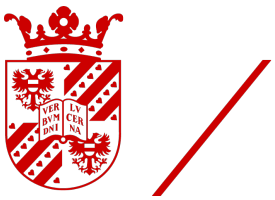


university of groningen

Robust Variable Selection in Sparse Regression: An Analysis of Cellwise and Rowwise Outlier Handling Methods

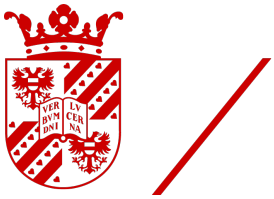
Ferran Llorca Mataix
S4847113

June 16, 2025

Bachelor's Thesis Econometrics and Operations Research

Supervisor: dr. J. Klooster

Second assessor: dr. M.K. Osterhaus



university of groningen

Robust Variable Selection in Sparse Regression: An Analysis of Cellwise and Rowwise Outlier Handling Methods

Ferran Llorca Mataix

Abstract

Cellwise contamination—a modern perspective on outlier modeling—differs from the traditional rowwise approach by allowing individual entries in the data matrix to be independently corrupted. This scenario challenges both classical and modern, recently developed robust estimators. Accordingly, we evaluate proposed methods under each contamination scheme through an extensive Monte Carlo study in a high-dimensional, sparse setting—a context common in fields such as genomics. We focus on ℓ_1-penalized approaches and assess their out-of-sample predictive accuracy and variable-selection perfomance. Our results indicate that DDC preprocessing delivers the best trade-off between prediction error and variable selection, while GR-ALasso excels at identifying the true predictors when contamination is minimal.

1 Introduction

Outliers—data points that deviate from the general pattern—can severely distort statistical analyses and lead to erroneous conclusions. The field of robust statistics (Hampel et al., 1986; Huber and Ronchetti, 2009) addresses this issue by developing methods that remain reliable even in the presence of such anomalies. Since outliers may arise from measurement errors, data entry mistakes, or rare but meaningful events, and because manual data cleaning is often impractical, robust methods aim to mitigate their influence and provide more reliable estimates. This is especially crucial, for instance, in high-stakes fields such as medicine, where misinterpretation of data can directly affect patient care.

Consider Figure 1.1, which illustrates two ways of thinking about outliers. On the left side, the figure shows the case of *rowwise* contamination, where all variables in a given row are assumed to be outliers—for example, an observation originating from a different population. On the right side, the figure shows *cellwise* contamination (Alqallaf et al., 2009), where individual entries, or cells, within the data matrix may be contaminated. This perspective allows for partial contamination within rows, meaning that some elements of a row may still contain valid information—ideally, information that the method used for inference can still recover.

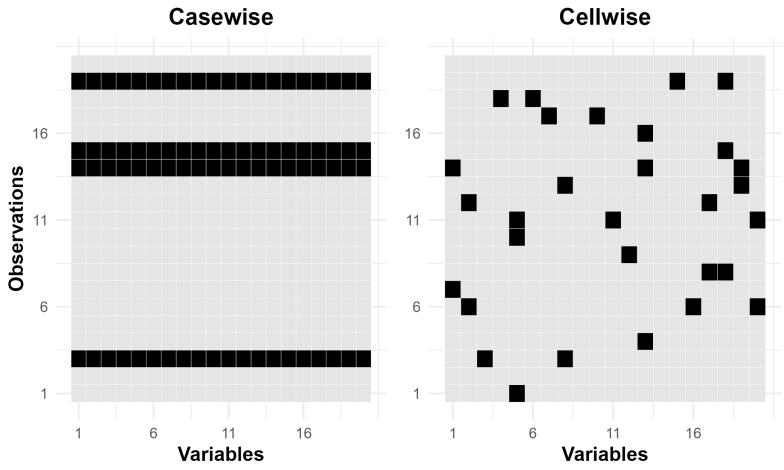


Figure 1.1: Casewise and cellwise contamination examples. Black cells simulate outliers, whereas the rest are considered to be clean entries.

