enetLTS: Robust and Sparse Methods for High Dimensional Linear, Binary, and Multinomial Regression

Fatma Sevinc KURNAZ [1] **and Peter FILZMOSER** [2]

1 Department of Statistics, Yildiz Technical University, Istanbul, Turkey **2** Institute of Statistics and Mathematical Methods in Economics, TU Wien, Vienna, Austria

Summary

enetLTS is an R package (R-Development-Core-Team, 2021) that provides a fully robust version of the elastic net estimator for high dimensional linear, binary, and multinomial regression. The elastic net penalization provides intrinsic variable selection and coefficient estimates for highly correlated variables, in particular for high dimensional low sample size data sets, and it has been extended to generalized linear regression models (Friedman et al., 2010). Combining these advantages with trimming outliers yields the robust solutions. The main idea of the algorithm is to search for outlier-free subsets on which the classical elastic net estimator can be applied. Outlier-free subsets are determined by trimming the penalized log-likelihood function for the considered regression model. The algorithm starts with 500 elemental subsets only for one combination of the elastic net parameters α and λ, and takes the *warm start* strategy for subsequent combinations in order to save the computation time. The final reweighting step is added to improve the statistical efficiency of the proposed estimators. From this point of view, the enet-LTS estimator can be seen as a trimmed version of the elastic net regression estimator for linear, binary, and multinomial regression (Friedman et al., 2010). Selecting model with the optimal tuning parameters is done via cross-validation, and various plots are available to illustrate model selection and to evaluate the final model estimates.

Statement of need

A number of new robust linear regression methods have been developed during the last decade in the context of high dimensional data, such as Alfons et al. (2013); Alfons (2021); Kepplinger et al. (2021). However, to the best of our knowledge, robust logistic (both binary and multinomial) regression for high dimensional data is not available elsewhere. The package enetLTS therefore provides researchers with access to robust solutions and variable selection at the same time with high-dimensional linear and logistic regression data. It has already been used in several benchmark studies in the statistical literature, e.g. (Insolia et al., 2021, 2022; Monti & Filzmoser, 2021), as well as in empirical research, e.g. (Jensch et al., 2022; Segaert et al., 2018).

Example: Robust and Sparse Linear Regression (`family="gaussian"`)

We have considered the NCI-60 cancer cell panel data (Reinhold et al., 2012) in order to provide an example for the enetLTS model. The NCI-60 data set includes 60 human cancer cell lines with nine cancer types, which are breast, central nervous system, colon, leukemia, lung, melanoma, ovarian, prostate and renal cancers. In this example, we regress the protein

expression on gene expression data. Using the Affymetrix HG-U133A chip and normalizing with the GCRMA method, the number of predictors is obtained as 22,283. One observation with missing values is omitted. This data set is available in the package `robustHD`.

As in Alfons (2021) we determine the response variable with one of the protein expressions which is 92th protein. Out of the gene expressions of the 22,283 genes for predictors, we have considered the gene expressions of the 100 genes that have the highest (robustly estimated) correlations with the response variable. The code lines for loading and re-organizing the response variable and the predictors is as follows:

```r
# load data
library("robustHD")
data("nci60")  # contains matrices 'protein' and 'gene'

# define response variable
y <- protein[, 92]
# screen most correlated predictor variables
correlations <- apply(gene, 2, corHuber, y)
keep <- partialOrder(abs(correlations), 100, decreasing = TRUE)
X <- gene[, keep]
```

The package enetLTS can either be installed from `CRAN` or directly from `Github`. The main function is enetLTS(), and the default `family` option is gaussian, which corresponds to linear regression.

```r
# install and load package
install.packages("enetLTS")
# alternatively install package from Github
# library(devtools)
# install_github("fatmasevinck/enetLTS",force=TRUE)
library(enetLTS)
# fit the model for family="gaussian"
set.seed(1)
fit.gaussian <- enetLTS(X, y, crit.plot=TRUE)
## [1] "optimal model: lambda = 0.1043 alpha = 0.8"

fit.gaussian

## enetLTS estimator

## Call:  enetLTS(xx = X, yy = y, crit.plot=TRUE)

## number of the nonzero coefficients:
## [1] 23

## alpha: 0.8
## lambda: 0.1043
## lambdaw: 0.1824974
```

