# Robust Cellwise Regularized Sparse Regression

Peng Su[1], Garth Tarr[1], Samuel Muller[*,1,2] and Suojin Wang[3]

[1]University of Sydney, [2]Macquarie University, [3]Texas A&M University
[*]*samuel.muller@mq.edu.au*

Biometrics 2023

# Take home messages

## Motivation

Cellwise outliers are a reality and call for robust methods

## Method

Cellwise regularized Lasso with **`regcell`** (available on Github)

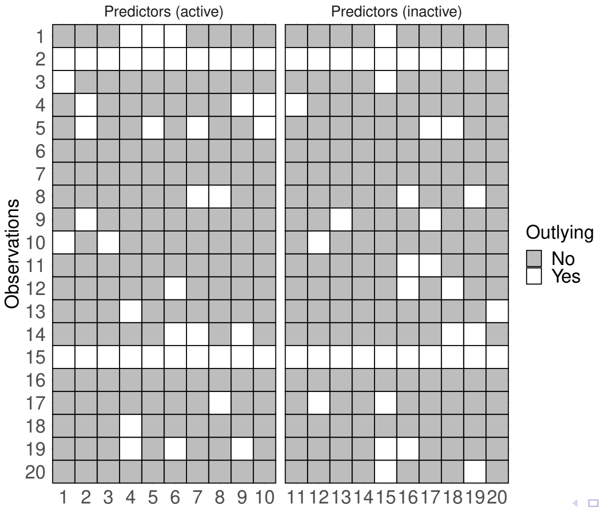
- Simultaneously identify outliers, select and estimate parameters through regularization

## Results

- Region of good selection and prediction performance
- But, not always

# Outline

# Cellwise outliers in context of variable selection



# Robust statistics for modern inference problems

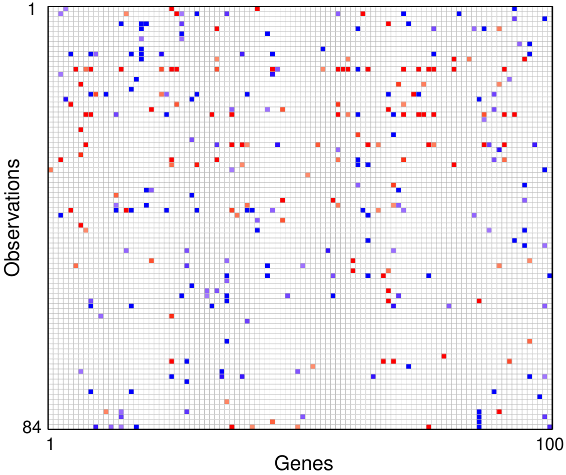
- Context of talk:
  - Cellwise outliers in the design matrix
  - Outliers in the response
  - Linear regression framework for now
  - Balancing competing elements when both $p$ and $p/n$ is large
  - Focus on selection and prediction

- Comments:
  - In the future, using resampling for additional inference considerations
  - **But**, resampling in context of cellwise outliers needs careful thought

# Motivation: Model hip T-score with few genes

- Bone mineral density data from Reppe et al (2010; Bone)
- Raw data:
  - 54,675 gene expression measurements of 84 Norwegian women
  - Outcome of interest is the total hip T-score
- Cleaned data:
  - Screen $p = 100$ genes that have the largest robust correlation with hip T-score
  - Screened variables exhibit contamination rate of 3.6% with probe (column) 236831 having highest contamination of 9.5% and observation (row) 13 of 22%

# Outlier cell map for 100 screened variables



# Growing cellwise outlier detection literature

- DDC: Detecting Deviation Cells (Rousseeuw and Bossch, 2018)
  Robustly predict $\hat{x}_{ij}$ from remaining variables, compare with $x_{ij}$
- Cellflager (Raymaekers and Rousseeuw, 2019)

$$\text{argmin}_{\boldsymbol{\Delta}_i} (\boldsymbol{x}_i - \boldsymbol{\Delta}_i - \boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{\Delta}_i - \boldsymbol{\mu}) + \lambda|\boldsymbol{\Delta}_i|$$

- Cellwise M-estimator (Debruyne et al., 2019)
  Detect rowwise outliers, then detect which cells contribute most
- **Read**: Challenges of cellwise outliers (Raymaekers and Rousseeuw, 2023; arXiv)