Traditional robust methods (Huber and Ronchetti, 2009; Hampel et al., 1986; Seheult and Green, 1989) were developed assuming contamination affected entire rows; this can limit their effectiveness when contamination occurs at the cell level. In such cases, important signals may be masked or lost if entire observations are downweighted. This issue is particularly critical in modern applications involving high-dimensional data, where the number of observations is already limited. For instance, in biomedical research, a study on triple-negative breast cancer (TNBC) (Segaert et al., 2019) demonstrated how modern preprocessing techniques revealed additional relevant gene expressions that classical robust estimators had overlooked. Similar problems arise in industrial contexts. Pfeiffer and Filzmoser (2023) showed that robust methods enhanced predictive performance and uncovered chemically meaningful markers of engine oil degradation from complex spectral and image data.

Not only is contamination a concern, but the data's high dimensionality also poses a challenge. To mitigate overfitting and improve stability, regularised estimators are

commonly employed, shrinking coefficients to yield more generalisable models. Ridge regression (Hoerl and Kennard, 1970) adds an ℓ_2 penalty to the ordinary least squares objective, alleviating multicollinearity and ensuring a unique solution even when $\mathbf{X}^\top\mathbf{X}$ is singular. While Ridge shrinks coefficients toward zero, it does not perform variable selection. In contrast, the Lasso, introduced by Tibshirani (1996) and further studied in high-dimensional contexts by Bühlmann and van de Geer (2011), uses an ℓ_1 penalty, promoting sparsity by setting some coefficients exactly to zero. This variable-selection feature not only stabilises estimation in high-dimensional settings but also produces simpler, more interpretable models by retaining only the most influential predictors.

This study evaluates the performance of ℓ_1-penalised estimators, focusing on their feature selection capabilities and predictive accuracy in sparse, high-dimensional settings under contamination. We employ Monte Carlo (MC) simulations to generate synthetic datasets and introduce controlled contamination. Both contamination regimes are considered, with the goal of determining whether a single method can be effectively applied regardless of the contamination type and the intended purpose, be it prediction or variable selection. Our results indicate that applying the DDC data-cleaning method (Rousseeuw and Van den Bossche, 2018) followed by the standard Lasso (Tibshirani, 1996) provides a favourable balance between variable selection and predictive accuracy across both contamination types. However, when the contamination rate is low, GR-ALasso (Su et al., 2023) outperforms the other methods in terms of feature selection.

The remainder of the paper is organised as follows. Chapter 2 introduces the contamination models and reviews the theoretical foundations of casewise and cellwise robustness. Chapter 3 formalises the methods compared. Chapter 4 describes the simulation framework, contamination scenarios, and performance metrics. Chapter 5 presents the simulation results. Finally, Chapter 6 provides a conclusion, followed by the discussion and final remarks in Chapter 7.

2 Literature Review

One of the most well-known contamination models was introduced by Huber (1964). It assumes that observations are drawn from a mixture distribution of the form $X \sim (1-\varepsilon)F + \varepsilon H$, where F denotes a well-specified core distribution (typically a normal distribution), H is an unknown contaminating distribution, and $\varepsilon \in [0, 0.5)$ controls the fraction of contamination. The goal is to perform inference on the central distribution F, even when some of the observed data may originate from the contaminating distribution H.

Under this contamination model, Huber (1964) introduced M-estimators as robust alternatives to classical estimators, aiming to reduce the influence of outliers on parameter estimates. In the context of regression, these estimators are obtained by minimising a loss function applied to the residuals, defined as the differences between the observed responses and the predicted values. For example, the Huber loss function behaves quadratically for small residuals, like in ordinary least squares (OLS), but transitions to a linear loss for large residuals, thereby down-weighting the influence of potential outliers in the response variable.

