UNIVERSITÄT DORTMUND
SFB 475
Komplexitätsreduktion in multivariaten Datenstrukturen

UNIVERSITÄT DORTMUND

An evolutionary algorithm for LTS-Regression: A comparative study

Oliver Morell, Fachbereich Statistik, Universität Dortmund
Thorsten Bernholt, Robin Nunkesser, Fachbereich Informatik, Universität Dortmund
Roland Fried, Joachim Kunert, Fachbereich Statistik, Universität Dortmund

1.) Least Trimmed Squares Regression

If outliers occur, Ordinary Least Squares (OLS)–Regression can lead to an estimation, which does not fit well at all to the observations.

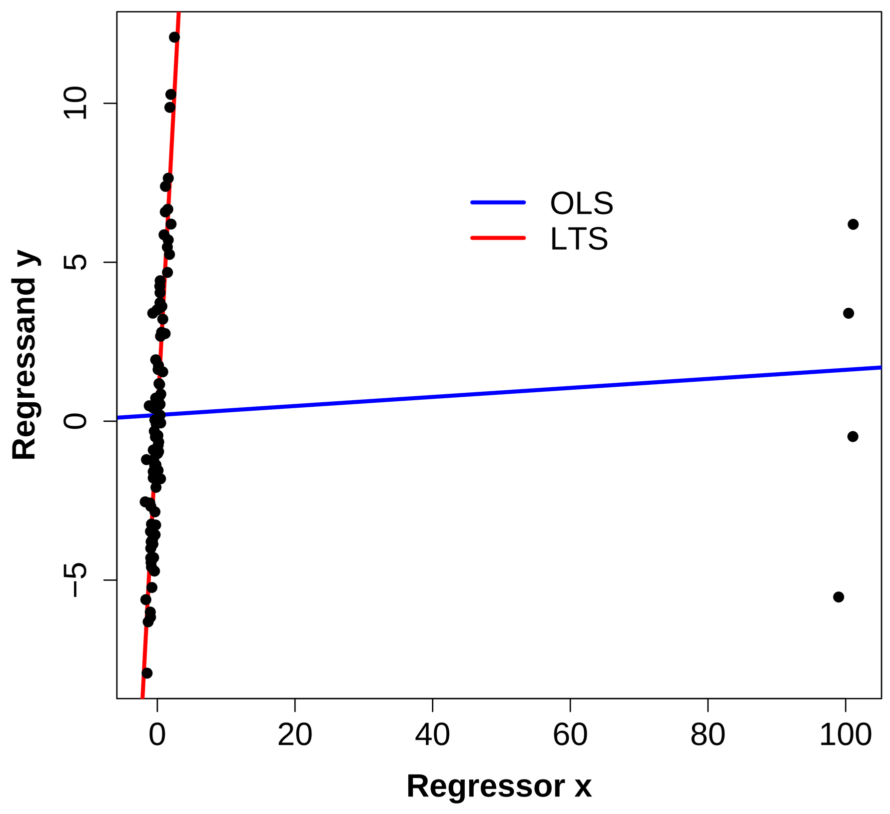


Figure 1: *Simple regression with outliers in the regressor*

Robust regression methods like Least Trimmed Squares (LTS) then still achieve a good fit to the majority of the observations.

$$\widehat{\beta}_{LTS} \;=\; \underset{\widehat{\beta}\,\in\,\mathbb{R}^p}{\arg\min} \sum_{i=1}^{h} \left(\widehat{e}_{(i)}\right)^2$$

with

- n number of observations $y_1, \ldots, y_n$
- p number of regressors with slopes $\beta_1, \ldots, \beta_p$
- h number of residuals included in the trimmed sum.

$$h = \left\lfloor \frac{n}{2} \right\rfloor + \left\lfloor \frac{p+1}{2} \right\rfloor$$

to achieve an optimal breakdown point of 50% asymptotically.

- $\widehat{e}_1, \ldots, \widehat{e}_n$ estimated model residuals:

$$\widehat{e}_i \;=\; y_i - \sum_{j=1}^{p} x_{ij}\widehat{\beta}_j$$

Exact computation of LTS is very demanding, particularly if p is large.

2.) Fast–LTS (FL) (Rousseeuw & van Driessen, 2006)

0.) Start with an OLS–Estimation $\beta^{(0)}$ of β based on p observations.
1.) Calculate the residuals of all observations w. r. t. the current fit $\beta^{(l)}$.
2.) Take the indices of the h smallest squared residuals

$$J_1 \;=\; \left\{ i \in \{1,\,2,\,\ldots,\,n\} :\; \widehat{e}_i \in \left\{ (\widehat{e}_{(1)})^2,\;\ldots,\;(\widehat{e}_{(h)})^2 \right\} \right\}.$$

3.) Calculate a new OLS–Estimation $\beta^{(l+1)}$ for the observations in J_1.
4.) Stop when the LTS–criterion no longer decreases.
Go to **1.)** otherwise.

Intercept adjustment possible to further reduce the LTS–criterion.

Adjustable parameters in R–procedure `lqs`: number of iterations (`nsamp`), number of observations in the start population (`psamp`) and intercept adjustment (`adjust`).

3.) Evolutionary Algorithm (EA) for LTS

0.) Start with an OLS–Estimation $\beta^{(0)}$ of β based on p observations.
1.) Change these p observations with probability π by a move and with probability $1-\pi$ by a mutation.
2.) Compute the new LTS–criterion–value.
3.) Stop when n_W generations do not supply an improvement.

Mutation: replace one of the p points by one of the $n-p$ other points.

