

Algorithms for LTS regression

October 26, 2009

Outline

- ▶ Robust regression.
- ▶ LTS regression.
- ▶ Adding row algorithm.
 - ▶ Branch and bound algorithm (BBA).
 - ▶ Preordering BBA.
- ▶ Structured problems
 - ▶ Generalized linear model.
 - ▶ Seemingly unrelated regressions.
- ▶ Conclusions.

Ordinary least squares (OLS)

- ▶ Ordinary linear model (OLM):

$$y = A\beta + \varepsilon,$$

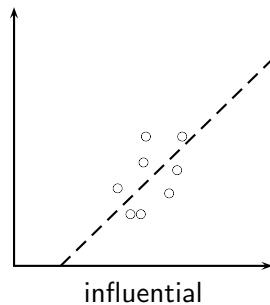
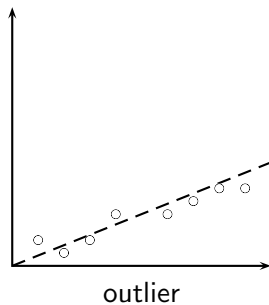
where $y \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$, $\beta \in \mathbb{R}^n$ and $\varepsilon \in \mathbb{R}^m$.

- ▶ Objective function:

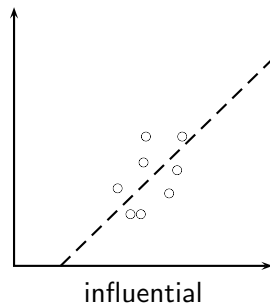
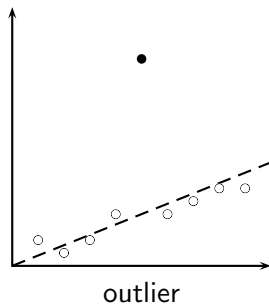
$$\text{RSS}(\beta) = \sum_{i=1}^n \varepsilon_i^2.$$

- ▶ OLS estimation is sensitive to outliers.

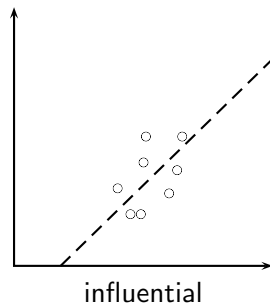
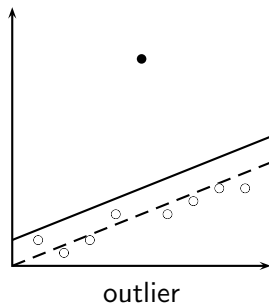
Outliers



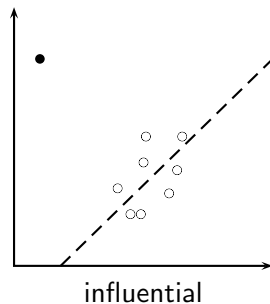
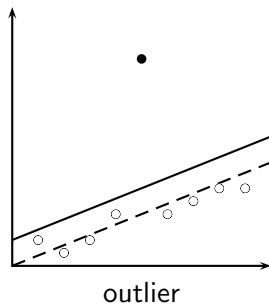
Outliers



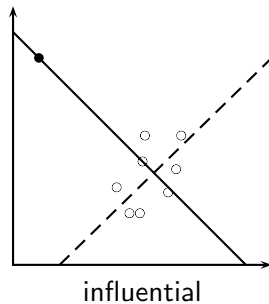
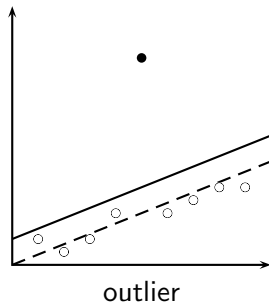
Outliers



Outliers



Outliers



Robust regression

- ▶ **Breakdown point**: smallest fraction of contamination that can bias the estimator arbitrarily.
- ▶ OLS: one observation is enough to contaminate the estimator; i.e. breakdown point is $1/n \rightarrow 0$.
- ▶ **High-breakdown** estimators can resist contamination of nearly 50% of the data.
- ▶ Known algorithms: least median of squares (LMS), least trimmed squares (**LTS**).

