Revista Colombiana de Estadística
ISSN: 0120-1751
revcoles_fcbog@unal.edu.co
Universidad Nacional de Colombia
Colombia

Kan-Kilinç, Betül; Alpu, Ozlem
Combining Some Biased Estimation Methods with Least Trimmed Squares Regression and its Application
Revista Colombiana de Estadística, vol. 38, núm. 2, julio, 2015, pp. 485-502
Universidad Nacional de Colombia
Bogotá, Colombia

Revista Colombiana de Estadística
July 2015, Volume 38, Issue 2, pp. 485 to 502
DOI: http://dx.doi.org/10.15446/rce.v38n2.51675

Combining Some Biased Estimation Methods with Least Trimmed Squares Regression and its Application

Combinación de algunos métodos de estimación sesgados con regresión de mínimos cuadrados recortados y su aplicación

Betül Kan-Kılınç[1,a], Ozlem Alpu[2,b]

[1]Department of Statistics, Science Faculty, Anadolu University, Eskisehir, Turkey
[2]Department of Statistics, Faculty of Arts and Sciences, Eskisehir Osmangazi University, Eskisehir, Turkey

Abstract

In the case of multicollinearity and outliers in regression analysis, the researchers are encouraged to deal with two problems simultaneously. Biased methods based on robust estimators are useful for estimating the regression coefficients for such cases. In this study we examine some robust biased estimators on the datasets with outliers in x direction and outliers in both x and y direction from literature by means of the R package `ltsbase`. Instead of a complete data analysis, robust biased estimators are evaluated using capabilities and features of this package.

***Key words*:** Biased Estimator, Least Trimmed Squares, Robust Estimation.

Resumen

En el caso de multicolinealidad y outliers en análisis de regresión, los investigadores se enfrentan a tener que tratar dos problemas de manera simultánea. Métodos sesgados basados en estimadores robustos son útiles para estimar los coeficientes de regresión en estos casos. En este estudio se examinan algunos estimadores sesgados robustos en conjuntos de datos con outliers en x y outliers tanto en x como en y por medio del paquete ltsbase de R. En lugar de un análisis de datos completos, los estimadores sesgados robustos son evaluados usando las capacidades y características de este paquete.

***Palabras clave*:** estimadores sesgados, mínimos cuadrados recortados, robusta estimación.

[a]Professor. E-mail: bkan@anadolu.edu.tr
[b]Professor. E-mail: oalpu@ogu.edu.tr

1. The Least Trimmed Squares

Least Trimmed Squares (LTS) or Least Trimmed Sum of Squares is one of a number of methods for robust regression (Rousseeuw & Leroy 1987). There exists several algorithms for calculating the LTS estimates in the literature: Ruppert & Carrol (1980), Neykov & Neytchev (1991), Tichavsky (1991), Atkinson & Weisberg (1991), Ruppert (1992), Stromberg (1993), Hawkins (1994), Hossjer (1995), Rousseeuw & van Driessen (1999), Agullo (2001), Hawkins & Olive (2002), Willems & van Aelst (2005), Jung (2005), Li (2005), Cizek (2005), Rousseeuw & van Driessen (2006).

Peter Rousseeuw introduced several robust estimators including LTS in his works. LTS is a statistical robust technique for fitting a linear regression model to a set of n points given a trimming parameter h as it is insensitive due to outliers $(n/2 \leq h \leq n)$. More formally, LTS estimator is defined on an objective function which is minimized by

$$\min_{\hat{\beta}} \sum_{i=1}^{h} (e^2)_{i:n} \tag{1}$$

where $(e^2)_{i:n}$ is the i^{th} smallest residual or distance when the residuals are ordered in ascending order. As h is the number of good data points, LTS estimator obtaines a robust estimate by trimming the $(n-h)$ data points having the largest residuals from the data set. Note that, when $h = n$, it is equivalent to the ordinary least squares estimator. It is also possible to take h close to the number of good points as the more accurate estimates are rational to the number of good points. For small sample sizes the existing algorithms are fine, however the computation time increases with the larger size of data set. Hence other possible ways for fitting are considered. Rousseeuw & van Driessen (1999) proposed a fast algorithm based on a random sampling for computing LTS which was finally published as Rousseeuw & van Driessen (2006). In this study, only the FAST-LTS algorithm proposed by Rousseeuw and van Driessen will be considered.

The paper unfolds as follows: Section 2 outlines the contributions to LTS in the presence of multicollinearity. Section 3 explains some robust biased estimators. The next section introduces the `ltsbase` package and gives statistical analysis of the example datasets in subsections. Finally, the last section presents the remarkable difference between the `ltsbase` and previous algorithms in `R`.

2. Contributions to LTS in the Presence of Multicollinearity

Multicollinearity is a common problem in many areas, i.e., economical, technical and medical applications. This problem has been examined in literature from different points of view like estimation and testing the hypothesis of parameters, removal and diagnostic tools. Several diagnostic tools such as condition number, condition indices, variance inflation factors, singular value decomposition, etc. have been suggested and used for detection of multicollinearity Belsley (1991), Heumann, Shalabh, Rao & Toutenburg (2008), Wissmann, Toutenburg & Shalabh (2007). In this study, we focus exclusively on the Variance Inflation Factor for $\hat{\beta}_i$ with the following form $VIF = 1/(1 - R_i^2)$ and Condition Number, $\lambda_{\max}/\lambda_{\min}$, in order to diagnose the multicollinearity. Here, R_i^2 is the coefficient of determination and $\lambda_{\max}$, $\lambda_{\min}$ refer to the maximum and minimum eigenvalues of the corresponding matrix, respectively.

