire, w = bushfire.weights, alpha = 0.95)
> fit
```

```
Weighted BACON: Robust location, covariance, and distances
Initialized by method: V2
Converged in 3 iterations (alpha = 0.95)
```

The argument `alpha` determines the $(1-\alpha)$-quantile $\chi^2_{\alpha,d}$ of the chi-square distribution with d degrees of freedom.[4] All observations whose Mahalanobis distances are smaller than $\chi^2_{\alpha,d}$ are selected into the subset of outlier-free data. It is recommended to choose `alpha` on grounds of an educated guess of the share of "good" observations in the data. Here, we guessed that 95% of the observations are not outliers. In general, the choice of `alpha` does not exert great influence on the result. For instance, the specifications `alpha = 0.95`, `alpha = 0.9`, and `alpha = 0.8` yield the same result.

By default, the initial subset is determined by the Euclidean norm (initialization method: `version = "V2"`). This initialization method is robust because it is based on the coordinate-wise (weighted) median but the resulting estimators of center and scatter are *not affine equivariant*. Let $T(\cdot)$ denote an estimator of a parameter of interest (e.g., covariance matrix) and let $\boldsymbol{X}$ denote the $(n \times p)$ data matrix. An estimator T is affine equivariant if and only if

$$T(\boldsymbol{AX} + \boldsymbol{b}) = \boldsymbol{A}T(\boldsymbol{X}) + \boldsymbol{b},$$

for any nonsingular $(m \times n)$ matrix $\boldsymbol{A}$ and any n-vector $\boldsymbol{b}$. Although version `"V2"` of the BACON method leads to estimators that are not affine equivariant in the above sense, Billor et al. (2000) point out that the method is nearly affine equivariant. There exists an alternative initialization method (`"version = V1"`) which is based on the coordinate-wise (weighted) means; therefore, it is affine equivariant but *not robust*.

From the above output, we see that the algorithm converged in three iterations. In case the algorithm does not converge, we may increase the maximum number of iterations (default: `maxiter = 50`) and toggle `verbose = TRUE` to (hopefully) learn more why the method did not converge.

In the next step, we want to study the result in more detail. In particular, we are interested in the estimated center and scatter (or covariance) matrix. To this end, we can call the `summary()` method on the object `fit`.

[4]The degrees of freedom d is a function of the number of variables p, the number of observations n, and the size of the current subset m; see `methods.pdf` in the `inst/doc` folder of the package.

```
> summary(fit)

Weighted BACON: Robust location, covariance, and distances
Initialized by method: V2
Converged in 3 iterations (alpha = 0.95)

Number of detected outliers: 24 (63.16%)

Robust estimate of location:
   X1    X2    X3    X4    X5
108.0 148.9 274.8 218.2 279.4

Robust estimate of covariance:
        X1     X2      X3     X4     X5
X1   391.3  303.5 -1410.5 -284.5 -240.1
X2   303.5  262.4  -935.3 -166.5 -147.6
X3 -1410.5 -935.3  7343.3 1765.2 1413.1
X4  -284.5 -166.5  1765.2  467.7  365.0
X5  -240.1 -147.6  1413.1  365.0  287.7

Distances (cutoff: 20.79):
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.348   1.958   2.744   7.827  12.979  23.128
```

The method detected 24 outliers. The method `is_outlier()` returns a vector of logicals whether an observation has been flagged as an outlier.

```
> which(is_outlier(fit))

 [1]  7  8  9 10 11 12 31 32 33 34 35 36 37 38
```

The center and covariance (scatter) matrix can be extracted with the auxiliary functions, respectively, `center()` and `cov()`.

