Erasmus
School of
Economics

Sparse Robust Regression and Model Selection

Andreas Alfons

ICORS Workshop, September 20, 2021

Erasmus University Rotterdam

Content

Motivation

Motivation

Many empirical applications typically have data with $p > n$ or $p \gg n$

- Gene expression
- fMRI
- Chemometrics
- Financial or macroeconomic time series

Two common strategies for model selection:

- Add penalty on coefficent estimates to objective function
  $\longrightarrow$ Certain penalties allow for sparse model estimates
- Sequentially add variables according to their importance

Motivation

Many empirical applications typically have data with $p > n$ or $p \gg n$

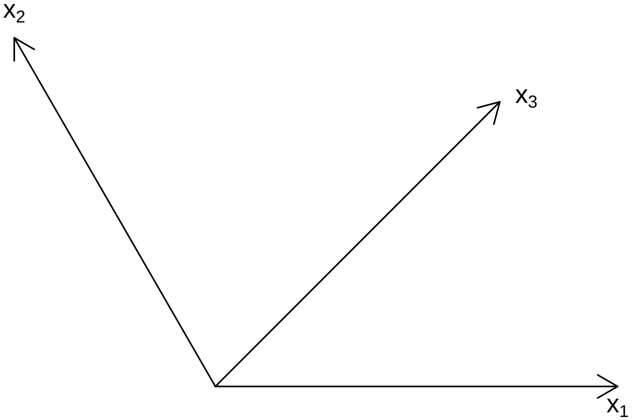
- Gene expression
- fMRI
- Chemometrics
- Financial or macroeconomic time series

Two common strategies for model selection:

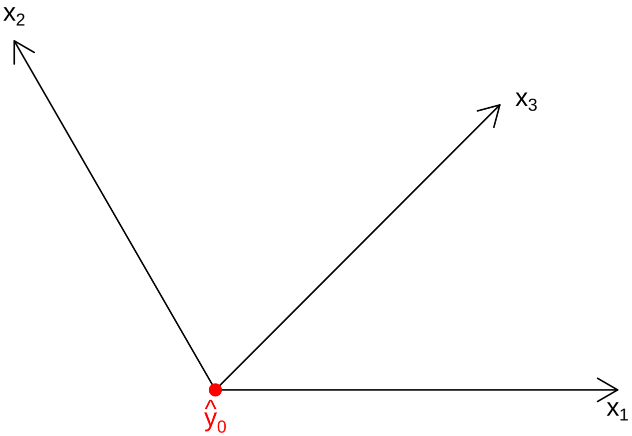
- Add penalty on coefficent estimates to objective function
  $\longrightarrow$ Certain penalties allow for sparse model estimates
- Sequentially add variables according to their importance