LTS regression

- ▶ Objective function:

$$\text{RSS}_h(\beta) = \sum_{i=1}^h (\varepsilon^2)_{[i]},$$

where the **coverage** h may lie between $m/2$ and m .

- ▶ Equivalent to finding the h -subset with optimal LS objective function.
- ▶ Naive algorithm: enumerate all subsets.
- ▶ Number of h -subsets: $\binom{m}{h} = \frac{m!}{h!(m-h)!}$.
- ▶ Computational load is prohibitive for $m > 30$.

LTS regression

- ▶ Approximate algorithms: PROGRESS (Rousseeuw and Leroy, 1987), FSA (Hawkins, 1994), **FAST-LTS** (Rousseeuw and Van Driessen, 2006).
- ▶ Exact algorithm: branch and bound (Agulló, 2001).

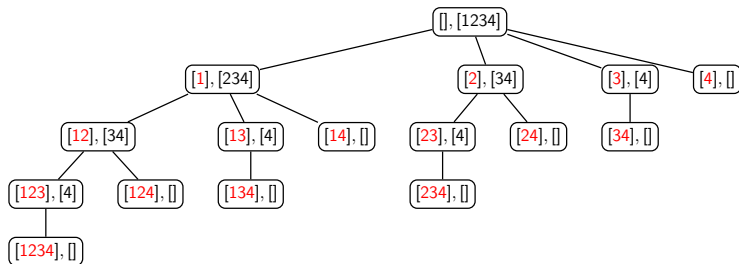
Issues

- ▶ In practice, coverage $h = 50\%$ is used.
- ▶ But: contamination rarely exceeds 15%–20% of data.
- ▶ High-breakdown seems like overkill: large portions of good data are ignored.
- ▶ Problem: find a good tradeoff between **robustness** (i.e. “discard all bad data”) and **efficiency** (i.e. “include all good data”).
- ▶ **Choice of the unknown coverage parameter h ?**

Adding row algorithm (ARA)

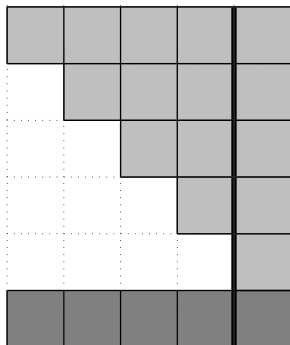
- ▶ Strong correspondence between row-selection techniques and procedures for computing all-variable-subsets regression.
- ▶ Computes the all-observation-subsets regression for a **range of coverage values** $[h_{\min}, h_{\max}]$.
- ▶ The organization of the algorithm is determined by the **all-subsets tree**.
- ▶ A node corresponds to an observation-subset model.
- ▶ The model is represented by the upper-triangular factor of the QR decomposition of (A, y) .
- ▶ A **Cholesky updating** algorithm is employed to move from one node to another.

Regression tree

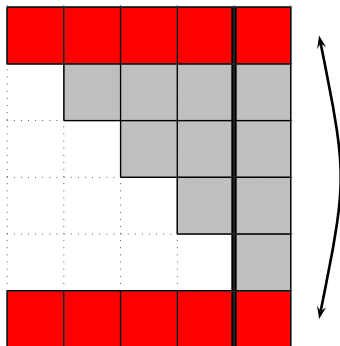


- ▶ Number of observations: $m = 4$.
- ▶ Number of nodes: 2^m .