Move: parallel shift of the estimated hyperplane, such that it hits a random one of the n-p remaining points.

π is controlled with the parameter percentagemove `PER`.

Simulated Annealing `ANN` for not getting trapped in a local optimum.

Intercept adjustment `ADJ` possible for reducing the LTS–value further.

Further parameters: waitforimprovement `WFI`, which is controlling n_W, total number of generations `GEN` and total number of iterations

4.) Designed Experiments for comparison of algorithms

Goal: Find optimal parameter setting leading to small LTS–criterion–values and small computation times for each algorithm.

Approach 1: Central Composite Designs to find a good Taylor–Approximation of second order.

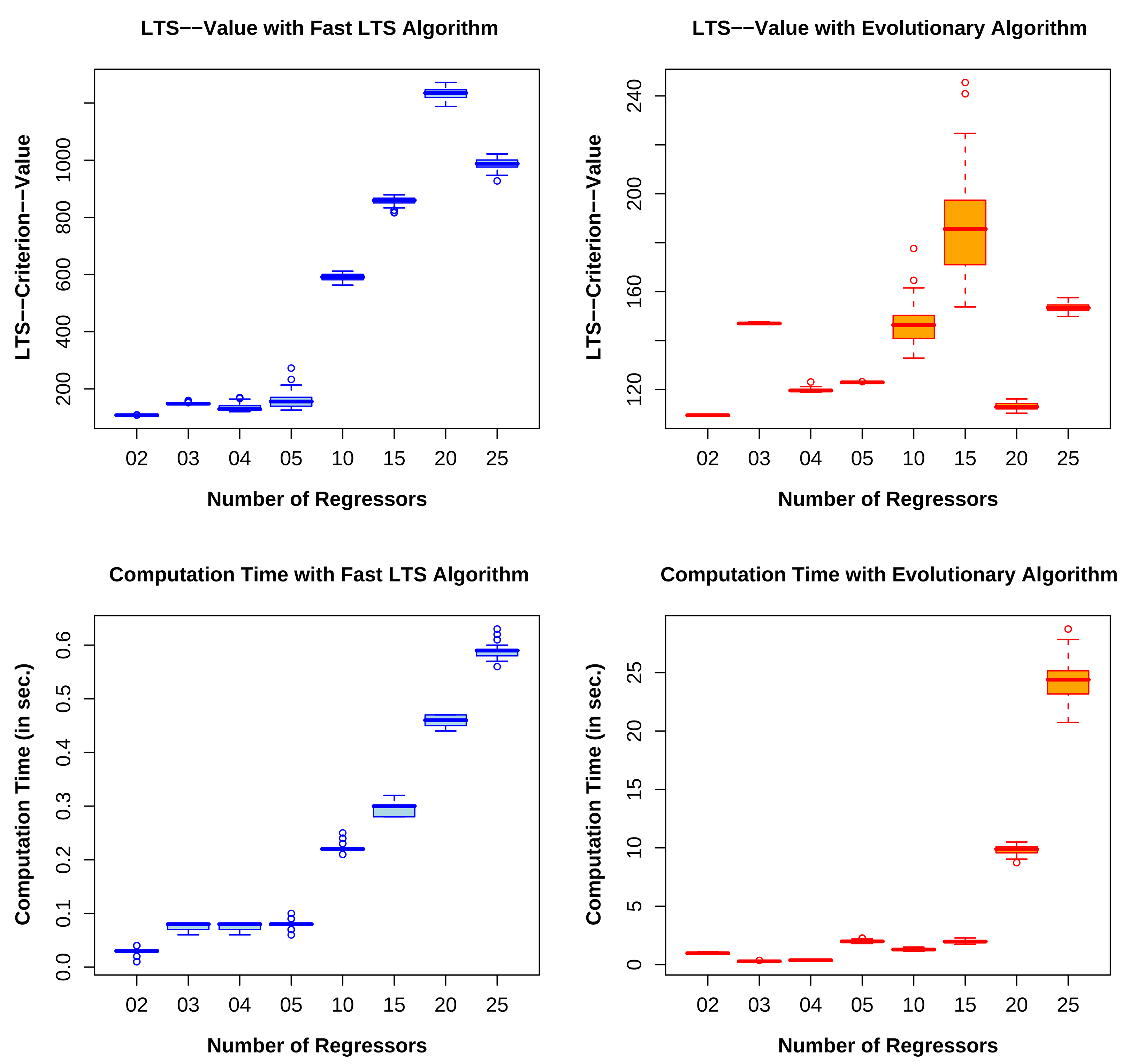
Grize's Scaled Lambda–Plot and Half–Normal–Plot to find the most important factors and interactions.

Approach 2: Orthogonal spacefilling Latin Hypercube Designs to find the region with the best values of both responses simultaneously.

Both approaches use desirability indices for bivariate optimization:

Findings for most of the datasets:

- both approaches deliver a similar optimal setting: a Taylor–Approximation of level two seems to be adequate.
- with an increasing number of regressors, EA delivers much better LTS–values than FL.
- higher computation times for the EA with optimal setting in comparison to the FL with optimal setting. Algorithmic tricks to speed up the implementation have not been considered so far.



References:

- Rousseeuw, P.J., van Driessen, K. (2006), Computing LTS Regression for Large Data Sets, *Data Mining and Knowledge Discovery*, **12**, 29–45.
- Morell, Oliver (2006), Vergleich von Algorithmen für die Least–Trimmed–Squares–Schätzung mittels Statistischer Versuchsplanung, *Diplomarbeit, Universität Dortmund*.