# Lasso type regularization: not robust

- Lasso regularization uses $L_1$ loss

$$\underset{\beta}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\beta\|_2^2 + \lambda|\beta|_1$$

- More general regularized objective loss

$$\underset{\beta}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\beta\|_2^2 + P_\lambda(|\beta|)$$

- **But**: Sometimes Lasso does well, even in presence of outliers

# Cellwise regularization: adjusting the $\boldsymbol{X}$

- Chen et al. (2013; ICML) suggested but did not further pursue adjusting $\boldsymbol{X}$

$$\operatorname*{argmin}_{\beta,\boldsymbol{\Delta}} \frac{1}{2}\|\boldsymbol{y} - (\boldsymbol{X} - \boldsymbol{\Delta})\beta\|_2^2 + \eta|\boldsymbol{\Delta}|_1$$

- Solution is non-convex and non-tractable because of the bi-linear term $\boldsymbol{\Delta}\beta$
- Targets dealing with cellwise x outliers (but may also adjust some y outliers)

# Cellwise regularization can be equivalent to Winsorization

- Modify the deviation of the design matrix

$$\underset{\boldsymbol{\Delta}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{\Delta}\|_F^2 + \eta|\boldsymbol{\Delta}|_1$$

- Solved by

$$\hat{\Delta}_{ij} = \begin{cases} \operatorname{sign}(x_{ij})(|x_{ij}| - \eta), & \text{if } |x_{ij}| > \eta \\ 0, & \text{if } |x_{ij}| \leq \eta \end{cases}$$

- This is equivalent to Winsorization
- How to combine minimising regression loss and Winsorization?

# Residual moderated Winsorization adjusts $\boldsymbol{X}$ more subtly

- Towards our solution: modify residual and deviation loss

$$\underset{\boldsymbol{\beta},\boldsymbol{\Delta}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{y} - (\boldsymbol{X} - \boldsymbol{\Delta})\boldsymbol{\beta}\|_2^2 + \underbrace{\frac{1}{2}\|\boldsymbol{X} - \boldsymbol{\Delta}\|_F^2 + \eta|\boldsymbol{\Delta}|_1}_{\text{winsorize elements in } \boldsymbol{X}}$$

- That is, minimise objective loss by shrinking only a few cells
- Cellwise outliers: expect in addition to a large ‘cell deviation’ a large residual

# Cellwise regularization: better allowing for y outliers

- Add $\zeta$ term to accommodate for y outliers

$$\underset{\beta,\boldsymbol{\Delta},\zeta}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{y}-(\boldsymbol{X}-\boldsymbol{\Delta})\beta-\zeta\|_2^2+\frac{1}{2}\|\boldsymbol{X}-\boldsymbol{\Delta}\|_F^2+\eta|\boldsymbol{\Delta}|_1+\theta|\zeta|_1$$

- Add $\lambda|\beta|_1$ to select simultaneously active variables

$$\underset{\beta,\boldsymbol{\Delta},\zeta}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{y}-(\boldsymbol{X}-\boldsymbol{\Delta})\beta-\zeta\|_2^2+\frac{1}{2}\|\boldsymbol{X}-\boldsymbol{\Delta}\|_F^2+\lambda|\beta|_1+\eta|\boldsymbol{\Delta}|_1+\theta|\zeta|_1$$

# Github: https://github.com/PengSU517/regcell

README.md

## regcell

- This package provides the functions to compute the CR-Lasso (cellwise regularized Lasso) proposed by Peng Su, Samuel Muller, Garth Tarr and Suojin Wang. The manuscript could be found soon on Arxiv.
- We added a demonstration (demo) in vignettes.
- We also created an online R repository with some example scripts. https://posit.cloud/content/6051440

To get started, you can install the package using:

```
remotes::install_github("PengSU517/regcell", build = FALSE)
```

For macOS users, if there are some problems with `gfortran`, you can try install the GNU Fortran compiler from this page: https://mac.r-project.org/tools/.

If there are still some errors, you could extract functions from `R` and `src` folders.