Cholesky update

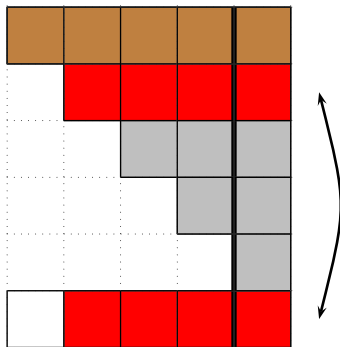
 (R, z) (x^T, y)

Cholesky update



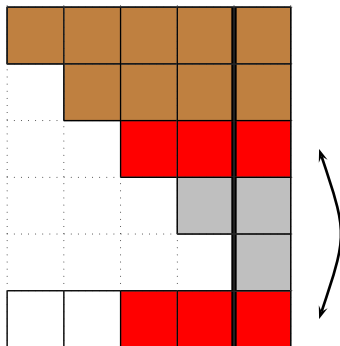
Givens rotation: G_1

Cholesky update



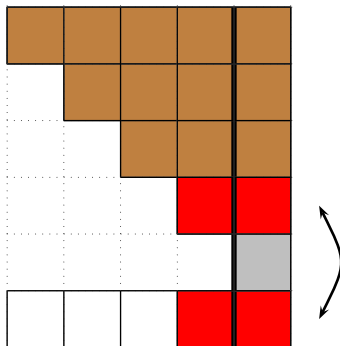
Givens rotation: G_2

Cholesky update



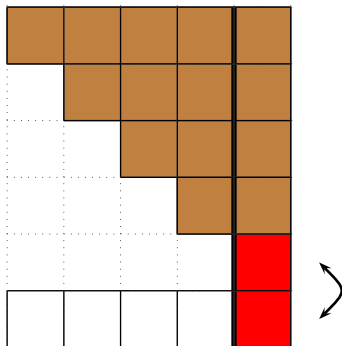
Givens rotation: G_3

Cholesky update



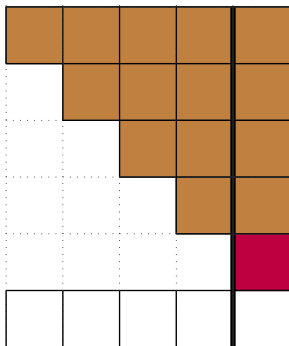
Givens rotation: G_4

Cholesky update



Givens rotation: G_5

Cholesky update



The RSS.

Branch and bound algorithm (BBA)

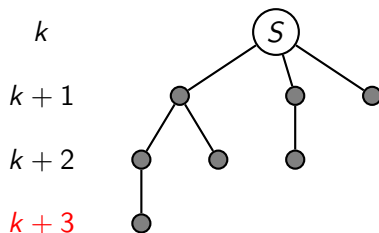
- ▶ ARA is prohibitive even for a moderate number of observations.
- ▶ Fundamental property: given two subsets of observations S_A and S_B ,

$$\text{RSS}(S_A) \leq \text{RSS}(S_B) \quad \text{if} \quad S_A \subset S_B,$$

where $\text{RSS}(S_Z)$ denotes the residual sum of squares of the LS estimator of the model S_Z .

- ▶ In other words: updating the model by one observation increases the RSS.
- ▶ Can be used to restrict the number of evaluated subsets (reduce search space i.e. number of generated nodes).

Cutting test



- ▶ Residual lookup table:

$$\dots < \rho_k < \rho_{k+1} < \rho_{k+2} < \rho_{k+3} < \dots$$

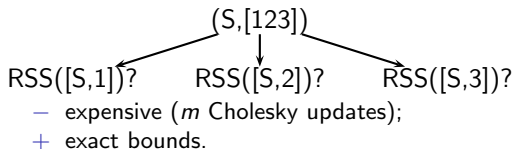
- ▶ Cutting test:

$$b_S = \text{RSS}(S) > \rho_{k+3},$$

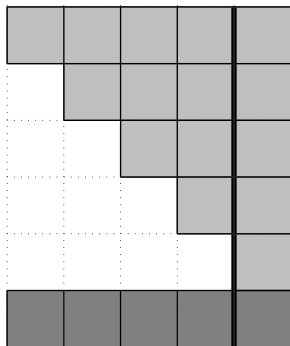
where S is a set of k observations and b_S the node bound.