23 out of 100 independent variables are selected by the enetLTS model based on optimal combination of $\alpha = 0.8$ and $\lambda = 0.1043$. Here $\lambda_w = 0.1824974$ corresponds to updated tuning parameter for reweighted model.

The main idea to obtain an outlier-free subset is to carry out concentration steps (C-steps). This means that in each iteration of the algorithm, the value of the objective function improves. Thus, one has to start with several initial subsets, and the C-steps will lead at least to a local optimum.

For the argument `hsize` one needs to provide a numeric value with the trimming percentage

---

used in the penalized objective function. The default value is 0.75. The argument `nsamp` is a numeric vector: The first element gives the number of initial subsamples to be used. The second element gives the number of subsamples to keep after a number of C-steps has been performed. For those remaining subsets, additional C-steps are performed until convergence. The default is to start the C-steps with 500 initial subsamples for a first combination of tuning parameters α and λ, and then to keep the 10 subsamples with the lowest value of the objective function for additional C-steps until convergence. For the next combination of tuning parameters α and λ, the algorithm makes use of the $warmstart$ idea, which means that the best subset of the neighboring grid value is taken, and C-steps are started from this best subset until convergence. The `nsamp` entries can also be supplied by the user. These arguments are the same for the other `family` options.

The main function `enetLTS()` allows the user to specify a sequence of values for α for the elastic net penalty. If this is not provided, a default sequence of 41 equally spaced values between 0 and 1 is taken. For the other tuning parameter λ that keeps the strength of the elastic net penalty, a user supplied sequence is available. If not provided, the default for `family="gaussian"` is chosen with steps of size -0.025 lambda0 with $0 \leq \lambda \leq$ lambda0, where lambda0 is determined as in Alfons (2021).

After computing all candidates based on the best subsets for certain combinations of α and λ, the combination of the optimal tuning parameters is defined by 5-fold cross-validation. The evaluation criterion for 5-fold cross-validation is summarized by a heat map, see Figure 1, if the argument `crit.plot` is assigned to "TRUE".

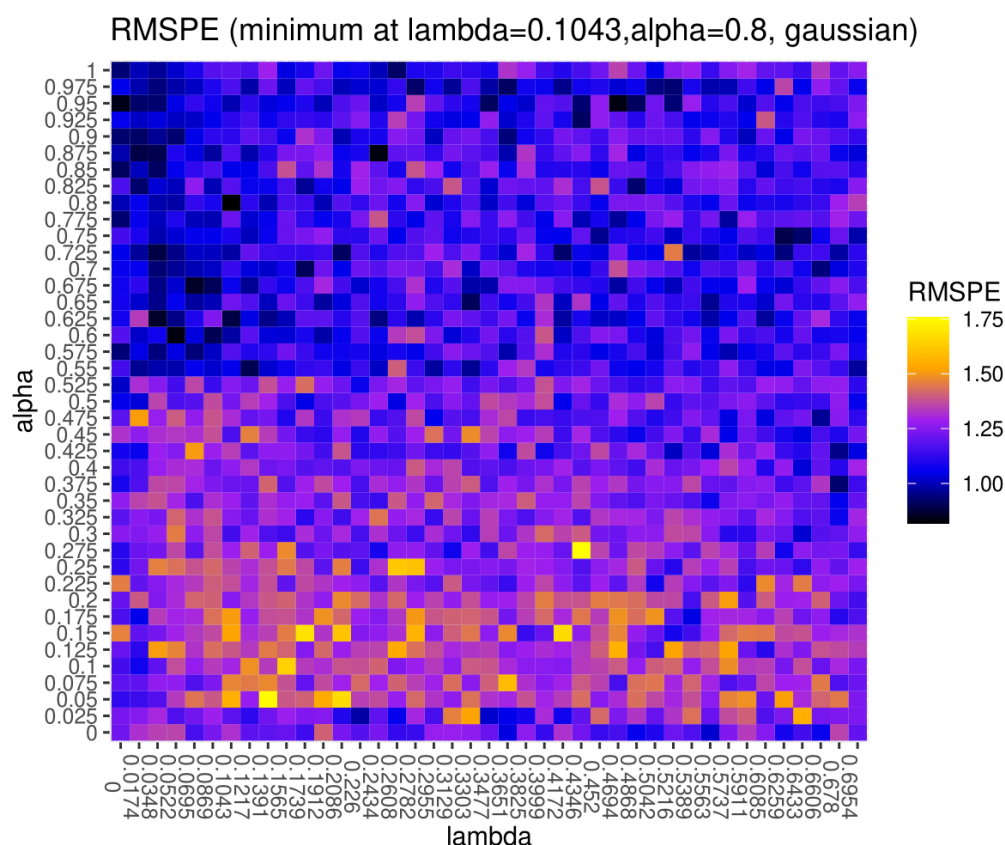## RMSPE (minimum at lambda=0.1043,alpha=0.8, gaussian)