When multicollinear datasets have also outliers, researchers are forced to deal with those problems simultaneously. For this purpose, Kan, Alpu & Yazici (2013) studied the effectiveness of some robust biased estimators via a simulation study for different types of outliers. Also they provided a dataset with outliers in y direction to show the performance of biased estimators based on LTS.

In this paper, Kan Kilinc B. and Alpu O. (2013) introduce a new package `ltsbase`, implemented in the `R` System for statistical computing and available on `http:/CRAN.r.project.org/package=ltsbase`. It can be used to perform a biased estimation based on a robust method (Kan Kilinc B. and Alpu O. 2013).

Differently from Kan et al. (2013), we expand on some robust biased estimators for the datasets with outliers in x direction and outliers both in x and y direction by means of the `ltsbase` package. Hence this study will help close the considerable gap in the estimation of the Ridge and Liu parameters in the presence of multicolinearity and outliers by using the LTS method.

3. Robust Biased Estimators

In standard linear regression, consider the model

$$y_i = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi} + \epsilon_i, i = 1, \ldots, n \tag{2}$$

where $\beta{=}(\beta_0, \beta_1, \ldots, \beta_p)'$ is the unknown parameter vector, $\mathbf{X}_{(n\times(p+1))}$ is a fixed matrix of full rank of observations and ϵ_i are iid random variables with mean 0 and variance $\sigma^2 I_n$. The estimation of the regression coefficients, $\hat{\beta}$, is generally obtained by Ordinary Least Squares (OLS) method. However, large numbers of regressors in multiple linear regression analysis can cause serious problems in estimation and prediction.

A serious ill conditioned problem is characterized by the fact that the smallest eigenvalue of the $\mathbf{X}^{'}\mathbf{X}$ is much smaller than unity. In other words, the matrix $\mathbf{X}^{'}\mathbf{X}$ has a determinant which is close to zero, which makes it ill conditioned so that the matrix can not be inverted. Here, the least squares solution is still unbiased but is plagued by a large variance. Hence thr OLS solution yields a vector $\hat{\beta}$ coefficients which are too large in absolute value (Marquardt & Snee 1975).

For any design matrix $\mathbf{X}$, the quantity $\mathbf{X}^{'}\mathbf{X} + k\mathbf{I}$ is always invertible where $\mathbf{I}$ is a $(p+1) \times (p+1)$ identity matrix. Thus, Hoerl & Kennard (1970) suggested a ridge regression estimator, $\hat{\beta}_{Ridge} = (\mathbf{X}^{'}\mathbf{X}{+}k\mathbf{I})^{\mathbf{-1}}\mathbf{X}^{'}\mathbf{y}$ where $k \geq 0$, and Liu (1993) proposed another biased estimator, $\hat{\beta}_{Liu} = (\mathbf{X}^{'}\mathbf{X} + \mathbf{I})^{\mathbf{-1}}(\mathbf{X}^{'}\mathbf{X} + d\mathbf{I})\hat{\beta}$, $0 < d < 1$.

The Ridge and Liu regressions penalize the size of the regression coefficients. Here, both k and d are tuning (biasing) parameters which control the strength of the penalty term.

In our study, the biasing parameters *klts*, *dlts* and the MSE values of two robust biased estimators $\hat{\beta}_{ltsRidge}, \hat{\beta}_{ltsLiu}$ are examined when outliers and multicollinearity exist in the dataset. *klts* and *dlts* are considered as the robust choice is of the biasing parameters k and d. In application we have different robust biased estimations since the robust biasing parameters change by the user increment. Thus, we might choose the biasing parameters *klts* and *dlts* which minimize MSE value. To illustrate the performance of the robust biased estimators, the MSE criterion is used. Here, MSE($\hat{\beta}_\bullet$)=trCov($\hat{\beta}_\bullet$)+bias($\hat{\beta}_\bullet$)$'$bias($\hat{\beta}_\bullet$) where *tr* denotes the trace and $\hat{\beta}_\bullet$ present is the robust biased estimators (Kan et al. 2013).

4. The ltsbase Package: Features and Functions

The `R` System has many packages and functions- e.g., `MASS:lqs()` (Venables & Ripley 2002), `robustbase:ltsReg()` (Rousseeuw & van Driessen 1999), and `sparseLTSEigen:RcppEigen()` (Alfons, A. 2013), to perform least trimmed squares regression and related statistical methods. The `ltsbase` package has a number of features not available in current R packages and fills the existing gap in the `R` statistical environment which is the convenient comparison for biased estimations based on the LTS method.

The `ltsbase` package includes centering, scaling, singular value decomposition (svd) and the least trimmed squares method. Hence centering or scaling the data is not required by the user. On the other hand, when computing $\hat{\beta}_{Ridge}$ numerically, the matrix inversion is avoided because of inverting $\mathbf{X}'\mathbf{X}$ can be computationally expensive. Rather, the svd is utilized. So that, the regression coefficients of each model are estimated. The package `ltsbase` has three functions to serve three purposes. First, it is the minimum MSE (Mean Squared Error) value which is extracted by calling `ltsbase()` function. Then the fitted values and the residuals of the corresponding model might be extracted as well. To return these values, one should use the `ltsbaseDefault()` function. Finally, the biasing parameters and regression coefficients for the corresponding model at minimum MSE value might be extracted by using `ltsbaseSummary()` function. Furthermore, the `ltsbase` package was designed especially to create "comparison of MSE" graphics based on the methods used in the analysis. Hence it allows users to see visual output without creating each graphic individually.