While the rowwise contamination perspective laid the groundwork for much of modern robust statistics, its binary assumption—that entire observations are either completely clean or fully contaminated—can be restrictive in more complex settings. In practice, contamination may affect only some variables within an observation. Leek et al. (2010) demonstrated that high-throughput technologies—capable of processing large volumes of

data, such as genomic measurements—can introduce systematic differences due to external factors such as lab conditions, personnel, or reagent batches. This points to the presence of more intricate and potentially structured contamination patterns. Such phenomena are not limited to biological data: Kaszuba (2014) observed similar effects in financial markets, where extreme values in multivariate stock return data often occur simultaneously across variables, suggesting dependencies among contaminated entries. These findings highlight the complexity of real-world data.

A modern approach to outliers was developed by Alqallaf et al. (2009). A cellwise contamination model is proposed, where a random vector $\mathbf{Y} \in \mathbb{R}^p$ follows a nominal distribution with density given by: $f_{\mathbf{Y}}(\mathbf{y}) = h\left((\mathbf{y}-\boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}_0^{-1}(\mathbf{y}-\boldsymbol{\mu}_0)\right)$, with $\boldsymbol{\mu}_0$ representing the location parameter of interest. Instead of observing $\mathbf{Y}$ directly, we observe a contaminated version $\mathbf{X}$, generated as:

$$\mathbf{X} = (\mathbf{I}-\mathbf{B})\mathbf{Y} + \mathbf{B}\mathbf{Z}, \tag{2.1}$$

where $\mathbf{B} = \text{diag}(B_1, \ldots, B_p)$ is a diagonal matrix of independent Bernoulli random variables with $\mathbb{P}(B_i = 1) = \epsilon_i$. Here, $\mathbf{Z}$ is an arbitrary contamination vector, and $\mathbf{Y}$, $\mathbf{B}$, and $\mathbf{Z}$ are mutually independent. This model allows the contamination structure to vary via the dependence structure of $\{B_i\}$.

Under the cellwise contamination model, consider the *Fully Independent Contamination Model* (FICM), in which each entry indicator B_j for $j = 1, \ldots, p$ is an independent Bernoulli(ϵ) random variable. The probability that a row is completely uncontaminated is $\Pr(\text{clean row}) = (1-\epsilon)^p$, which becomes very small as either the number of predictors or the contamination rate increases. Consequently, traditional robust estimators—built on Huber's assumption that a majority of observations are fully "clean"—lose their reliability in this setting (Rousseeuw and Van den Bossche, 2018; Agostinelli et al., 2015).

Despite their known limitations under cellwise contamination, classical robust estimators remain relevant and are included in this study to evaluate their performance. In particular, we focus on robust estimators extended by an ℓ_1 penalty. For example, Alfons et al. (2013) proposed an estimator combining the least trimmed squares (LTS) (Rousseeuw, 1984) with an ℓ_1 regularisation penalty. This combination enables the LTS estimator to be applied in high-dimensional contexts while simultaneously performing variable selection. Moreover, the resulting estimator is more robust than classical M-estimators due to its higher breakdown point. The breakdown point, first introduced by Hampel (1971), is defined as the smallest fraction of contamination in the data that can cause an estimator to yield arbitrarily poor results. Thus, the sparse LTS (SLTS) estimator can tolerate a larger proportion of contamination before breaking down.

Another classical robust estimator enhanced by the addition of an ℓ_1 penalty is the least absolute deviations (LAD) estimator. The resulting LAD-Lasso, introduced by Wang et al. (2007), employs the least absolute deviation loss function and exhibits improved robustness compared to the standard Lasso when outliers occur in the response variable. However, this estimator still has a relatively low breakdown point because it is not robust to contamination in the design matrix and provides only limited protection against large values in the response.

Cellwise robust estimators are a relatively recent development, with much research focused on addressing outliers at the individual cell level. For example, Su et al. (2023) introduced the GR-ALasso, which first constructs a robust covariance matrix using pairwise estimators and then applies the adaptive Lasso to the adjusted design matrix and response vector. This two-step approach maintains high accuracy even with substantial contamination, provided the outliers are of moderate size. More recently, Su et al. (2024)

proposed the CR-Lasso, which simultaneously minimises a regression loss and a cell deviation penalty to achieve sparse feature selection in the presence of cellwise corruption. Through simulations and real-data experiments, CR-Lasso has demonstrated competitive performance compared to existing methods.