LARS and lasso

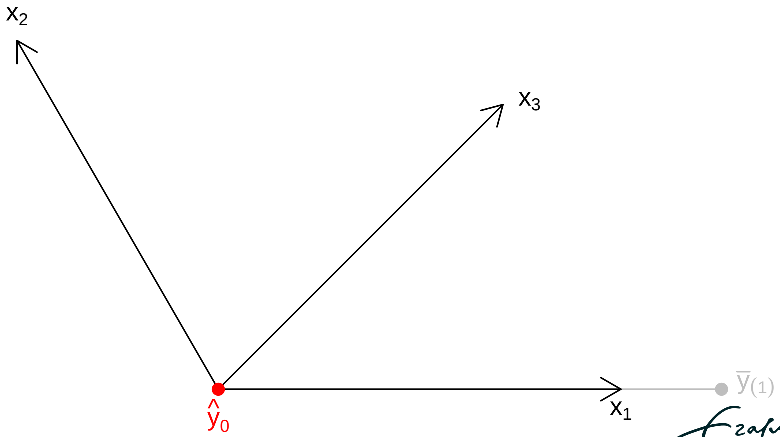
Least angle regression (LARS): Idea



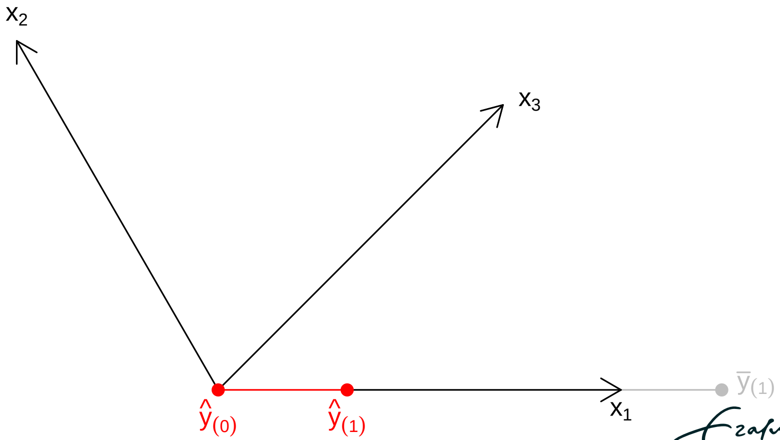
Least angle regression (LARS): Idea



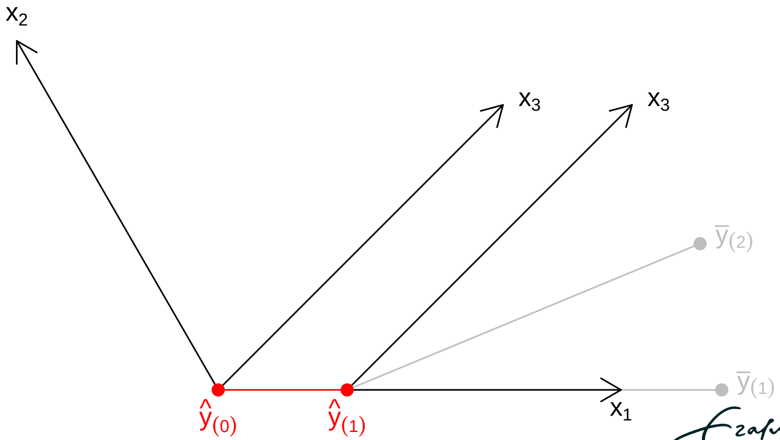
Least angle regression (LARS): Idea



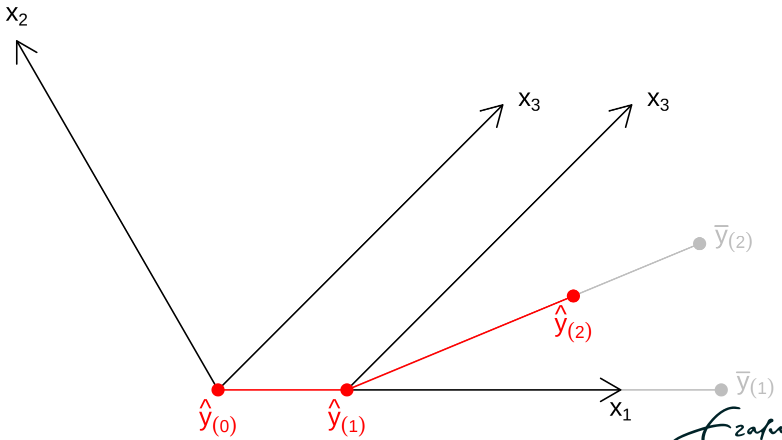
Least angle regression (LARS): Idea



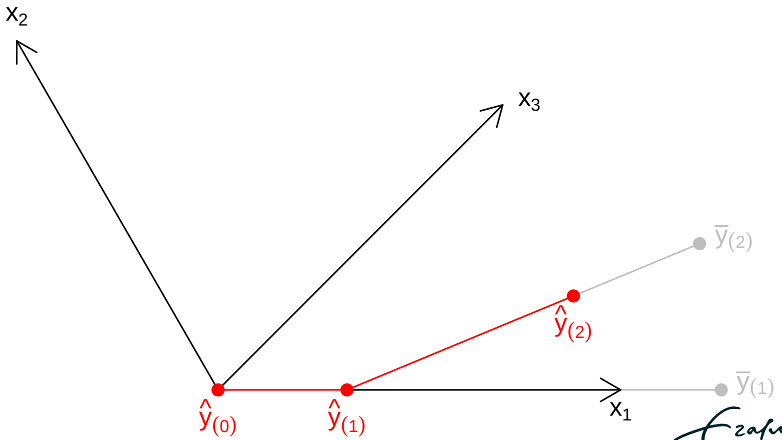
Least angle regression (LARS): Idea



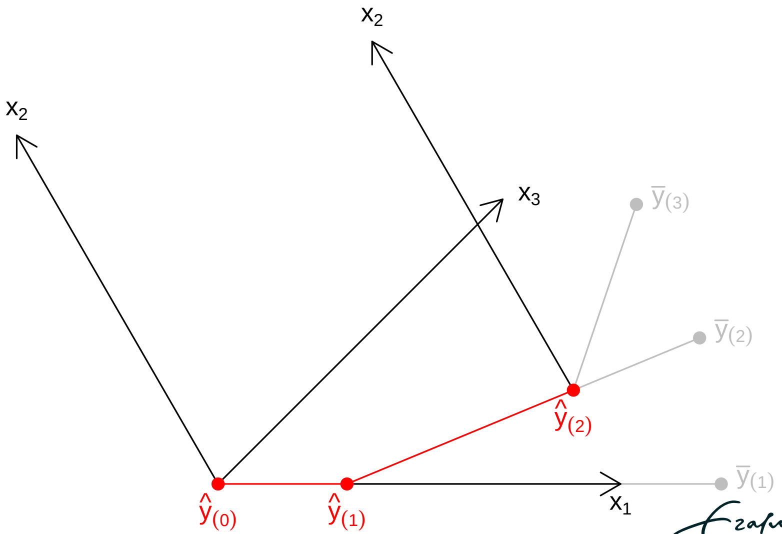