Observation preordering

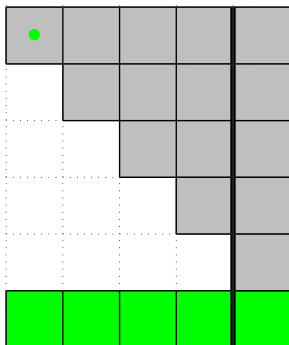
- ▶ BBA: computational efficiency rises when more nodes are cut i.e. when bigger subtrees are bounded with bigger values.
- ▶ Strategies: sort observations in decreasing order of
 1. absolute LS residuals: estimate the observation-subset model and determine the residuals $\hat{\varepsilon}$
 - + cheap (solve one linear system);
 - approximate bounds;
 2. partial increment in RSS:



Computing the bound

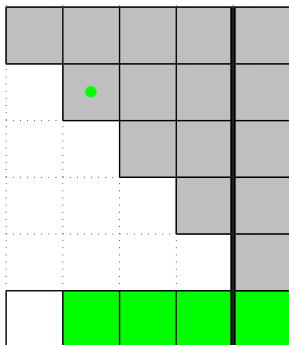
 (R, z) (x^T, y)

Computing the bound



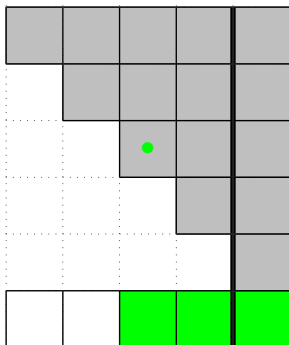
Givens rotation: G_1

Computing the bound



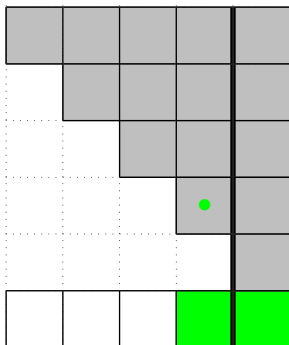
Givens rotation: G_2

Computing the bound



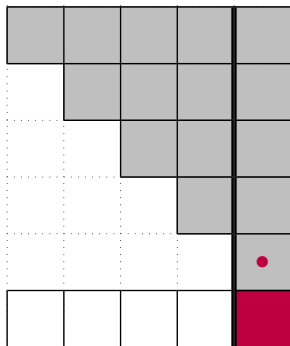
Givens rotation: G_3

Computing the bound



Givens rotation: G_4

Computing the bound



The RSS!

Computing the bound

Benefits:

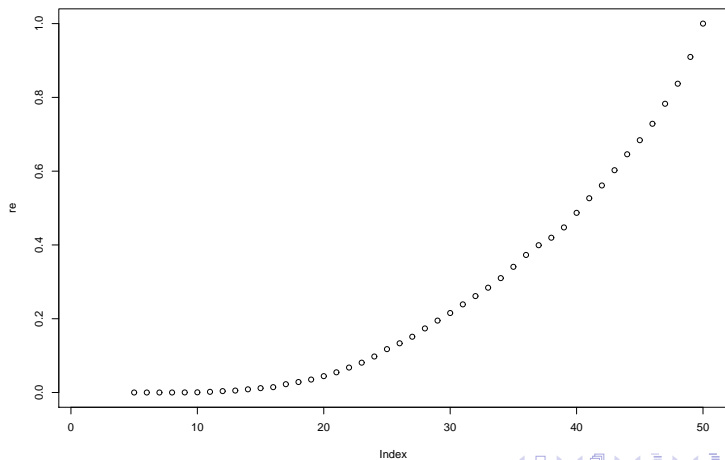
- ▶ the upper-triangular factor is not modified;
- ▶ half the computational cost;
- ▶ avoids matrix-copy operations.