Figure 1: Heatmap for 5-fold cross-validation

To determine updated parameter λ (`lambdaw`) in a reweighting step, 5-fold cross-validation based on the `cv.glmnet()` function from `glmnet` (Friedman et al., 2021) is used.

Several plots are available for the results. `plotCoef.enetLTS()` visualizes the coefficients where the coefficients which are set to zeros are shown, `plotResid.enetLTS()` plots the values of residuals vs fitted values, and `plotDiagnostic.enetLTS()` allows to produce various diagnostic plots for the final model fit. Some examples of these plots are shown in Figure 2.
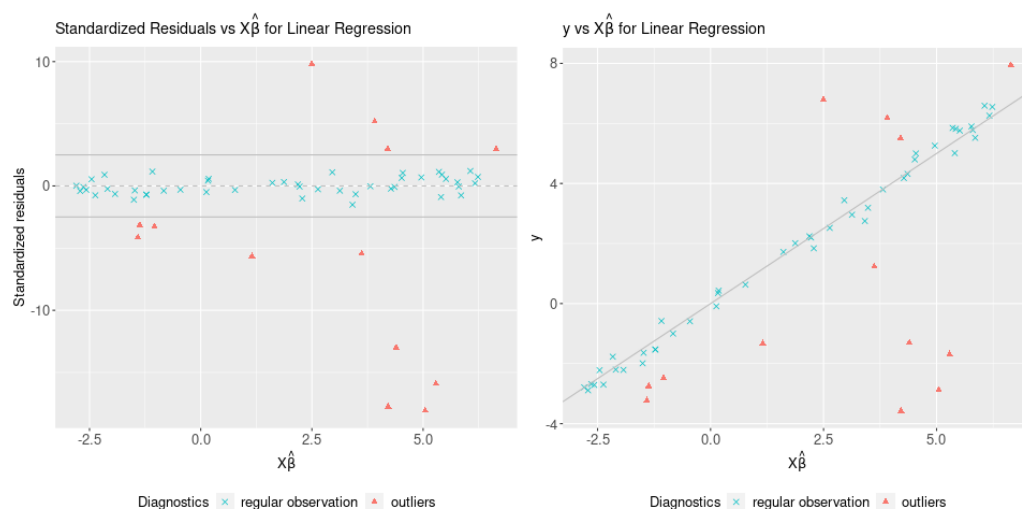

Figure 2: Examples of plot functions of residuals (left); diagnostic (right) for linear regression