Least angle regression (LARS): Idea



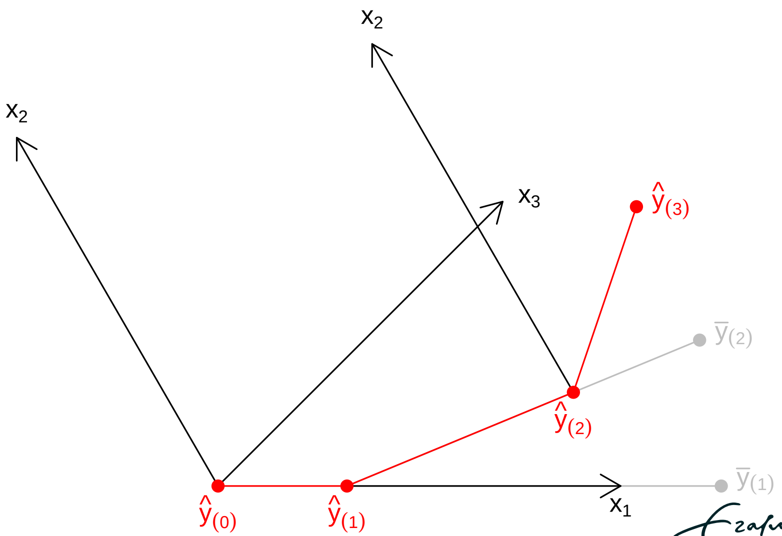
Least angle regression (LARS): Idea



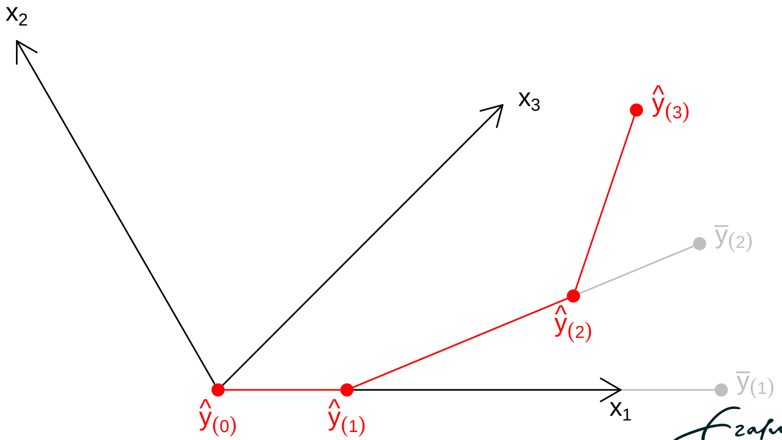
Least angle regression (LARS): Idea



Least angle regression (LARS): Idea



Least angle regression (LARS): Idea



Least angle regression (LARS): Algorithm

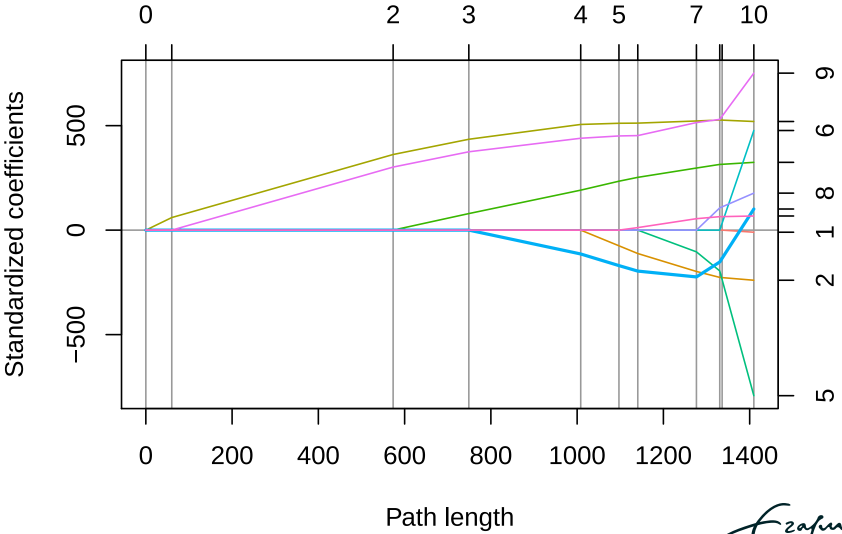
Model selection algorithm based on forward selection approach (Efron et al., 2004)