# Tuning parameters: selecting $\lambda$ using the BIC

- Many criteria and not yet fully optimised for our method
- We explored with AIC and BIC using the Loss

$$L = 2 \cdot \sum_{i=1}^{n} \rho_{\mathrm{H}} \left( \frac{y_i - (\boldsymbol{x}_i^\star - \hat{\boldsymbol{\Delta}}_i^\star)\hat{\beta}^\star}{\hat{\sigma}}; \theta \right)$$

$$= \left\| \frac{\boldsymbol{y} - (\boldsymbol{X}^\star - \hat{\boldsymbol{\Delta}}^\star)\hat{\beta}^\star}{\hat{\sigma}} - \hat{\zeta}^\star \right\|_2^2 + 2\theta|\hat{\zeta}^\star|_1.$$

- Then $\mathrm{AIC} = L + 2k$ and $\mathrm{BIC} = L + \log(n)k$

# Tuning parameters: selecting $\lambda$, $\eta$ and $\theta$

- BIC as the default to tune $\lambda$
- Set $\eta = z_{0.995} = 2.576$, similar as in DDC (Rousseeuw and Bossche, 2018; Technometrics)
- Set a conservative $\theta = 1$, alternatively $\theta = z_{0.995}$ or other plausible values

# Comparing five alternative methods with CR-Lasso

1. Sparse Shooting S-estimator (SSS)
2. Robust Lars (RLars)
3. Adaptive Lasso regularized MM-estimator (MM-Lasso)
4. Sparse least trimmed squares (SLTS)
5. Lasso

# Moderate dimensional setting

- Sample size $n = 200$
- Number of features $p = 50$ including $p_1 = 10$ active
- Response $\boldsymbol{y}$ is generated by choosing
  - $\beta = (\mathbf{1}_{10}^\top, \mathbf{0}_{p-10}^\top)^\top$
  - $\varepsilon_i \sim N(0, 3^2)$
  - $\boldsymbol{x}_i \sim N(\mathbf{0}, \Sigma_{xx})$ *or* $\boldsymbol{x}_i \sim t_4(\mathbf{0}, \Sigma_{xx})$ , with $\Sigma_{ij} = 0.5^{|i-j|}$
- Contamination rate e varies over 0%, 2% and 5%
- Outliers are generated equally from $N(\gamma, 1)$ and $N(-\gamma, 1)$

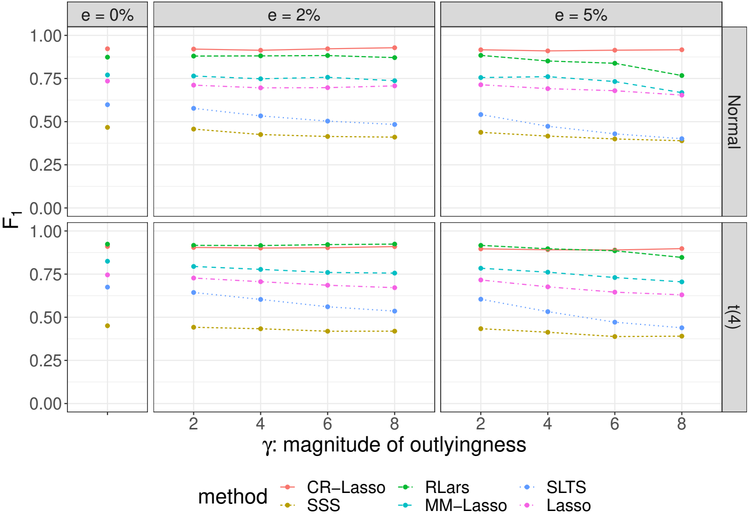
*Additional settings are shown in Su et al (2023; Preprint)*