Example 1

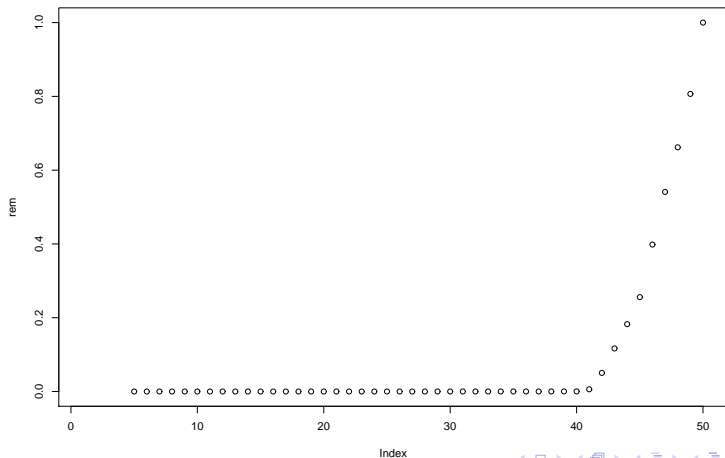
- ▶ Dataset:
 - ▶ random data matrix X ($m = 50, n = 4$);
 - ▶ fixed coefficients β ;
 - ▶ $y = X\beta + \varepsilon$.
- ▶ Outliers: construct modified model (X_m, y_m) from (X, y) by replacing ten observations with random data.
- ▶ Compute LTS estimates and determine relative efficiencies:

$$\text{RE}(h) = \frac{\sigma_h^2}{\sigma_{\text{LS}}^2}, \quad \text{where } h = h_{\min}, \dots, h_{\max}.$$

Relative efficiencies: original model (X, y)



Relative efficiencies: contaminated model (X_m, y_m)



Example 2

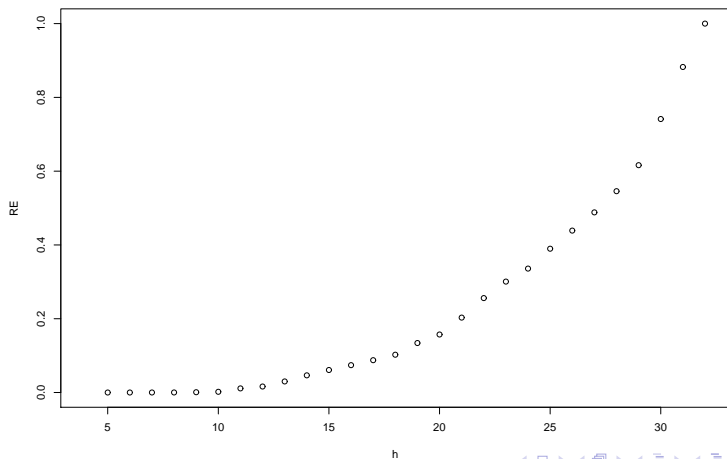
- ▶ Dataset: randomly generated according to

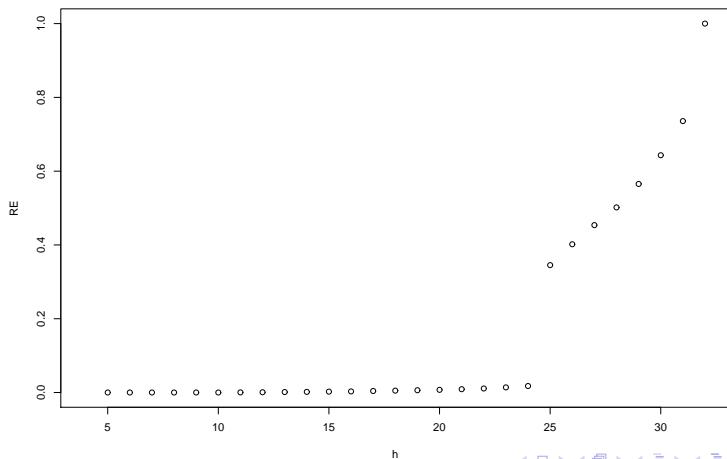
$$y = x_{i,1} + x_{i,2} + \dots + x_{i,n-1} + 1 + e_i,$$

where $e_i \sim N(0, 1)$ is the error term and $x_{i,j} \sim N(0, 100)$ are the explanatory variables.

- ▶ Outliers: replace some of the $x_{i,1}$ by values that are normally distributed with mean 100 and variance 100.