- Start with most correlated predictor
- Move along equi-angular vector until a new predictor is equally correlated and add that predictor to the active set
- Update coefficients of active predictors along solution path

Least angle regression (LARS): Properties

- Simple formula for the step size when next predictor is added
- Solution path is piecewise linear
  $\longrightarrow$ Efficient computation
- Applicable to high-dimensional data by limiting the number of steps

LARS: piecewise linear solution path vs path length



Least absolute shrinkage and selection operator (lasso)

Different parametrization than proposed by Tibshirani (1996):

$$\hat{\boldsymbol{\beta}}_{\text{lasso}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (y_i - \mathbf{x}_i' \boldsymbol{\beta})^2 + n\lambda \|\boldsymbol{\beta}\|_1$$

$\longrightarrow$ Can be computed through LARS framework (Efron et al., 2004)

$\longrightarrow$ But modern implementations use a coordinate descent algorithm (Friedman et al., 2010) or an ADMM algorithm (Boyd et al., 2010)

Least absolute shrinkage and selection operator (lasso)

Different parametrization than proposed by Tibshirani (1996):

$$\hat{\beta}_{\text{lasso}} = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - \mathbf{x}_i'\beta)^2 + n\lambda\|\beta\|_1$$

$\longrightarrow$ Can be computed through LARS framework (Efron et al., 2004)

$\longrightarrow$ But modern implementations use a coordinate descent algorithm (Friedman et al., 2010) or an ADMM algorithm (Boyd et al., 2010)

Relationship between LARS and lasso

Modification of the LARS algorithm:

- If the coefficient of an active predictor reaches 0, drop that predictor from the active set
- Continue algorithm with reduced active set

$\longrightarrow$ Lasso solution path

$\longrightarrow$ If no coefficient changes signs, LARS solution path is identical to lasso solution path

Relationship between LARS and lasso

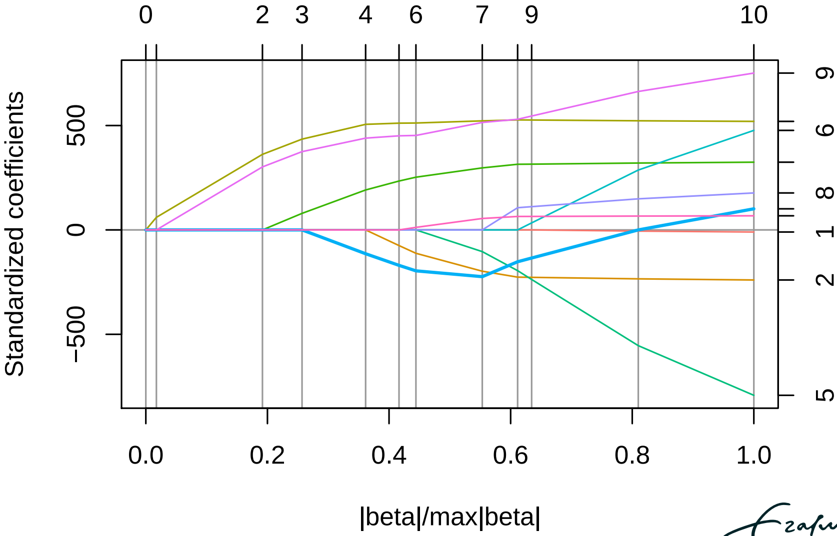
Modification of the LARS algorithm:

- If the coefficient of an active predictor reaches 0, drop that predictor from the active set
- Continue algorithm with reduced active set