# Performance metrics

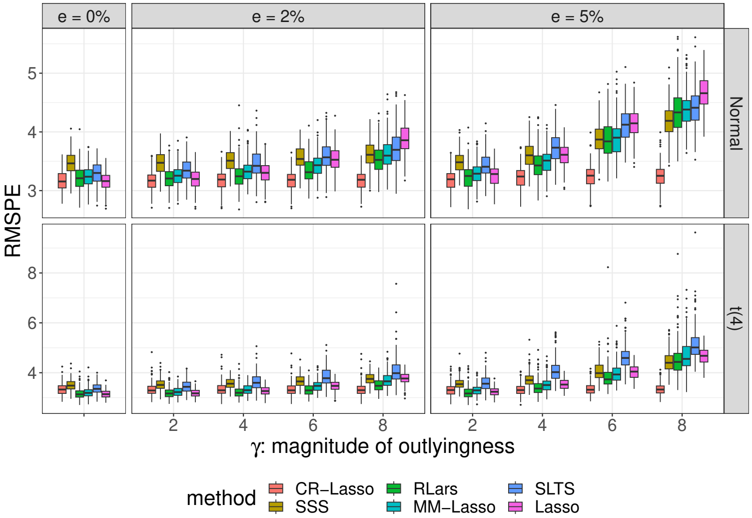
- Root mean squared prediction error (RMSPE)
- Number of true positives (TP)
- Number of false negatives (FN)
- Number of false positives (FP)
- Balancing TP, FN and FP through

$$F_1 = \frac{2\,\mathrm{TP}}{2\,\mathrm{TP} + \mathrm{FN} + \mathrm{FP}}$$

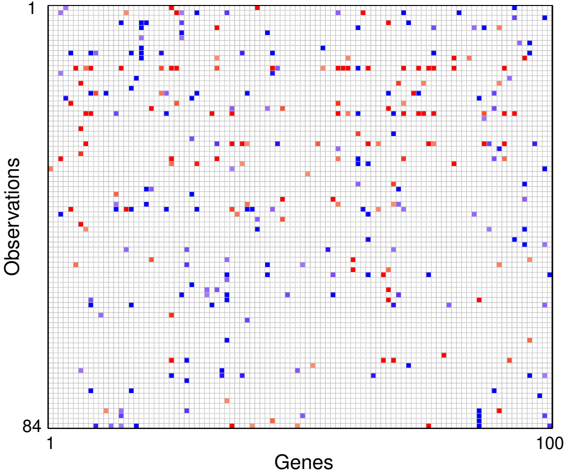
# Prediction results: Selection accuracy



# Prediction results: Mean squared prediction errors



# Recall $\boldsymbol{X}$ in bone mineral density data



# Simulation with this real $\boldsymbol{X}$

**Repeat 200 times:**

1. Obtain a clean (imputed) dataset $\check{\boldsymbol{X}}$ using DDC
2. Randomly pick ten active predictors in each simulation run and for these set $\beta_j \sim U(1, 1.5)$
3. Generate an artificial response $\boldsymbol{y} = \check{\boldsymbol{X}}\beta + \varepsilon$ using screened clean predictors and $\varepsilon \sim N(\mathbf{0}, 0.5^2 \boldsymbol{I})$
4. Train model on 80% observations from the original (contaminated) dataset
5. Validate on the remaining 20% of the imputed (clean) dataset to assess prediction performance

# Results

- **Best performance**
- ***Second best performance***

| | CR-Lasso | SSS | RLars | MM-Lasso | SLTS | Lasso |
|---|---|---|---|---|---|---|
| RMSPE | **1.32** | 2.40 | 1.73 | 1.91 | 2.75 | ***1.64*** |
| TP | ***9.14*** | 6.55 | 7.92 | 7.84 | 6.14 | **9.31** |
| TN | 74.32 | ***75.88*** | **77.05** | 73.49 | 75.75 | 67.83 |
| $F_1$ | **0.55** | 0.45 | ***0.52*** | 0.48 | 0.41 | 0.46 |

## Motivation

Cellwise outliers are a reality and call for robust methods

## Method

Cellwise regularized Lasso with `regcell` (available on Github)

- Simultaneously identify outliers, select and estimate parameters through regularization

## Results

- Region of good selection and prediction performance
- But, not always

## Future work

Heavy tails in predictors; stability selection; robust inference through resampling

# Contact me on samuel.muller@mq.edu.au

**Acknowledgements:**

- Chinese Scholarship Council #201906360181 (Peng Su)
- Australian Research Council Discovery Project #210100521 (Garth Tarr and Samuel Muller)
- Sydney Mathematical Research Institute International Visitor Program (Suojin Wang)

**Reference:**

Su, P., Muller, S., Tarr, G., and Wang, S. (2023).
CR-Lasso: Robust cellwise regularized sparse regression.
*arXiv preprint*, http://arxiv.org/abs/2307.05234.