An alternative, more general approach is to apply a preprocessing step that cleans the data before fitting any estimator. One such procedure is the Detecting Deviating Data Cells (DDC) method by Rousseeuw and Van den Bossche (2018). Although DDC is primarily a data-cleaning technique rather than a Lasso variant, it deserves inclusion due to its broad applicability. The method first identifies outliers by examining both individual variables and their pairwise correlations, then replaces flagged values with predictions to produce a cleaned data matrix. After this preprocessing, any estimator can be applied to the cleaned data, reducing the influence of outliers while preserving most of the original information. Moreover, in high-dimensional settings, where the number of predictors is large, DDC can improve prediction performance by exploiting multivariate relationships to more accurately impute contaminated cells.

We therefore review and evaluate existing approaches, ranging from preprocessing techniques like DDC to integrated methods such as GR-ALasso and adaptations of classical robust estimators for high-dimensional data, to assess their strengths and limitations under rowwise contamination and the modern cellwise contamination scenarios.

3 Problem Formulation

3.1 Classical Linear Regression Model

In this study, we employ the classical linear regression (CLR) model. The assumptions required for the validity of the CLR model are detailed in Appendix A. The model is defined for each observation as:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i, \quad i = 1, \ldots, n, \tag{3.1}$$

where $y_i \in \mathbb{R}$ is the response variable for observation i, $x_{ij} \in \mathbb{R}$ for $j = 1, \ldots, p$ are the predictor values, $\beta_0 \in \mathbb{R}$ is the intercept, $\beta_j \in \mathbb{R}$ for $j = 1, \ldots, p$ are the regression coefficients quantifying the influence of each predictor on y_i, and $\varepsilon_i \in \mathbb{R}$ represents the unobservable error term capturing the random noise in the data, which is generated from a normal distribution with mean zero and variance σ^2, i.e., $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. We set $\beta_0 = 0$ for notational simplicity. In matrix notation, the model can be written as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{3.2}$$

where $\mathbf{y} \in \mathbb{R}^n$ is the vector of responses, defined as $\mathbf{y} = (y_1, y_2, \ldots, y_n)^\top$; $\mathbf{X} \in \mathbb{R}^{n \times p}$ is the design matrix of predictors (with the intercept column omitted since β_0 is set to 0), defined as $\mathbf{X} = (\mathbf{x}_1^\top, \mathbf{x}_2^\top, \ldots, \mathbf{x}_n^\top)^\top$; $\boldsymbol{\beta} \in \mathbb{R}^p$ is the vector of regression coefficients, given by $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)^\top$; and $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ is the vector of unobservable random errors, defined as $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)^\top$.

However, consider these two further restrictions. The first being that the number of predictors is much greater than the number of observations, i.e., $p \gg n$. This yields a high-dimensional data case. Furthermore, we decompose $\boldsymbol{\beta}$ into $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^\top, \boldsymbol{\beta}_2^\top)^\top$, where we also define $p = p_1 + p_2$, and write Equation (3.2) in the same way as Yüzbaşı et al. (2020):

$$\mathbf{y} = \mathbf{X}_1\boldsymbol{\beta}_1 + \mathbf{X}_2\boldsymbol{\beta}_2 + \boldsymbol{\varepsilon}. \tag{3.3}$$

Here, $\boldsymbol{\beta}_1 = (\beta_1, \beta_2, \ldots, \beta_{p_1})^\top$ denotes the coefficients of the relevant predictors we aim to estimate, while $\boldsymbol{\beta}_2 = (\beta_{p_1+1}, \beta_{p_1+2}, \ldots, \beta_{p_1+p_2})^\top$ corresponds to covariates that do not contribute to the response but add noise. All other terms are as previously defined. This sparse setting motivates the use of an ℓ_1 penalty, which enables variable selection and hopefully accurately recovers the vector of true coefficients $\boldsymbol{\beta}_1$. However, both cellwise and rowwise contamination are likely to impair performance, so additional measures are needed to ensure estimator robustness.