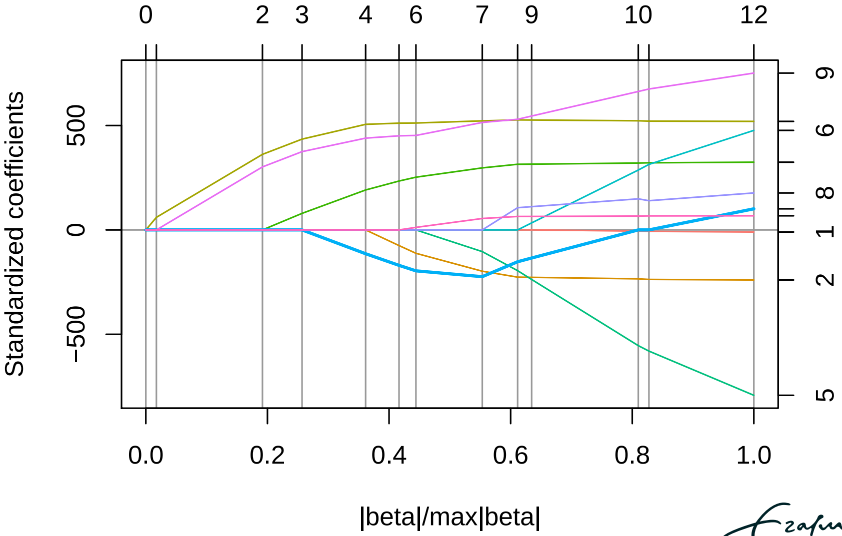
$\longrightarrow$ Lasso solution path

$\longrightarrow$ If no coefficient changes signs, LARS solution path is identical to lasso solution path

LARS: piecewise linear solution path vs L_1 norm



Lasso: piecewise linear solution path vs L_1 norm



Robust least angle regression

Robust least angle regression (RLARS)

Hybrid procedure (Khan et al., 2007):

1. Sequence predictors based on robust correlations
2. Fit robust regression models along the sequence

⟶ Applicable to high-dimensional data by limiting the number of steps

⟶ Implemented in function `rlars()` of R package `robustHD`

Robust least angle regression (RLARS)

Hybrid procedure (Khan et al., 2007):

1. Sequence predictors based on robust correlations
2. Fit robust regression models along the sequence

$\longrightarrow$ Applicable to high-dimensional data by limiting the number of steps

$\longrightarrow$ Implemented in function `rlars()` of R package `robustHD`

Robust groupwise least angle regression (RGrpLARS)

Robust extension of LARS to groupwise variable selection (Alfons et al., 2016):

1. Sequence of predictor groups based on R^2 after initial data cleaning
2. Fit robust regression models along the sequence using original data

⟶ Applicable to high-dimensional data for some of the proposed data cleaning approaches

⟶ Implemented in function `rgrplars()` of R package `robustHD`

Robust groupwise least angle regression (RGrpLARS)

Robust extension of LARS to groupwise variable selection (Alfons et al., 2016):

1. Sequence of predictor groups based on R^2 after initial data cleaning
2. Fit robust regression models along the sequence using original data

$\longrightarrow$ Applicable to high-dimensional data for some of the proposed data cleaning approaches

$\longrightarrow$ Implemented in function `rgrplars()` of R package `robustHD`

Sparse least trimmed squares (trimmed lasso)

Sparse least trimmed squares regression

Objective function:

$$\hat{\beta}_{\mathrm{sparseLTS}} = \arg\min_{\beta} \sum_{i=1}^{h} (\boldsymbol{r}^2(\beta))_{i:n} + h\lambda \|\beta\|_1$$

with

$h \leq n$	
$\boldsymbol{r}^2(\beta) = (r_1^2, \ldots, r_n^2)'$	squared residuals
$(\boldsymbol{r}^2(\beta))_{1:n} \leq \ldots \leq (\boldsymbol{r}^2(\beta))_{n:n}$	order statistics

Sparse least trimmed squares regression

Combining...

- Least trimmed squares regression for robustness (Rousseeuw and Van Driessen, 2006)
- Lasso for sparsity (Tibshirani, 1996)

$\longrightarrow$ C-step algorithm for computation

$\longrightarrow$ Reweighting step to increase efficiency

$\longrightarrow$ Details and theory in Alfons et al. (2013) and Öllerer et al. (2015)

Sparse least trimmed squares regression

Combining...

- Least trimmed squares regression for robustness (Rousseeuw and Van Driessen, 2006)
- Lasso for sparsity (Tibshirani, 1996)

$\longrightarrow$ C-step algorithm for computation

$\longrightarrow$ Reweighting step to increase efficiency

$\longrightarrow$ Details and theory in Alfons et al. (2013) and Öllerer et al. (2015)

Sparse least trimmed squares regression

Combining...