The `ltsbase()` function is the main function of the `ltsbase` package. This function computes the minimum MSE values for six methods: OLS, Ridge, Ridge based on LTS, LTS, Liu, and Liu based on LTS for sequences of biasing parameters. It returns a comprehensive output presenting the biasing parameters and the coefficients for the models at minimum MSE value. Basically, the following code line executes the main function:

```
R>ltsbase(xdata,y,print=FALSE,plot=FALSE,alpha=0.50,by=0.001)
```

Here, `xdata` is a data frame including regressors and `y` is a response variable. The values of MSE and the comparison of MSE values of the four methods (Ridge, Ridge based on LTS, Liu, Liu based on LTS) in lines (with different colours and line type) on a plot obtained by setting `plot` and `print` parameters `TRUE`. The `alpha` in the function is the percentage (roughly) of squared residuals whose sum will be minimized by the LTS regression method. It requires a value between 0.5 and 1. The last argument `by` is a number giving the increment of the sequence where the biasing parameters are defined.

In the following two sections the usage of `ltsbase` package is illustrated by two examples presenting two different cases of outliers .

4.1. Case Study 1: Outliers in x Direction

An artificial dataset `hbk` involving outliers with 75 observations for three regressors x_1, x_2, x_3 and one response variable y was created by Hawkins, Bradu & Kass (1984), the raw data (hereafter refered to as the *hbk* data) being found in Appendix A.1. Since *hbk* is a well-known data set, the analysis of variance and parameter estimates of OLS will not be shown here. However, some diagnostic measures for the OLS analysis may be found in Appendix A.2. Of particular interest is the placement of leverage points among the remaining data points. Mason & Gunst (1985) showed that collinearity can be increased without bound by increasing the leverage of a point (Mason & Gunst 1985). They also showed that a q-variate leverage point can produce $q-1$ independent collinearities (Chatterjee & Hadi 2006). A closer look at the diagnostics of points are given in Figure 1.

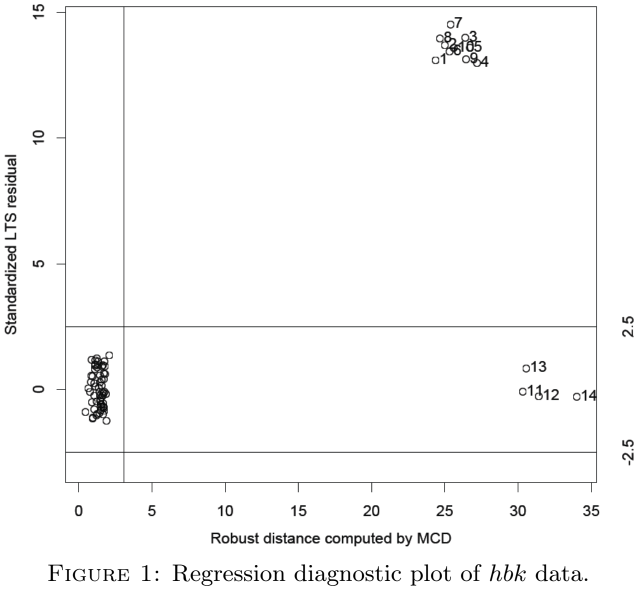

Figure 1: Regression diagnostic plot of *hbk* data.

In Figure 1, multiple high leverage points which may cause the multicollinearity are observed. The figure identifies all 14 leverage points. The four good leverage points of them have small standardized LTS residuals but a large robust distance, and the 10 bad leverage points $(1, 2, \ldots, 10$ numbered) have large standardized LTS residuals and a large robust distance (see Appendix A.3).

4.1.1. ltsbase Function

Let y denote the vector of response values and *xdata* the regressors. Also regressors are assumed to be given in a data frame (not in a matrix or in an array). To fit Ridge and Liu regression models based on LTS, we call the `ltsbase` function.

```
R> model1=ltsbase(xdata,y,print=FALSE,plot=TRUE,alpha=0.875,by=0.001)
```

Here, when `print` is TRUE the user can call all the values calculated in the analysis. Also, when `plot` is TRUE, the function produces the lines of all MSE values versus biasing parameters. The `alpha` is the percentage (roughly) of squared residuals whose sum will be minimized by 0.875 and `by` is the increment of the sequence, by default 0.001. The LTS regression method minimizes the sum of the h smallest squared residuals, where $h > n/2$, i.e. at least half the number of observations must be used. The default value of h (when alpha=1/2) is roughly $n/2$, where n is the total number of observations, but by setting alpha, the user may choose higher values up to n.