```
> center(fit)

      X1       X2       X3       X4       X5
108.0156 148.8594 274.8438 218.2500 279.4219
```

The robust Mahalanobis distances, whose summary statistic is printed by the `summary()` method, can be extracted with the `distance()` method.

An application of this function is the following code snipped

```
> hist(distance(fit), breaks = 20)
> abline(v = fit$cutoff, lty = 2)
```

the resulting graph is shown in Figure 1. The vertical dotted line shows the cutoff threshold that has been used by `wbacon()` for outlier detection/ nomination.

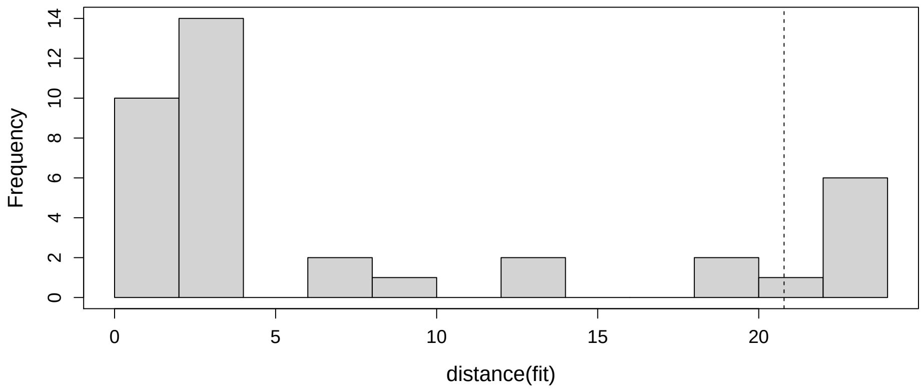

Figure 1: Histogram of distances from the center (bushfire data)

3.2 Philips data

Old television sets had a cathode ray tube with an electron gun. The emitted beam runs through a diaphragm that lets pass only a partial beam to the screen. The diaphragm consists of 9 components. The Philips data set contains $n = 667$ measurements on the $p = 9$ components (variables); see Rousseeuw and van Driessen (1999).

Data preparation

The `philips` data can be loaded from the `R` package `cellWise` (Raymaekers and Rousseeuw, 2020). These data do not have sampling weights.

```
> data(philips, package = "cellWise")
> head(philips)
```

```
        X1     X2    X3    X4    X5    X6     X7     X8     X9
[1,] 0.153 -0.259 0.140 0.514 2.242 0.443 -0.021 -0.035 -0.065
[2,] 0.119 -0.309 0.132 0.518 2.269 0.458 -0.018 -0.035 -0.053
[3,] 0.173 -0.296 0.138 0.516 2.266 0.461 -0.023 -0.026 -0.052
[4,] 0.135 -0.306 0.139 0.522 2.288 0.464 -0.015 -0.031 -0.051
[5,] 0.143 -0.278 0.139 0.519 2.284 0.465 -0.016 -0.018 -0.054
[6,] 0.140 -0.284 0.159 0.531 2.287 0.465 -0.004 -0.024 -0.052
```

Outlier detection

We compute the BACON algorithm but this time with the initialization method `version = "V1"`.

```
> fit <- wBACON(philips, alpha = 0.99, version = "V1")
> fit
```

```
Weighted BACON: Robust location, covariance, and distances
Initialized by method: V1
Converged in 9 iterations (alpha = 0.99)
```

The center of the data is estimated to be

```
> print(center(fit), digits = 2)
```

```
    X1     X2     X3     X4     X5     X6     X7     X8     X9
-0.041 -0.316 -0.051  0.438  2.122  0.434 -0.104 -0.067 -0.089
```

and the BACON algorithm detected