Relative efficiencies: original model (X, y)



Relative efficiencies: contaminated model (X_m, y_m)

Generalized least squares (GLS)

- ▶ The general linear model (GLM) is given by

$$y = X\beta + \varepsilon, \quad \varepsilon \sim (0, \sigma^2\Omega)$$

where $y \in \mathbb{R}^m$, $X \in \mathbb{R}^{m \times n}$, $\beta \in \mathbb{R}^n$ and $\Omega \in \mathbb{R}^{m \times m}$ ($m \geq n$).

- ▶ Ω is assumed to be of full rank.
- ▶ The BLUE of β minimizes the objective function

$$\|X\beta - y\|_{\Omega^{-1}}^2 = (X\beta - y)^T \Omega^{-1} (X\beta - y).$$

Generalized linear least squares problem (GLLSP)

- ▶ The GLS is reformulated as

$$\hat{\beta}, \hat{u} = \underset{\beta, u}{\operatorname{argmin}} \|u\|^2 \quad \text{subject to} \quad y = X\beta + Bu,$$

where $B \in \mathbb{R}^{m \times p}$ such that $\Omega = BB^T$ and $u \sim (0, \sigma^2 I_m)$
($p \geq m - n$).

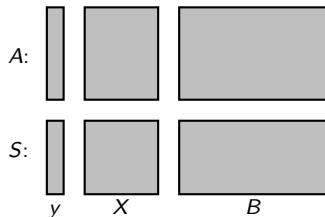
- ▶ Ω may be singular.
- ▶ Residual sum of squares (RSS):

$$\operatorname{RSS}(\hat{\beta}) = \|\hat{u}\|^2.$$

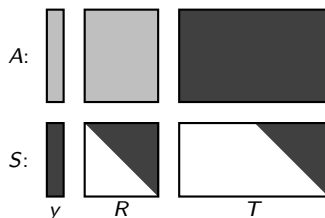
- ▶ Computational tool: generalized QR decomposition (GQRD) of X and B .

Solving the GLLSP

- ▶ In a node on level n of the ARA tree:

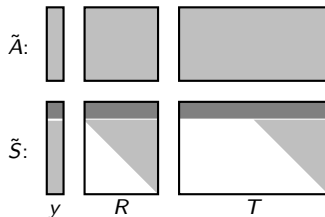


- ▶ Compute GQRD and transform GLLSP:

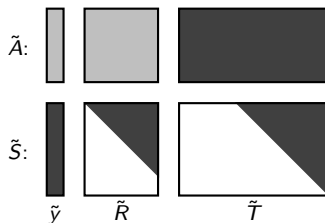


Updating the GLLSP

- ▶ For each new node:

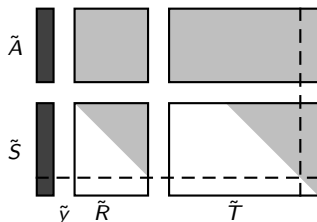


- ▶ Update GQRD and transform GLLSP:



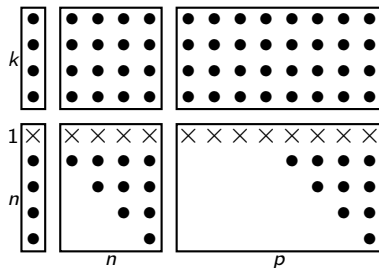
Updating the GLLSP

- ▶ Reduce the GLLSP:



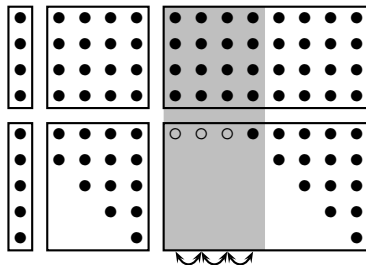
Givens sequence

- Update the GQRD by the use of Givens rotations.



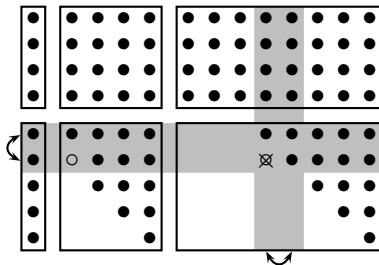
Givens sequence

- Update the GQRD by the use of Givens rotations.