As reported in the previous section, *hbk* data is used to highlight the specific features of `ltsbase` and how to interpret the results. The aim of this analysis is to find the MSE value among some methods such as OLS, Ridge, Ridge based on LTS, LTS, Liu and Liu based on LTS. After running the code, the outputs are given in the following:

```
R> model1
$list.mse
      OLS    Ridge   LTS.Ridge    LTS       Liu    LTS.Liu
1 0.3911056  0.345    0.068   0.1659851 0.3324078   0.067
$list.bias.par
  ridge.k  lts.k  liu.d  lts.liu.d
1   0.003  0.008  0.845   0.673
$list.coef.all
      OLS       LTS      Ridge     Liu   LTS.Ridge   LTS.Liu
X1          0.2501        0.1634  -0.4355  -0.4187   -0.6774   -0.4413
X2         -0.7892    0.2507   0.3509   0.3152   -0.1558   -0.0934
X3          1.2885         0.7591         1.2268   1.2048          0.2924    0.3010
```

The returned output contains three elements: (1) the smallest MSE values obtained by each method, (2) biasing parameters differ in sequence of [0,1], and (3) the coefficients of the corresponding regression model at minimum MSE.

Here, the minimum MSE value is obtained as 0.067 by Liu based on the LTS method. The corresponding biasing parameter *dlts* at the minimum MSE value is as 0.673. Hence, the coefficient vector of the regression model is estimated as $\hat{\beta}^{\star} = (-0.4413, -0.0933, 0.3010)'$.

Furthermore, `ltsbase` produces the MSE values for Ridge, Ridge based on LTS, Liu and Liu based on LTS methods against the different biasing parameters `(k,klts,d,dlts)=seq(0,1,0.001)` when `print=TRUE` and plots a graph when `plot=TRUE`. (See Figure 2).

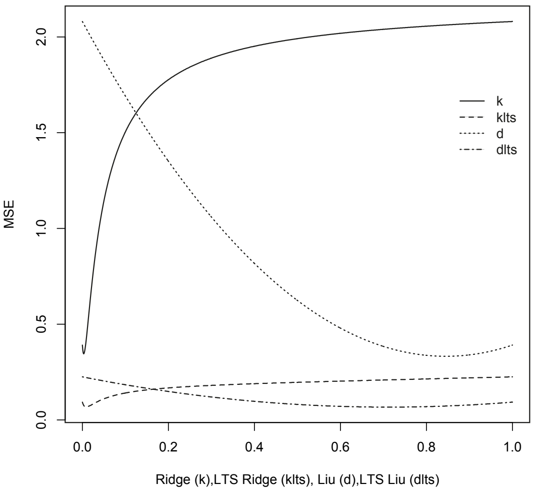

Figure 2: Biasing parameters versus MSE values for four methods for outliers in x direction.

The colors and line types of curves represent the values of the biasing parameters versus MSE values. For instance, the black-line curve is obtained by ridge regression and the blue-dotted curve is from Liu estimation (See top right legend of the Figure). As the `plot` argument in `ltsbase` function supports a layout of MSE values versus biasing parameters for four methods, one can easily provide the immediate visual information about the MSE values. Note that each line is also plotted in different types for print color as gray.

As can be readily seen in Figure 2, the model is identical to the OLS regression model at `(k,klts,d,dlts)=(0,0,0,0)`. The aim of the plotting is actually an exploratory tool to show the sensitivity of the MSE values to the methods being used here. On the figure, each method is traced along its biasing parameter scale beginning at 0 and ending at 1. As k increases, the MSE values assosciated with Ridge regression are increasing and then almost horizontal after a certain point of k. The same pattern is followed by Ridge regression based on LTS. However, the MSE values obtained by the LTS method are much smaller than those obtained by Ridge regression as the biasing parameter $klts$ increases. On the other hand, following the blue-dotted curve which is produced by the Liu estimation, the MSE values rises at low levels of d and falls steeply as the biasing parameter d increases. Observing the MSE values of Liu based on the LTS method as $dlts$ increases, note how the MSE value decreases slightly and then levels out.

4.1.2. ltsbaseSummary Function

A summary of the analysis produced by the `ltsbaseSummary` function showing the biasing parameter at minimum MSE values. The following code runs the summary of the biased LTS method.

```
R> ltsbaseSummary(model1)
best mse at  lts.liu.d
1   0.67241
corresponding coefficients
[1] -0.44124229 -0.09328311  0.30078153
best mse
[1] 0.067
```

Here we have three results: (1) the best biasing parameter which gives the minimum MSE among the others, (2) the regression coefficients of the corresponding regression model at the best biasing parameter, (3) the minimum MSE value. It is also possible to see in Figure 2 that the MSE value begins to stabilize at around $dlts = 0.65$ and shows a slight downward trend at $dlts = 0.67$ which is the minimum among the other methods. It also extracts the coefficients of the corresponding model.

4.1.3. ltsbaseDefault Function

The fitted values and residuals of the corresponding model are also extracted as one of the returned outputs by `ltsbase` package (see Appendix A.4).

As seen, there are substantial differences among available packages related to LTS in R and the `ltsbase` is currently the only one to offer together: (1) lists of MSE values, biasing parameters and model coefficients, (2) MSE values versus biasing parameters (available if plot is set to TRUE), (3) fitted values and residuals.

4.2. Case Study 2: Outliers in Both x and y Direction

Maguna, Nunez, Okulik & Castro (2003) examined the toxicity of carboxylic acids on the basis of several molecular descriptors in their research. They reported the results of a QSPR study and obtained quite reasonable estimates compared to the previous theoretical calculations. The aim of their experiment was to predict the toxicity of carboxylic acids on the basis of several molecular descriptors.