```
> sum(is_outlier(fit))
```

```
[1] 132
```

outliers.

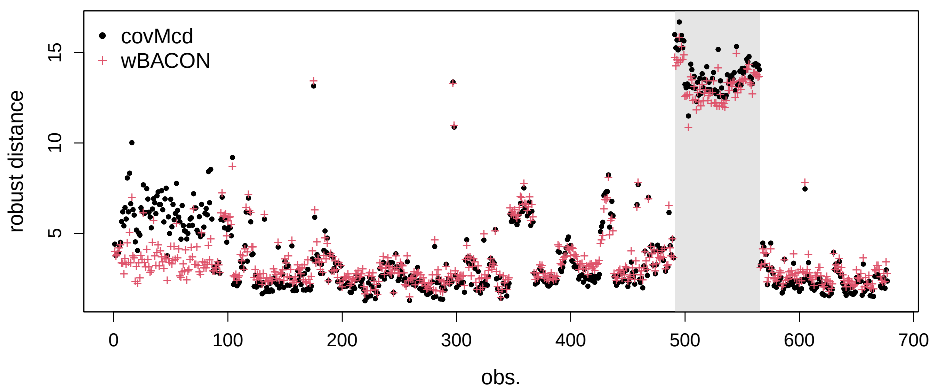

Figure 2: Robust Mahalanobis distances of the BACON algorithm and the fast MCD (philips data)

Comparison with MCD

It is instructive to compare the detected outlier patterns of the BACON method with the patterns detected by the fast minimum covariance determinant (fast MCD) of Rousseeuw and van Driessen (1999). The fast MCD is implemented as function `covMcd` in the `R` package `robustbase` of Mächler et al. (2020). In terms of computational costs, fast MCD is much more expensive than the BACON algorithm.

```
> library(robustbase)
> fit_mcd <- covMcd(philips)
```

The robust Mahalanobis distances of the BACON algorithm and the fast MCD are shown in Figure 2. The outlier patterns of the two methods are very similar. In particular, the BACON algorithm detects the strongly deviating group of observations with no. 491–565 (highlighted by the gray background); see Rousseeuw and van Driessen (1999) for a discussion of these observations. The computed distances of the first 100 observations are slightly different for the two detection methods.

4 Robust linear regression

The `education` data is on education expenditures in 50 US states in 1975 (Chatterjee and Hadi, 2012, Chap. 5.7). The data can be loaded from the `robustbase` package.

```
> data(education, package = "robustbase")
```

It is convenient to rename the variables.

```
> names(education)[3:6] <- c("RES", "INC", "YOUNG", "EXP")
> head(education)
```

```
  State Region RES  INC YOUNG EXP
1    ME      1 508 3944   325 235
2    NH      1 564 4578   323 231
3    VT      1 322 4011   328 270
4    MA      1 846 5233   305 261
5    RI      1 871 4780   303 300
6    CT      1 774 5889   307 317
```

The measured variables for the 50 states are:

`State` State

`Region` group variable with outcomes: 1=Northeastern, 2=North central, 3=Southern, and 4=Western

`RES`: Number of residents per thousand residing in urban areas in 1970

`INC`: Per capita personal income in 1973 ($US)

`YOUNG`: Number of residents per thousand under 18 years of age in 1974

`EXP`: Per capita expenditure on public education in a state ($US), projected for 1975

Model fit

We want to regress education expenditures (`EXP`) on the variables `RES`, `INC`, and `YOUNG` by the BACON algorithm, and obtain

```
> reg <- wBACON_reg(EXP ~ RES + INC + YOUNG, data = education)
> reg

Call:
wBACON_reg(formula = EXP ~ RES + INC + YOUNG, data = education)

Regression on the subset of 49 out of 50 observations (98%)

Coefficients:
(Intercept)          RES          INC        YOUNG
 -277.57731      0.06679      0.04829      0.88693
```

The instance `reg` is an object of the class `roblm`. The printed output of `wBACON_reg` is identical with the one of the `lm` function. In addition, we are told the size of the subset on which the regression has been computed. The observations not in the subset are considered outliers (here 3 out of 50 observations, i.e. 6%).

The `summary()` method can be used to obtain a summary of the estimated model.