Example: Robust and Sparse Binary Regression (`family="binomial"`)

For binary regression, we have considered the same NCI-60 data set with some modifications. In order to provide an example for binary regression, the response variable is re-organized as follows. If `mean(y)` is smaller than 0.5, the response will be assigned to 0, otherwise, the response will be assigned to 1. The predictors are the same as in the previous section.

```
y <- protein[, 92]
# for binary class
y.binom <- ifelse(y <= mean(y),0,1)
```

For the binary regression, the `family` argument of `enetLTS()` function should be set to `"binomial"`.

```
alphas <- seq(0,1,length=41)
l0 <- lambda00(X, y.binom, normalize = TRUE, intercept = TRUE)
lambdas <- seq(l0,0.001,by=-0.025*l0)

# fit the model for family="binomial"
set.seed(12)
fit.binomial <- enetLTS(X, y.binom, family="binomial", alphas=alphas,
                                    lambdas=lambdas)

fit.binomial

## enetLTS estimator

## Call:  enetLTS(xx = X, yy = y.binom, family = "binomial", alphas = alphas,
##                lambdas = lambdas)

## number of the nonzero coefficients:
## [1] 48
```

```
## alpha: 0.325
## lambda: 0.0011
## lambdaw: 0.01456879
```

48 out of 100 independent variables are selected by the enetLTS model based on optimal combination of $\alpha = 0.325$ and $\lambda = 0.0011$.

The main function `enetLTS()` provides similar options for the values of the elastic net penalty. For the tuning parameter λ, a user supplied sequence option is available. If this is not provided, the default is chosen with steps of size -0.025 lambda00 with $0 \leq \lambda \leq$ lambda00, where lambda00 is determined based on the robustified point-biserial correlation, see Kurnaz et al. (2018).

The evaluation criterion results for to the candidates of tuning parameters is available in a heat map if the argument `crit.plot` is assigned to "TRUE" (which is omitted here). To determine the updated parameter λ (`lambdaw`) for the reweighting step, 5-fold cross-validation based on the `cv.glmnet()` function is used from the `glmnet` package for the current `family` option.

Similarly, `plotCoef.enetLTS()` visualizes the coefficients. The other plot functions are reorganized for binary regression. In `plotResid.enetLTS()`, residuals are turned into the deviances, and this plot function produces two plots which are deviances vs index, and deviances vs fitted values (link function). `plotDiagnostic.enetLTS()` shows the response variable vs fitted values (link function). Some of these plots are presented in Figure 3.
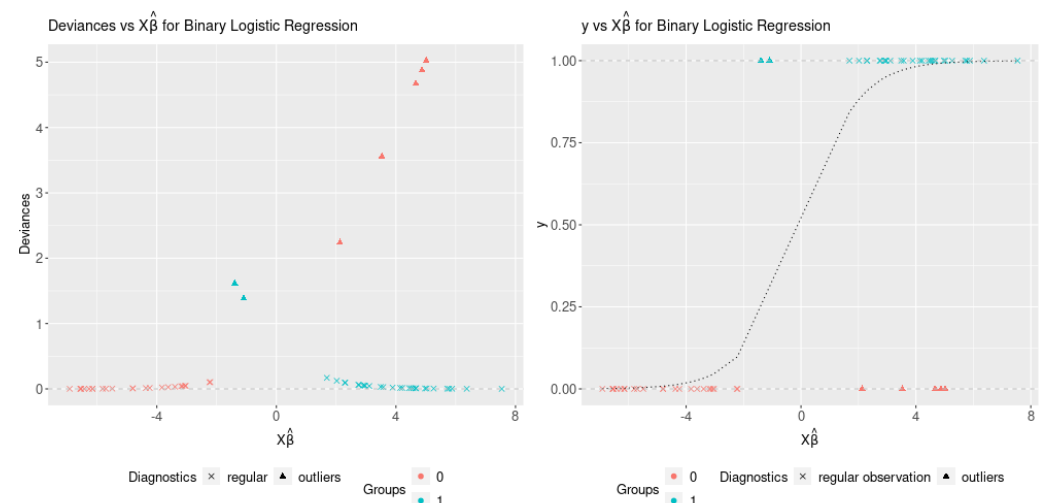