- Least trimmed squares regression for robustness (Rousseeuw and Van Driessen, 2006)
- Lasso for sparsity (Tibshirani, 1996)

$\longrightarrow$ C-step algorithm for computation

$\longrightarrow$ Reweighting step to increase efficiency

$\longrightarrow$ Details and theory in Alfons et al. (2013) and Öllerer et al. (2015)

C-step

Objective function in terms of subset H:

$$Q(H, \beta) = \sum_{i \in H} (y_i - \mathbf{x}_i' \beta)^2 + h\lambda \|\beta\|_1$$

Step k with current subset H_k:

- Obtain lasso solution $\hat{\beta}_{H_k} = \arg\min_\beta Q(H_k, \beta)$
- Compute squared residuals $\boldsymbol{r}_k^2 = (r_{k,1}^2, \dots, r_{k,n}^2)'$
- Construct H_{k+1} from observations with smallest squared residuals:

$$H_{k+1} = \left\{ i \in \{1, \dots, n\} : r_{k,i}^2 \in \{(\boldsymbol{r}_k^2)_{j:n} : j = 1, \dots, h\} \right\}$$

C-step

Objective function in terms of subset H:

$$Q(H, \beta) = \sum_{i \in H} (y_i - \mathbf{x}_i' \beta)^2 + h\lambda \|\beta\|_1$$

Step k with current subset H_k:

- Obtain lasso solution $\hat{\beta}_{H_k} = \arg\min_\beta Q(H_k, \beta)$
- Compute squared residuals $\boldsymbol{r}_k^2 = (r_{k,1}^2, \ldots, r_{k,n}^2)'$
- Construct H_{k+1} from observations with smallest squared residuals:

$$H_{k+1} = \left\{ i \in \{1, \ldots, n\} : r_{k,i}^2 \in \{(\boldsymbol{r}_k^2)_{j:n} : j = 1, \ldots, h\} \right\}$$

Basic C-step algorithm

1. Obtain m initial subsets of size h from elementary 3-subsets
2. For $j = 1, \ldots, m$ do C-steps until convergence
3. Return $\hat{\beta}_{\mathrm{sparseLTS}}$ corresponding to the subset with the lowest value of the objective function

$\longrightarrow$ Improvements as in FAST-LTS algorithm

$\longrightarrow$ Implemented in function `sparseLTS()` of R package `robustHD`

Basic C-step algorithm

1. Obtain m initial subsets of size h from elementary 3-subsets
2. For $j = 1, \ldots, m$ do C-steps until convergence
3. Return $\hat{\beta}_{\mathrm{sparseLTS}}$ corresponding to the subset with the lowest value of the objective function

$\longrightarrow$ Improvements as in FAST-LTS algorithm

$\longrightarrow$ Implemented in function `sparseLTS()` of R package `robustHD`

Reweighted estimator

Weights from outlier detection via raw estimator:

$$w_i = \begin{cases} 1 & \text{if } |(r_i - \hat{\mu}_{\text{raw}})/\hat{\sigma}_{\text{raw}}| \leq \Phi^{-1}(1-\delta) \\ 0 & \text{if } |(r_i - \hat{\mu}_{\text{raw}})/\hat{\sigma}_{\text{raw}}| > \Phi^{-1}(1-\delta) \end{cases} \qquad i = 1, \ldots, n$$

$\longrightarrow$ Reweighted estimator given by weighted lasso fit

$$\hat{\beta}_{\text{reweighted}} = \arg\min_{\beta} \sum_{i=1}^{n} w_i (y_i - \mathbf{x}_i' \beta)^2 + \lambda n_w \|\beta\|_1$$

with

$n_w = \sum_{i=1}^{n} w_i$ number of detected good data points

Reweighted estimator

Weights from outlier detection via raw estimator:

$$w_i = \begin{cases} 1 & \text{if } |(r_i - \hat{\mu}_{\text{raw}})/\hat{\sigma}_{\text{raw}}| \leq \Phi^{-1}(1-\delta) \\ 0 & \text{if } |(r_i - \hat{\mu}_{\text{raw}})/\hat{\sigma}_{\text{raw}}| > \Phi^{-1}(1-\delta) \end{cases} \qquad i = 1, \ldots, n$$

$\longrightarrow$ Reweighted estimator given by weighted lasso fit

$$\hat{\boldsymbol{\beta}}_{\text{reweighted}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} w_i (y_i - \mathbf{x}_i' \boldsymbol{\beta})^2 + \lambda n_w \|\boldsymbol{\beta}\|_1$$

with

$n_w = \sum_{i=1}^{n} w_i$ number of detected good data points

Breakdown point

Finite sample breakdown point (FBP):

$$\varepsilon^*(\hat{\beta}; \boldsymbol{Z}) = \min \left\{ \frac{m}{n} : \sup_{\tilde{\boldsymbol{Z}}} \|\hat{\beta}(\tilde{\boldsymbol{Z}})\|_2 = \infty \right\}$$

with

$\boldsymbol{Z} = (\boldsymbol{X}, \boldsymbol{y})$	original sample
$\tilde{\boldsymbol{Z}}$	contaminated sample with m points replaced by arbitrary values