```
> summary(reg)

Call:
wBACON_reg(formula = EXP ~ RES + INC + YOUNG, data = education)

Residuals:
    Min      1Q  Median      3Q     Max
-81.128 -22.154  -7.542  22.542  80.890

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -277.57731  132.42286  -2.096 0.041724 *
RES            0.06679    0.04934   1.354 0.182591
INC            0.04829    0.01215   3.976 0.000252 ***
YOUNG          0.88693    0.33114   2.678 0.010291 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 35.81 on 45 degrees of freedom
Multiple R-squared:  0.4967,	Adjusted R-squared:  0.4631
F-statistic:  14.8 on 3 and 45 DF,  p-value: 7.653e-07
```

The summary output of `wBACON_reg` is identical with the output of the `lm` estimate on the subset of outlier-free data,

```
> summary(lm(EXP ~ RES + INC + YOUNG, data = education[!is_outlier(reg), ]))
```

where we have used `is_outlier()` to extract the set of declared outliers from `reg` (the summary output of the `lm` estimate is not shown).

Tuning

By default, `wBACON_reg` uses the parametrization $\alpha = 0.95$, `collect = 4`, and `version = "V2"`. These parameters are used to call the `wBACON` algorithm on the design matrix. Then, the same parameters are used to compute the robust regression.

To ensure a high breakdown point, `version = "V2"` should not be changed to `version = "V1"` unless you have good reasons. The main "turning knob" to tune the algorithm is `alpha`, which defines the (1−`alpha`) quantile of the Student t-distribution. All observations whose distances/discrepancies[5] are smaller (in absolute value) than the quantile are selected into the subset of "good" data. By choosing smaller values for `alpha` (e.g., 0.7), more observations are selected (ceteris paribus) into the subset of "good" data (and vice versa).

The parameter `collect` specifies the initial subset size, which is defined as $m = p \cdot collect$. It can be modified but should be chosen such that m is considerably smaller than the number of observations n. Otherwise there is a high risk of selecting too many "bad" observations into the initial subset, which will eventually bias the regression estimates.

In case the algorithm does not converge, we may increase the maximum number of iterations (default: `maxiter = 50`) and toggle `verbose = TRUE` to (hopefully) learn more why the method did not converge.

Model diagnostics

The methods `coef()`, `vcov()`, and `predict()` work exactly the same as their `lm` counterparts. This is also true for the first three `plot` types (`which %in% 1:3`), that is

1: Residuals vs Fitted,

2: Normal Q-Q,

3: Scale-Location

The plot types `4:6` of `plot.lm` are not implemented for objects of the class `roblm` because it is not sensible to study the standard regression influence diagnostics in the presence of outliers in the model's design space. Instead, type four (`which = 4`) plots the robust Mahalanobis distances with respect to the non-constant design variables against the standardized residual. This plot has been proposed by Rousseeuw and van Zomeren (1990).