Figure 3: Examples of plot functions of deviances (left); diagnostic (right) for binary regression

Example: Robust and Sparse Multinomial Regression (`family="multinomial"`)

The fruit data set has been well-known in the context of robust discrimination studies. Therefore, we have considered the fruit data set in order to illustrate multinomial regression. It contains spectral information with 256 wavelengths for observations from 3 different cultivars of the same fruit, named D, M, and HA, with group sizes 490, 106, and 500. This data set is available in the R package `rrcov`.

```
# load data
library(rrcov)
data(fruit)

d <- fruit[,-1]  # first column includes the fruit names
X <- as.matrix(d)
```

```
# define response variable
grp <- c(rep(1,490),rep(2,106),rep(3,500))
y <- factor(grp-1)
```

With `family="multinomial"`, the model `enetLTS()` produces the results of multinomial regression. Here user supplied values of `lambdas` are considered.

```
lambdas=seq(from=0.01,to=0.1,by=0.01)
set.seed(4)
fit.multinom <- enetLTS(X, y, family="multinomial", lambdas=lambdas,
                        crit.plot=FALSE)
## [1] "optimal model: lambda = 0.01 alpha = 0.02"


fit.mutinom

## enetLTS estimator

## Call:  enetLTS(xx = X, yy = y, family = "multinomial", lambdas=lambdas,
##               crit.plot = FALSE)

## number of the nonzero coefficients:
## [1] 704

## alpha: 0.02
## lambda: 0.01
## lambdaw: 0.003971358
```

704 out of 1096 independent variables are selected by the enetLTS model based on optimal combination of $\alpha = 0.2$ and $\lambda = 0.01$. Here $\lambda_w = 0.003971358$ corresponds to updated tuning parameter for reweighted model. The effect of tuning parameter α on the model is very clear from less sparsity.

The main function `enetLTS()` provides similar options for the α sequence of the elastic net penalty. The default for the tuning parameters λ are values from 0.95 to 0.05 with steps of size -0.05, see Kurnaz & Filzmoser ([2022](#)).

The combination of the optimal tuning parameters is evaluated by 5-fold cross-validation. A heat map is available if the argument `crit.plot` is assigned to "TRUE". As for the other models, an updated tuning parameter λ (`lambdaw`) for the reweighting step is obtained by the `cv.glmnet()` function from the package glmnet ([Friedman et al., 2021](#)).

The plot functions are adjusted for multinomial regression. `plotCoef.enetLTS()` gives the coefficients plots which includes group information. In `plotResid.enetLTS()`, residuals are turned into deviances, as in the binary regression case, with group information. `plotDiagnostic.enetLTS()` shows the scores of all groups in the space of the first two principal components.

Especially for 'family="multinomial"', run time is long because the algorithm is based on repeated C-steps.

Related Software

The package `robustHD` provides the sparseLTS estimator for linear regression based on trimming of the lasso penalized for high dimensional linear regression ([Alfons, 2021](#)). The package pense provides implementations of robust S- and MM-type estimators using elastic net regularization for linear regression ([Kepplinger et al., 2021](#)). These packages are designed for linear regression, but not extended for binary or multinomial regression. On the other hand, the package glmnet implements the elastic net estimator for linear, binary, multinomial regression models and

more (Friedman et al., 2021). The procedure of the R package enetLTS (Kurnaz et al., 2022) is implemented using the R package glmnet (Friedman et al., 2021). Therefore, taking the advantages of this internal usage, the package enetLTS provides a robust and sparse estimator based on trimming of the elastic net penalized for high dimensional linear, binary and multinomial regression.