Breakdown point theorem

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{h} (\boldsymbol{\rho}(\boldsymbol{y} - \boldsymbol{X}\beta))_{i:n} + h\lambda\|\beta\|_1$$

where

$$\begin{array}{ll} h \le n & \\ \rho(x) \quad \text{convex, symmetric, } \rho(0) = 0 \text{ and } \rho(x) > 0 \text{ for } x \neq 0 & \\ \boldsymbol{\rho}(\boldsymbol{y} - \boldsymbol{X}\beta) := (\rho(y_1 - \mathbf{x}_1\beta), \ldots, \rho(y_n - \mathbf{x}_n\beta))' & \text{losses} \\ (\boldsymbol{\rho}(\boldsymbol{y} - \boldsymbol{X}\beta)))_{1:n} \le \ldots \le (\boldsymbol{\rho}(\boldsymbol{y} - \boldsymbol{X}\beta))_{n:n} & \text{order statistics} \end{array}$$

$\longrightarrow$ Breakdown point of the estimator $\hat{\beta}$:

$$\varepsilon^*(\hat{\beta}; \boldsymbol{Z}) = \frac{n - h + 1}{n}$$

Breakdown point of selected estimators

Sparse LTS:

$$\varepsilon^*(\hat{\beta}_{\mathrm{sparseLTS}}; \boldsymbol{Z}) = \frac{n - h + 1}{n}$$

Lasso:

$$\varepsilon^*(\hat{\beta}_{\mathrm{lasso}}; \boldsymbol{Z}) = \frac{1}{n}$$

$\longrightarrow$ Note: Breakdown point does not depend on dimension p

Breakdown point of selected estimators

Sparse LTS:

$$\varepsilon^*(\hat{\beta}_{\mathrm{sparseLTS}}; \boldsymbol{Z}) = \frac{n - h + 1}{n}$$

Lasso:

$$\varepsilon^*(\hat{\beta}_{\mathrm{lasso}}; \boldsymbol{Z}) = \frac{1}{n}$$

$\longrightarrow$ Note: Breakdown point does not depend on dimension p

Questions

1. For a small enough h (i.e., a large enough trimming proportion), the sparse LTS has a breakdown point larger than 50%.

   $\longrightarrow$ How is this possible?

   $\longrightarrow$ Does this make sense from the perspective of robust statistics?

2. The lasso is equivalent to the constrained optimization problem

$$\min_{\beta} \sum_{i=1}^{n} (y_i - \mathbf{x}_i' \beta)^2 \qquad \text{subject to} \qquad \|\beta\|_1 \leq t$$

   We also have equivalence of the L_1 and L_2 norms.

   $\longrightarrow$ How is it possible that the breakdown point of the lasso is 0%?

Penalized S- and MM-estimator

Penalized Elastic Net S-Estimator (PENSE)

Objective function:

$$\hat{\beta}_{\text{PENSE}} = \arg\min_{\beta} \hat{\sigma}^2(\beta) + \lambda_S \left(\alpha \|\beta\|_1 + \frac{1-\alpha}{2} \|\beta\|_2^2 \right)$$

$$\text{with} \quad \frac{1}{n} \sum_{i=1}^{n} \rho \left(\frac{r_i(\beta)}{\hat{\sigma}(\beta)} \right) = b$$

where

$r_i(\beta) = y_i - \mathbf{x}_i'\beta$	residuals
$b = \mathbb{E}_Z[\rho(Z)]$ with $Z \sim \mathcal{N}(0,1)$	consistency parameter

Penalized Elastic Net S-Estimator (PENSE)

Combining...

- S-estimator for robustness (Salibian-Barrera and Yohai, 2006)
- Elastic net for regularization and sparsity (Friedman et al., 2005)

⟶ Iteratively reweighted elastic net (IRWEN) algorithm based on initial estimator for computation

⟶ Implemented in function `pense()` of R package `pense`

⟶ Robust, but not efficient

⟶ Details and theory in Cohen Freue et al. (2019)

Penalized Elastic Net S-Estimator (PENSE)

Combining...

- S-estimator for robustness (Salibian-Barrera and Yohai, 2006)
- Elastic net for regularization and sparsity (Friedman et al., 2005)

$\longrightarrow$ Iteratively reweighted elastic net (IRWEN) algorithm based on initial estimator for computation

$\longrightarrow$ Implemented in function `pense()` of R package `pense`

$\longrightarrow$ Robust, but not efficient

$\longrightarrow$ Details and theory in Cohen Freue et al. (2019)

Penalized Elastic Net S-Estimator (PENSE)

Combining...