One of the concerns is how well our method performs when the data have outliers in both directions. We explore this on a data frame with 38 observations on the 10 variables used in application and the description of the data set is given in Table 1. In the table, the toxicity is defined as the response variable and the remaining variables are considered as regressors.

In Figure 3, the placements of outliers are presented and the points are identified by numbers. It is seen that while the observations 23, 28, 32, 34, 35, 36, and 37 are identified as outliers in the x direction, the observation $11, 12$, and 13 are identified as outliers in the y direction. The remaining data are all well-behaved or good leverage points.

Table 1: Description of the toxicity data.

Abrreviations	Variables
toxicity	aquatic toxicity, defined as $\log(IGC50^{(-1)})$; typically the response
logKow	the partition coefficient
pKa	the dissociation constant
ELUMO	energy of the lowest unoccupied molecular orbital
Ecarb	electrotopological state of the carboxylic group
Emet	electrotopological state of the methyl group
RM	molar refractivity
IR	refraction index
Ts	surface tension
P	polarizability

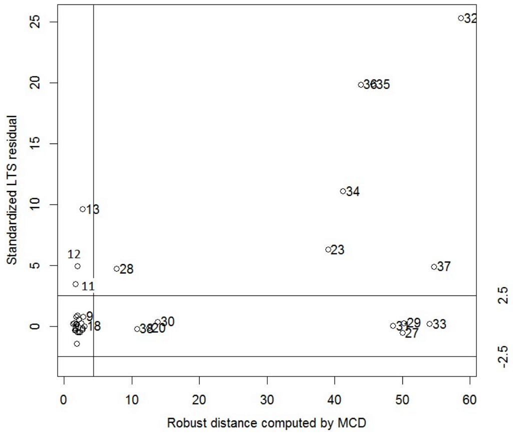

Figure 3: Regression diagnostic plot of toxicity data.

Secondly, we use on the data to determine whether there is multicollinearity among regressors or not. The procedure has been used for *hbk* data and is repeated for toxicity data in terms of multicollinearity and outliers. To detect multicollinearity for toxicity data, the same measures given in Appendix A.2 are used and interpreted in Appendix B.2. Considering all indicators together, there is severe multicollinearity, therefore it can be said that this is fairly effective on the results.

Due to the presence of multicollinearity and outliers in the toxicity data, neither `MASS::lqs` nor `robustbase::ltsReg` in `R` are suitable to cope with those problems. Currently, the `ltsbase` package deals with both multicollinearity and outliers simultaneously and offers a wide array of features including a graphical comparison for the analysis.

4.2.1. ltsbase Function

This subsection provides illustrations of code `ltsbase` for toxicity data and returns the following components of the biased estimation based on the LTS method:

```
R > model2=ltsbase(xdata,y,plot=TRUE)
$list.mse
      OLS      Ridge  LTS.Ridge     LTS       Liu     LTS.Liu
1 0.8828511   0.561   0.415    0.8883284  0.5370347   0.398
$list.bias.par
  ridge.k  lts.k   liu.d  lts.liu.d
1   0.011  0.005   0.593   0.712
$list.coef.all
            OLS     LTS    Ridge    Liu   LTS.Ridge  LTS.Liu
logKow  1.0470  0.5141 -0.2834 -0.2624   -0.2613 -0.2657
pKa     0.0657  0.1348  0.1849  0.1640    0.0851  0.1498
ELUMO  -0.4179 -0.3367 -0.5446 -0.4417   -0.4331 -0.4336
Ecarb  -0.0449  0.1431  0.2965  0.2242    0.3296  0.2605
Emet    0.0954  0.6359  0.1456  0.1030   -0.0194  0.0221
RM     -0.4417 -0.8100  0.1852  0.1283    0.0191  0.0421
IR      0.3364  0.3499  0.0429  0.0334    0.6364  0.4978
Ts     -0.3351 -0.3873 -0.5398 -0.4415   -0.2238 -0.2014
P       0.1353  0.1457 -0.4058 -0.4706   -0.6176 -0.6392
```

The first component returns the smallest MSE values which are estimated for all methods among the sequence of interval [0,1] of biasing parameters. It can be seen that the smallest MSE values are obtained by biased estimations based on LTS. Next component presents the list of the biasing parameters obtained by each method. Finally, the list of the regression coefficients for the corresponding model are given in a data frame.

In Figure 4, MSE values versus different biasing parameters for four methods obtained by `ltsbase` are presented when there are outliers in both x and y directions. In the figure, it is possible to see approximately at which method the MSE value is at its smallest.

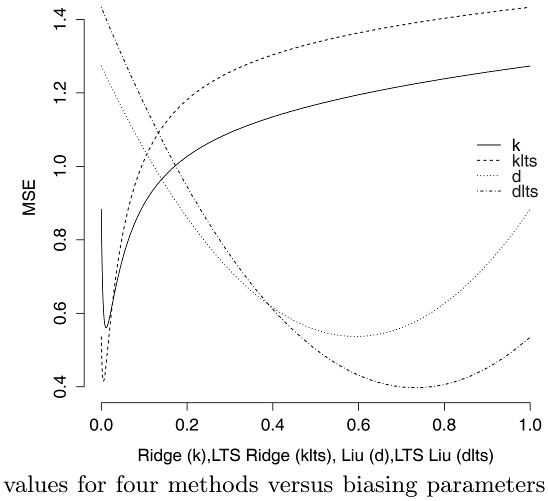

Figure 4: MSE values for four methods versus biasing parameters for outliers in both x and y direction.