Figure 3 shows `plot(reg, which = 4)`. The filled circles represent the outliers detected by the BACON algorithm. The two outlying observations with robust Mahalanobis distances (see abscissae) slightly below 1.0 are flagged as outliers because their standardized residual falls outside the interval spanned by $\pm t_{\alpha/(2m+2),m-p}$, where $t_{\alpha,m-p}$ is the $(1-\alpha)$ quantile of the Student t-distribution with $m-p$ degrees of freedom, m denoting the size of the final subset of outlier-free data. Here, we have $m = 47$, $\alpha = 0.95$ (see argument `alpha` of `wBACON_reg`), thus the interval is $[-2.42,\ 2.42]$. The outlier in the top right corner of Figure 3 is both a residual outlier and an outlier in the model's design space.

[5] See document `methods.pdf` in the folder `/inst/doc` of the package.

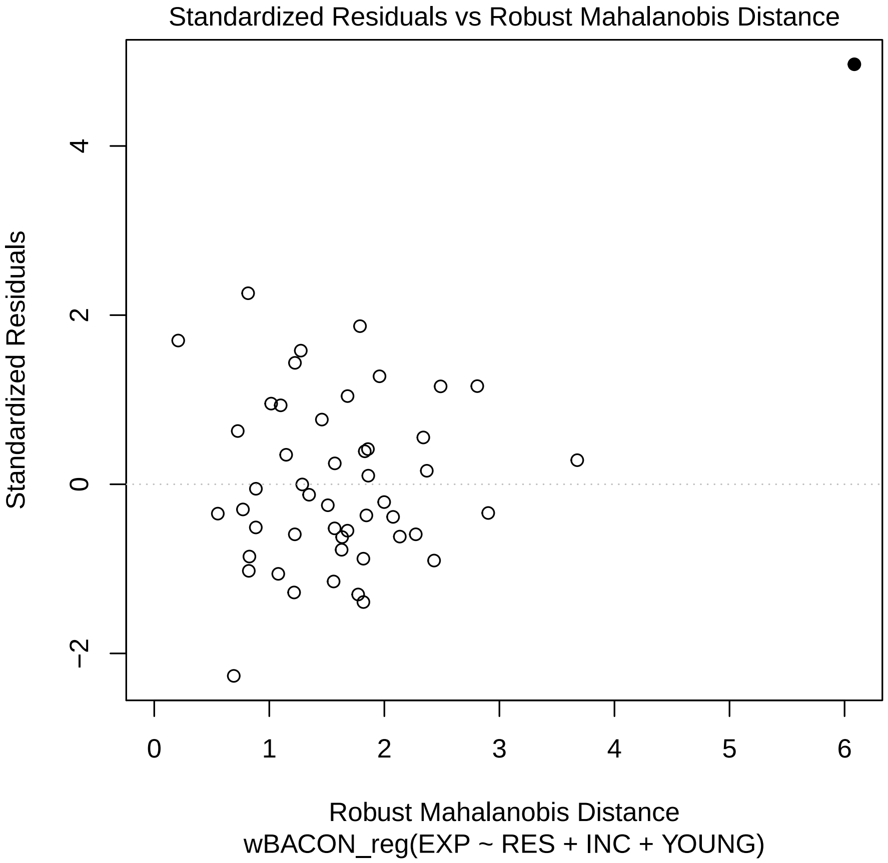

Figure 3: A

Note

For small samples, exclusion of outliers instead of downweighting efficiency outliers in x and y

References

Béguin, C. and B. Hulliger (2002): *Robust Multivariate Outlier Detection and Imputation with Incomplete Survey Data*, Deliverable D4/5.2.1/2 Part C: EUREDIT project, https://www.cs.york.ac.uk/euredit/euredit-main.html, research project funded by the European Commission, IST-1999-10226.

Béguin, C. and B. Hulliger (2008): "The BACON-EEM Algorithm for Multivariate Outlier Detection in Incomplete Survey Data," *Survey Methodology*, Vol. 34, No. 1, 91–103.

Billor, N., A. S. Hadi, and P. F. Vellemann (2000): "BACON: Blocked Adaptive Computationally-efficient Outlier Nominators," *Computational Statistics and Data Analysis*, 34, 279–298.

Chatterjee, S. and A. H. Hadi (2012): *Regression Analysis by Example*, 5th ed., Hoboken (NJ): John Wiley & Sons.

Hulliger, B. and T. Schoch (2009): "Robust multivariate imputation with survey data," in *Proceedings of the 57th Session of the International Statistical Institute*, Durban.

Hulliger, B. and M. Sterchi (2020): *modi: Multivariate Outlier Detection and Imputation for Incomplete Survey Data*, R package version 0.1-0.

Mächler, M., P. Rousseeuw, C. Croux, V. Todorov, A. Ruckstuhl, M. Salibian-Barrera, T. Verbeke, M. Koller, E. L. T. Conceicao, and M. Anna di Palma (2020): *robustbase: Basic Robust Statistics*, R package version 0.93-6.

Maronna, R. A. and V. J. Yohai (1995): "The Behavior of the Stahel-Donoho Robust Multivariate Estimator," *Journal of the American Statistical Association*, 90, 330–341.

Raymaekers, J. and P. Rousseeuw (2020): *cellWise: Analyzing Data with Cellwise Outliers*, R package version 2.2.1.

Rousseeuw, P. J. and K. van Driessen (1999): "A fast algorithm for the Minimum Covariance Determinant estimator", *Technometrics*, 41, 212–223.

Rousseeuw, P. J. and K. van Zomeren (1990): "Unmasking Multivariate Outliers and Leverage Points", *Journal of the American Statistical Association*, 411, 633–639.