Acknowledgements

References

Alfons, A. (2021). robustHD: An R package for robust regression with high-dimensional data. *Journal of Open Source Software*, *6*(67), 3786. https://doi.org/10.21105/joss.03786

Alfons, A., Croux, C., & Gelper, S. (2013). Sparse least trimmed squares regression for analyzing high-dimensional large data sets. *Annals of Applied Statistics*, *7*(1), 226–248. https://doi.org/10.1214/12-AOAS575

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, *33*(1), 1–22. https://doi.org/10.18637/jss.v033.i01

Friedman, J., Hastie, T., Tibshirani, R., Narasimhan, B., Tay, K., Simon, N., Qian, J., & Yang, J. (2021). Glmnet: Lasso and Elastic-Net Regularized Generalized Linear Models. *R Foundation for Statistical Computing, Vienna, Austria. R Package Version 4.1–3 Https://CRAN.R-Project.org/Package=glmnet*.

Insolia, L., Kenney, A., Calovi, M., & F.Chiaromonte. (2021). Robust variable selection with optimality guarantees for high-dimensional logistic regression. *Stats*, *4*(3), 665–681. https://doi.org/10.3390/stats4030040

Insolia, L., Kenney, A., Chianomante, F., & Felici, G. (2022). Simultaneous feature selection and outlier detection with optimality guarantees. *Biometrics*. https://doi.org/10.1111/biom.13553

Jensch, A., Lopes, M. B., Vinga, S., & Radde, N. (2022). ROSIE: RObust sparse ensemble for outlier detection and gene selection in cancer omics data. *Statistical Methods in Medical Research*, *31*(5), 947–958. https://doi.org/10.1177/09622802211072456

Kepplinger, D., Salibian-Barrera, M., & Freue, G. C. (2021). Pense: Penalized elastic net s/MM-estimator of regression. *R Foundation for Statistical Computing, Vienna, Austria. Https://CRAN.R-Project.org/Package=pense*.

Kurnaz, F. S., & Filzmoser, P. (2022). Robust and sparse multinomial regression in high dimensions. *Arxiv*. https://doi.org/10.48550/arXiv.2205.11835

Kurnaz, F. S., Hoffmann, I., & Filzmoser, P. (2018). Robust and sparse estimation methods for high-dimensional linear and logistic regression. *Chemometrics and Intelligent Laboratory Systems*, *172*, 211–222. https://doi.org/10.1016/j.chemolab.2017.11.017

Kurnaz, F. S., Hoffmann, I., & Filzmoser, P. (2022). enetLTS: Robust and sparse estimation methods for high-dimensional linear and binary and multinomial regression. *R Foundation for Statistical Computing, Vienna, Austria. R Package Version 1.1.0 Https://CRAN.R-Project.org/Package=enetLTS*.

Monti, G. S., & Filzmoser, P. (2021). Robust logistic zero-sum regression for microbiome compositional data. *Advances in Data Analysis and Classification*. https://doi.org/10.1007/s11634-021-00465-4

R-Development-Core-Team. (2021). *R Foundation for Statistical Computing Vienna Austria. Https://www.r-project.org/.*

Reinhold, W. C., Sunshine, M., Liu, H., Varma, S., Kohn, K. W., Morris, J. J., Doroshow, J., & Pommier, Y. (2012). CellMiner: A web-based suite of genomic and pharmacologic tools to explore transcript and drug patterns in the NCI-60 cell line set. *Cancer Research*, *72*(14), 3499–3511. https://doi.org/10.1158/0008-5472.can-12-1370

Segaert, P., Lopes, M. B., Casimiro, S., Vinga, S., & Rousseeuw, P. (2018). Robust identification of target genes and outliers in triple-negative breast cancer data. *Statistical Methods in Medical Research*, *28*, 3042–3056. https://doi.org/10.1177/0962280218794722