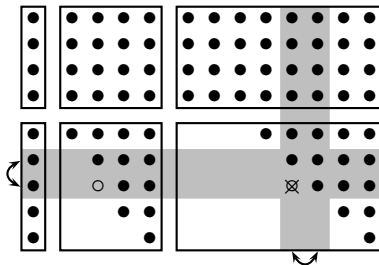
Givens sequence

- Update the GQRD by the use of Givens rotations.



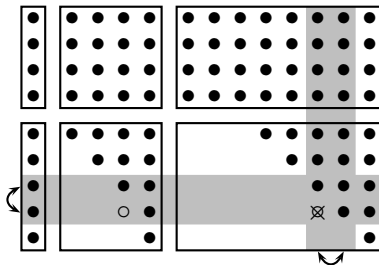
Givens sequence

- Update the GQRD by the use of Givens rotations.



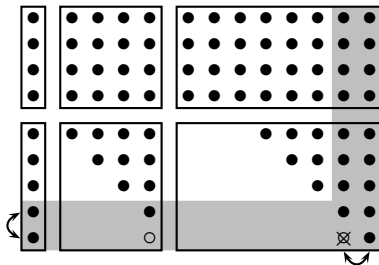
Givens sequence

- Update the GQRD by the use of Givens rotations.



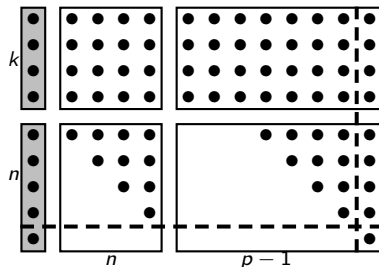
Givens sequence

- Update the GQRD by the use of Givens rotations.



Givens sequence

- Update the GQRD by the use of Givens rotations.



Seemingly unrelated regressions (SUR)

- ▶ G regressions:

$$y^{(i)} = X^{(i)}\beta^{(i)} + \varepsilon^{(i)}, \quad i = 1, \dots, G,$$

where $y^{(i)} \in \mathbb{R}^m$, $X^{(i)} \in \mathbb{R}^{m \times n_i}$ ($m \geq n_i$), $\beta^{(i)} \in \mathbb{R}^{n_i}$, $\varepsilon^{(i)} \in \mathbb{R}^m$ and $E(\varepsilon_k^{(i)}) = 0$.

- ▶ Contemporaneous disturbances are correlated:

$$\text{Var}(\varepsilon_t^{(i)}, \varepsilon_t^{(j)}) = \sigma_{ij} \quad \text{and} \quad \text{Var}(\varepsilon_s^{(i)}, \varepsilon_t^{(j)}) = 0 \quad \text{if} \quad s \neq t.$$

- ▶ Compact form:

$$\text{vec}(Y) = \bigoplus_{i=1}^G \left(X^{(i)} \right) \text{vec}(\{\beta^{(i)}\}_G) + \text{vec}(E),$$

where $Y = [y^{(1)} \quad \dots \quad y^{(G)}] \in \mathbb{R}^{m \times G}$, $E = [\varepsilon^{(1)} \quad \dots \quad \varepsilon^{(G)}] \in \mathbb{R}^{m \times G}$, $\text{vec}(E) \sim (0, \Sigma \otimes I_m)$ and $\Sigma = [\sigma_{ij}] \in \mathbb{R}^{G \times G}$.