As seen in Figure 4, the minimum MSE value is obtained by the LTS Liu method. Afterwards the user may have the exact calculation results by calling the `ltsbaseSummary` function.

4.2.2. ltsbaseSummary Function

The `ltsbaseSummary` function is designed to summarize the whole analysis and gives (1) the biasing parameter at minimum MSE value, (2) the regression coefficients of the model at minimum MSE, and (3) the value of minimum MSE, respectively.

```
R > ltsbaseSummary(model2)
 biasing parameter at best mse is  lts.liu.d
 1     0.712
 corresponding coefficients
 [1] -0.2657  0.1498 -0.4337  0.2606  0.0221  0.0421  0.4983
-0.2016 -0.6399
 best mse
 [1] 0.398
```

From the output, among the whole biasing parameters, the one which gives the minimum MSE is obtained by LTS Liu as 0.712.

4.2.3. ltsbaseDefault Function

The fitted values and residuals of the model which is summarized by `ltsbaseSummary` function are given in Appendix B.4.

5. Conclusions

The package `ltsbase` fills the existing gap in the `R` statistical environment and provides a convenient comparison for biased estimations based on the LTS method. The package has four important features both for users and package developers that are not available in at least some of the alternatives: `MASS::lqs` (Venables & Ripley 2002) and `robustbase:ltsReg` (Rousseeuw, P.J. and Croux, C. and Todorov, C. and Ruckstuhl, A. and Salibian-Barrera, M. and Verbeker, T. and Koller, M. and Maechler, M. 2012). First, the package provides the estimation of Ridge and Liu parameters based on the LTS method for the datasets in which both multicollinearity and outliers exist at the same time. Second, the estimates of biasing parameters at minimum MSEs are automatically calculated. Third, the user can easily obtain the MSE values of each model for comparison. Fourth, a graph of MSE values versus the biasing parameters for four biased methods are plotted as well.

In this study, all results are obtained using `R 3.0.1` (R Development Core Team 2013) with the packages `MASS` (version `7.3-26`), `robustbase` (version `0.9-8`) and `ltsbase` (version `1.0.1`).

Moreover, we introduce not only a program/package which analyses some of the biased techniques based on the LTS method but also a comparison of analysis using well-known datasets which are in the literature when outliers are existing in different directions is thought to be given and interpreted. Hence the analyst will practice with those datasets and hopefully `ltsbase` will gain confidence.

[Received: April 2014 — Accepted: March 2015]

References

Agullo, J. (2001), 'New algorithms for computing the least trimmed squares regression estimator', *Computational Statistics and Data Analysis* **36**(4), 425–439.

Alfons, A. (2013), *sparseLTSEigen: RcppEigen back end for sparse least trimmed squares regression*, R package version 0.2.0.
*http://CRAN.R-project.org/package=sparseLTSEigen

Atkinson, A. & Weisberg, S. (1991), Simulated annealing for the detection of multiple outliers using least squares and least median of squares fitting, *in* W. Stahel & S. Weisberg, eds, 'Directions in Robust Statistics and Diagnostics', Springer-Verlag, New York.

Belsley, D. (1991), *Conditioning Diagnostics: Collinearity and Weak Data in Regression*, 1 edn, John Wiley & Sons, New York.

Chatterjee, S. & Hadi, A. S. (2006), *Regression Analysis by Examples*, 4 edn, John Wiley & Sons, New York.

Cizek, P. (2005), 'Least trimmed squares in nonlinear regression under dependence', *Journal of Statistical Planning and Inference* **136**(11), 3967–3988.

Fox, J. & Weisberg, S. (2011), *An R Companion to Applied Regression*, 2 edn, Sage, Thousand Oaks, California.

Gujarati, D. (2004), *Basic Econometrics*, 4 edn, McGraw-Hill.

Hawkins, D. (1994), 'The feasible solution algorithm for least trimmed squares regression', *Computational Statistics and Data Analysis* **17**(2), 185–196.

Hawkins, D., Bradu, M. & Kass, G. (1984), 'Location of several outliers in multiple regression data using elemental sets', *Technometrics* **26**(3), 97–208.

Hawkins, D. & Olive, D. (2002), 'Inconsistency of resampling algorithms for high-breakdown regression estimators and a new algorithm', *Journal of the American Statistical Association* **97**(457), 136–159.

Heumann, C., Shalabh, Rao, C. & Toutenburg, H. (2008), *Linear Models and Generalizations- Least Squares and Alternatives*, 3 edn, Springer, New York.

Hoerl, K. & Kennard, R. (1970), 'Ridge regression: Biased estimation for nonorthogonal problems', *Technometrics* **12**(1), 55–67.

Hossjer, O. (1995), 'Exact computation of the least trimmed squares estimate in simple linear regression', *Computational Statistics and Data Analysis* **19**(3), 265–282.

Jung, K. (2005), 'Multivariate least-trimmed squares regression estimator', *Computational Statistics and Data Analysis* **48**(2), 307–316.

Kan, B., Alpu, O. & Yazici, B. (2013), 'Robust ridge and robust Liu estimator for regression based on the LTS estimator', *Journal of Applied Statistics* **40**(3), 644–655.