- S-estimator for robustness (Salibian-Barrera and Yohai, 2006)
- Elastic net for regularization and sparsity (Friedman et al., 2005)

$\longrightarrow$ Iteratively reweighted elastic net (IRWEN) algorithm based on initial estimator for computation

$\longrightarrow$ Implemented in function `pense()` of R package `pense`

$\longrightarrow$ Robust, but not efficient

$\longrightarrow$ Details and theory in Cohen Freue et al. (2019)

PENSE refined via M-estimator (PENSEM)

$\longrightarrow$ To increase efficiency, Cohen Freue et al. (2019) propose a penalized elastic net M-estimator, using the initial scale estimate from PENSE

$\longrightarrow$ However, this creates other issues and this approach will not be discussed further

PENSE refined via M-estimator (PENSEM)

$\longrightarrow$ To increase efficiency, Cohen Freue et al. (2019) propose a penalized elastic net M-estimator, using the initial scale estimate from PENSE

$\longrightarrow$ However, this creates other issues and this approach will not be discussed further

Hands-on part with R

R packages and script

We will use:

⟶ R packages `robustHD` (version 0.7.0!) and `pense`

```
R> install.packages(c("robustHD", "pense"))
```

⟶ R script `ICORS2021_workshop_script.R` available from `https://personal.eur.nl/alfons/ICORS2021.html`

⟶ Run the commands in your own R session along with me

NCI-60 cancer cell panel

- Data on 60 human cancer cell lines
- Available from `http://discover.nci.nih.gov/cellminer/`
- Protein expressions based on 162 antibodies
- Gene expression data with $p = 22\,283$

$\longrightarrow$ $n = 59$: one observation with all gene expressions missing

$\longrightarrow$ Use protein expression with largest MAD as response variable

$\longrightarrow$ Candidate predictors: $d = 100$ most correlated gene expressions

NCI-60 cancer cell panel

- Data on 60 human cancer cell lines
- Available from `http://discover.nci.nih.gov/cellminer/`
- Protein expressions based on 162 antibodies
- Gene expression data with $p = 22\,283$

  $\longrightarrow$ $n = 59$: one observation with all gene expressions missing

$\longrightarrow$ Use protein expression with largest MAD as response variable

$\longrightarrow$ Candidate predictors: $d = 100$ most correlated gene expressions

Discussion and conclusions

Some issues to look out for

- Residual scale is typically underestimated in high-dimensions

  $\longrightarrow$ Outlier detection via standardized residuals is prone to false positives

- BIC for regularization parameter selection can be unstable for values of λ close to 0 due to exact fit situations

  $\longrightarrow$ Cross-validation is preferred, but computationally expensive

Conclusions

$\longrightarrow$ Robust regression in high dimensions remains a challenging problem

$\longrightarrow$ R packages `robustHD` and `pense` provide promising functionality

$\longrightarrow$ A trimmed version of the elastic net (Kurnaz et al., 2017) is available in R package `enetLTS`, also for logistic regression

References I

Alfons, A., Croux, C., and Gelper, S. (2013). Sparse least trimmed squares regression for analyzing high-dimensional large data sets. The Annals of Applied Statistics, 7(1):226–248.

Alfons, A., Croux, C., and Gelper, S. (2016). Robust groupwise least angle regression. Computational Statistics & Data Analysis, 93:421–435.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2010). Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning, 3(1):1–122.

Cohen Freue, G., Kepplinger, D., Salibian-Barrera, M., and Smucler, E. (2019). Robust elastic net estimators for variable selection and identification of proteomic biomarkers. The Annals of Applied Statistics, 13(4):2065–2090.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. The Annals of Statistics, 32(2):407–499.

Friedman, J., Hastie, T., and Tibshirani, R. (2005). Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society, Series B, 67(2):301–320.

References II

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software, 33(1):1–22.

Khan, J., Van Aelst, S., and Zamar, R. (2007). Robust linear model selection based on least angle regression. Journal of the American Statistical Association, 102(480):1289–1299.

Kurnaz, F., Hoffmann, I., and Filzmoser, P. (2017). Robust and sparse estimation methods for high dimensional linear and logistic regression. Chemometrics and Intelligent Laboratory Systems, 172:211–222.

Öllerer, V., Croux, C., and Alfons, A. (2015). The influence function of penalized regression estimators. Statistics: A Journal of Theoretical and Applied Statistics, 49(4):741–765.

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

Salibian-Barrera, M. and Yohai, V. (2006). A fast algorithm for S-regression estimates. Journal of Computational and Graphical Statistics, 15(2):414–427.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B, 58(1):267–288.