GLLSP of the SUR model

- SUR-GLLSP: $\{\hat{\beta}^{(i)}\}_G, \hat{U} = \operatorname{argmin}_{\{\beta^{(i)}\}_G, U} \|U\|_F$ subject to

$$\operatorname{vec}(Y) = \left(\bigoplus_{i=1}^G X^{(i)} \right) \operatorname{vec}(\{\beta^{(i)}\}_G) + K \operatorname{vec}(U),$$

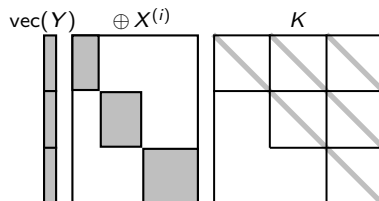
where $E = UC^T$, $C \in \mathbb{R}^{G \times G}$ is upper triangular such that $CC^T = \Sigma$, $K = C \otimes I_m$, $\operatorname{vec}(U) \sim (0, \sigma^2 I_M)$ and $M = Gm$.

- Objective function:

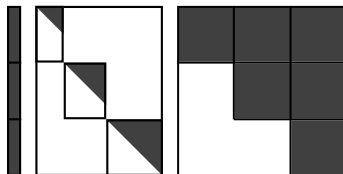
$$\operatorname{RSS}(\{\hat{\beta}^{(i)}\}_G) = \|\hat{U}\|_F.$$

Solving the SUR-GLLSP

- ▶ On level n_G of the subsets tree:

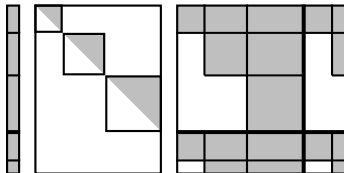


- ▶ Orthogonal transformation Q^T from the left:

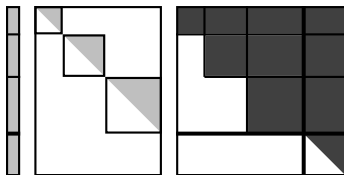


Solving the SUR-GLLSP

- ▶ Permute rows and columns:

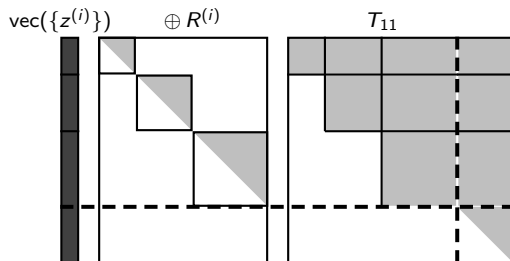


- ▶ Orthogonal transformation P^T from the right:



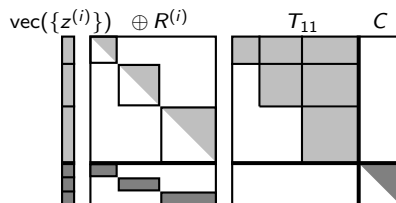
Solving the SUR-GLLSP

- ▶ Reduce SUR-GLLSP:

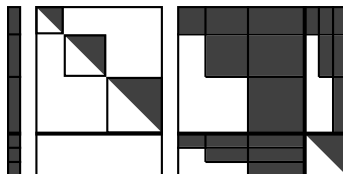


Updating the SUR-GLLSP

- ▶ For each new new node:

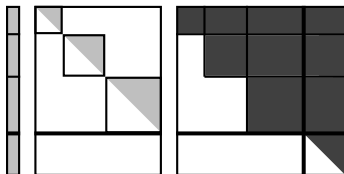


- ▶ Apply orthogonal transformation \tilde{Q}^T from the left:

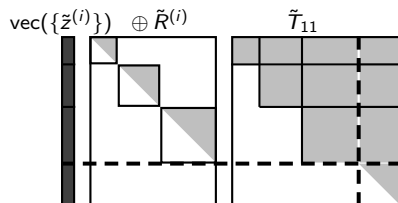


Updating the SUR-GLLSP

- ▶ Apply orthogonal transformation \tilde{P}^T from the right:



- ▶ Reduce SUR-GLLSP:



Conclusions

- ▶ Efficient method to compute exact LTS estimates for a coverage range.
- ▶ Allows
 - ▶ to assess the quality (i.e. degree of contamination) of the data;
 - ▶ to identify outliers;
 - ▶ to choose the “best” (e.g. the most efficient) estimate.
- ▶ Algorithm can be applied to other, structured linear models: black box.
- ▶ R package.