Kan Kilinc B. and Alpu O. (2013), *ltsbase: Ridge and Liu Estimates based on LTS Method*, R package version 101.
*http://CRAN.R-project.org/package=ltsbase

Li, L. (2005), ' An algorithm for computing exact least-trimmed squares estimate of simple linear regression with constraints', *Computational Statistics and Data Analysis* **48**(4), 717–734.

Liu, K. (1993), 'A new class of biased estimate in linear regression', *Communications in Statistics-Theory and Methods* **22**(2), 393–402.

Maguna, F. P., Nunez, M. B., Okulik, N. & Castro, E. A. (2003), 'Improved QSAR analysis of the toxicity of aliphatic carboxylic acids', *Russian Journal of General Chemistry* **73**(11), 1792–1798.

Marquardt, D. & Snee, R. (1975), 'Ridge regression in practice', *The American Statistician* **29**(1), 3–20.

Mason, R. & Gunst, R. (1985), 'Outlier-induced collinearities', *Technometrics* **27**(4), 401–407.

Neykov, N. & Neytchev, P. (1991), 'Least median of squares, least trimmed squares and S estimations by means of BMDP3R and BMDPAR', *Computational Statistics Quarterly* **4**, 281–293.

R Development Core Team (2013), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.
*http://www.R-project.org

Rousseeuw, P. J. & van Driessen, K. (1999), 'A fast algorithm for the minimum covariance determinant estimator', *Technometrics* **41**(3), 212–223.

Rousseeuw, P. & Leroy, A. (1987), *Robust Regression and Outlier Detection* , John Wiley & Sons, New York.

Rousseeuw, P. & van Driessen, K. (2006), 'Computing LTS regression for large data sets', *Data Mining and Knowledge Discovery* **12**(1), 29–45.

Rousseeuw, P.J. and Croux, C. and Todorov, C. and Ruckstuhl, A. and Salibian-Barrera, M. and Verbeker, T. and Koller, M. and Maechler, M. (2012), *robustbase: Basic Robust Statistics, R package version 0.9-8.* *http://CRAN.R-project.org/package=robustbase

Ruppert, D. (1992), 'Computing S estimators for regression and multivariate location/dispersion', *Journal of Computational and Graphical Statistics* **1**(3), 253–270.

Ruppert, D. & Carrol, R. (1980), 'Trimmed least squares estimation in the linear model', *Journal of the American Statistical Association* **75**, 828–838.

Stromberg, A. (1993), 'Computing the exact least median of squares estimate and stability diagnostics in multiple linear regression', *SIAM Journal on Scientific Computing* **14**(6), 1289–1299.

Tichavsky, P. (1991), 'Algorithms for and geometrical characterization of solutions in the LMS and the LTS linear regression', *Computational Statistics Quarterly* **6**(2), 139–151.

Venables, W. & Ripley, B. (2002), *Modern Applied Statistics with S*, 4 edn, Springer, New York.

Willems, G. & van Aelst, S. (2005), 'Fast and robust bootstrap for LTS', *Computational Statistics and Data Analysis* **48**(4), 703–715.

Wissmann, M., Toutenburg, H. & Shalabh (2007), *Role of Categorical Variables in Multicollinearity in Linear Regression Model*, Technical Report Department of Statistics University of Munich, Germany.

Appendix A. Appendices

These appendices consist of codes required to run the examples.

Appendix A.1. Illustrations of *hbk* Data

This section provides raw data, multicollinearity and outlier detection and code to show particular features of `ltsbaseDefault` function.

Appendix A.2. Raw Data

Here, we illustrate how to load *hbk* data and give develop raw data to some opinion about it:

```
R > library(robustbase)
R > data(hbk)
R > head(hbk)
    X1   X2   X3    Y
1 10.1 19.6 28.3  9.7
2  9.5 20.5 28.9 10.1
3 10.7 20.2 31.0 10.3
4  9.9 21.5 31.7  9.5
5 10.3 21.1 31.1 10.0
6 10.8 20.4 29.2 10.0
```

Then data are set up as:

```
R > y=hbk[,4]
R > xdata=data.frame(hbk[,1:3])
```

Appendix A.3. Detecting Multicollinearity

Detecting multicollinearity via VIFs:

For diagnosing the multicollinearity, the Variance Inflation Factors (`VIF`) can be used. These measures are based on the fact that a centered and scaled design matrix is the correlation matrix of regressors. The intercept term is then excluded while using this diagnostic. The homoscedastic variance of the estimate of j^{th} regression coefficient is then a function of multiple correlation from the regression of the j^{th} column on all other columns of the design matrix. The term around the multiple correlation is given as the variance inflation factor of the j^{th} regression coefficient. The following code runs the VIF calculation using the R package `car` (Fox & Weisberg 2011).

```
R > library(car)
R > vif(lm(y~., data=hbk[,-4]))
      X1       X2       X3
13.43200 23.85346 33.43249
```

As seen, VIF's are greater than 10 which means there is a multicollinearity problem.

Detecting multicollinearity via Condition Number:

The degree of multicollinearity can also be calculated using a Condition Number (CN) that is a ratio of the maximum eigenvalue divided by the minimum eigenvalue ($\lambda_{\max}/\lambda_{\min}$). As a rule of thumb, if the CN k is between 100 and 1000 there is moderate multicollinearity and if it exceeds 1000 there is severe multicollinearity.

```
R > xdata=hbk[,1:3]
R > eigen(t(xdata)%*%xdata)
[1] 22982.6676   155.7312   114.1612
```

Here, CN is calculated as 201.316 which indicates there is moderate multicollinearity.

Appendix A.4. Detecting Outliers via Plotting

The following code from library `robustbase` is used to detect outliers visually.

```
R> plot(ltsReg(xdata,y,intercept=TRUE,method="lts"),which=c("rdiag"))
```

The Figure 1 in Section 4 shows outliers in x direction.

Appendix A.5. Fitted Values and Residuals

The following code runs the `ltsbaseDefault` function given in Section 4.1.3. The function provides two structures: (1) the fitted values, (2) residuals. They are obtained for each method and given in separate columns to compare easily.

```
R > ltsbaseDefault(xdata,y,alpha=0.875,by=0.001)
$fitted.val
        OLS    LTS       Ridge    Liu     LTS.Ridge LTS.Liu
 [1,] 23.5237  28.0458  37.1980  36.0450  -1.6199   2.2311
 [2,] 23.4365  28.6288  38.5111  37.3029  -1.1782   2.5924
...
[74,]  0.3255   1.7660   2.7348   2.6212   0.1251   0.2762
[75,]  3.1095   2.1230   3.1995   3.1329   0.4947   0.6129
$res
        OLS       LTS      Ridge      Liu   LTS.Ridge  LTS.Liu
 [1,] -13.8237 -18.3458 -27.4980 -26.3450  11.3199   7.4689
 [2,] -13.3365 -18.5288 -28.4111 -27.2029  11.2782   7.5076
...
[74,]  -1.2255  -2.6660  -3.6348  -3.5212  -1.0251  -1.1762
[75,]  -2.9095  -1.9230  -2.9995  -2.9329  -0.2947  -0.4129
```

Appendix B. Illustrations of *toxicity* Data

This section provides tabulated data, multicollinearity detection and code to show particular features of `ltsbaseDefault` function.

Appendix B.1. Tabulated Data

Table 1 describes the response variable and several molecular descriptors with 38 observations (Maguna et al. 2003).

A closer look at the toxicity data is briefly given by `head()` :

```
R > head(toxicity)
  toxicity logKow  pKa ELUMO   Ecarb   Emet    RM    IR   Ts     P
1    -0.15   1.68 1.00  4.81 17.8635 1.4838 31.36 1.425 31.3 12.43
2    -0.33   0.94 0.98  4.68 16.9491 0.0000 22.10 1.408 30.4  8.76
3    -0.34   1.16 0.96  4.86 17.1806 0.2778 26.73 1.418 30.9 10.59
4     0.03   2.75 1.00  4.83 18.4794 3.5836 40.63 1.435 31.8 16.10
5    -0.57   0.79 0.97  4.80 16.8022 1.0232 22.14 1.411 32.5  8.77
```

Appendix B.2. Detecting Multicollinearity

Detecting multicollinearity via VIFs:

Some authors use the VIF as an indicator of multicollinearity. Hence it is commonly agreed that if the VIF of a variable exceeds 10, which will happen if R_j^2 exceeds 0.90, that variable is said to be highly collinear (Gujarati 2004). The following code runs the VIFs for toxicity data:

```
R > vif(lm(toxicity~.,data=toxicity))
 logKow  pKa ELUMO Ecarb  Emet    RM      IR      Ts     P
36.949 7.452 2.577 15.095 13.550 52.067 15.773 14.059 9.093
```

Here the maximum VIF is 52.067. So it is clear that there is strong evidence of multicollinearity in the data.

Detecting multicollinearity via Condition Number:

The following code runs the eigenvalue analysis for CN:

```
R > xdata=toxicity[,-1]
R > eigen(t(xdata)%*%xdata)
$values
[1] 1.180085e+05 5.589917e+03 1.592269e+03 2.764111e+02 5.570724e+01
[6] 1.203306e+01 6.371221e+00 9.131830e-01 8.541290e-03
```

Here, CN is obtained as 13883353 which is fairly large. This that indicates there is strong multicollinearity in the data.

Appendix B.3. Detecting Outliers via Plotting

The same code given in Appendix A.3 runs the code for detecting outliers via plotting.

Appendix B.4. Fitted Values and Residuals

The following code runs the example to get fitted values and residuals.

```
R > ltsbaseDefault(xdata,y,alpha=0.5)
$fitted.val
            OLS       LTS     Ridge       Liu  LTS.Ridge    LTS.Liu
 [1,] -23.02504 -32.33508 -23.75030 -23.75030 -14.225909 -52.170047
 [2,] -19.95794 -26.44037 -22.23670 -22.23670 -10.920423 -45.989539
...
[37,] -29.09435 -42.72803 -14.29480 -14.29480  -5.414905 -14.730020
[38,] -21.58501 -26.37637 -17.82639 -17.82639  -5.029148 -27.935130
$res
           OLS      LTS    Ridge      Liu LTS.Ridge   LTS.Liu
 [1,] 22.87504 32.18508 23.60030 23.60030 14.075909 52.020047
 [2,] 19.62794 26.11037 21.90670 21.90670 10.590423 45.659539
...
[37,] 29.69435 43.32803 14.89480 14.89480  6.014905 15.330020
[38,] 20.94501 25.73637 17.18639 17.18639  4.389148 27.295130
```