3.2 Estimators

3.2.1 Lasso

The first paper to introduce the *least absolute shrinkage and selection operator* (Lasso) was Tibshirani (1996). It argued that Lasso's ability to perform variable selection while adding bias, resulting in lower variance, made it an attractive estimator compared to ordinary least squares. It also compared favorably to Ridge regression, another regularisation method introduced by Hoerl and Kennard (1970). Unlike ridge regression, which can only shrink coefficients towards zero, Lasso can set coefficients at exactly zero, effectively selecting only the variables that truly influence the response, while still maintaining the bias-variance tradeoff. Formally, Lasso is defined as follows:

$$\hat{\boldsymbol{\beta}}^{\text{lasso}} = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \left\{ \|\boldsymbol{y} - \boldsymbol{X}\,\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 \right\}, \tag{3.4}$$

where $\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})$ is the ordinary least squares (OLS) objective function, now augmented with the ℓ_1 penalty introduced by Lasso, defined as $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} |\beta_j|$, where $\lambda \geq 0$ is a tuning parameter that is typically selected using methods like cross-validation. This involves splitting the data into training and testing sets, fitting the model with different values of λ, and choosing the one that results in the lowest prediction error on the test set. By iterating over many values of λ, the model balances complexity and accuracy to achieve the best performance.

The regularisation parameter λ in the Lasso determines the strength of shrinkage applied to the coefficients: larger values of λ drive more coefficients exactly to zero, while smaller values yield correspondingly milder shrinkage. This ability to enforce sparsity makes the Lasso especially useful in high-dimensional settings, where interpretability and model simplicity are crucial and ordinary least squares fails due to singular design matrices. Nonetheless, the Lasso can also be advantageous in low-dimensional contexts when variable selection is desired.

Zhao and Yu (2006) showed that the *Irrepresentable Condition* (included in Appendix A) is an almost necessary and sufficient condition for the Lasso to achieve model selection consistency. This condition, which depends on the structure of the design matrix, requires that the irrelevant variables not be too strongly correlated with the relevant ones. In practical terms, it sets a theoretical limit on Lasso's ability to recover the true support of the parameter vector, even in large samples.

Moreover, Wainwright (2009) complements this by showing that, under a Gaussian design, there exists an approximate sample-size threshold: $n \approx 2\log(p-s) + s$, above which the Lasso recovers the correct support with high probability, and below which it typically fails. Here, s is the number of active (nonzero) predictors. This means that the sample size needs to be sufficiently large to ensure accurate variable selection

Various aspects have been extensively examined in the literature, notably the oracle property of the Lasso (van de Geer and Bühlmann, 2009). Broadly speaking, the oracle

property means that the estimator performs as if the true underlying model were known in advance, consistently identifying the correct set of non-zero coefficients (model selection consistency) and estimating them with asymptotic efficiency. This behaviour depends on conditions such as the restricted eigenvalue condition (Bickel et al., 2009).

Unfortunately, the Lasso is highly sensitive to outliers. As shown by Alfons et al. (2013), its breakdown point is only $\frac{1}{n}$, meaning that a single contaminated observation can severely bias the estimates. This fragility arises from the squared-error loss, where an extreme response value y_i can dominate the fit. Moreover, cellwise outliers can distort the correlation structure of the design matrix, potentially violating the Irrepresentable Condition. As a result, the Lasso may fail to recover the true support and may even select predictors that contribute only noise.

3.2.2 LAD - Lasso

An improvement in the robustness properties of Lasso was introduced by the LAD-Lasso estimator, proposed by Wang et al. (2007). To address the issue of outliers in the response variable, they combined two ideas: the Least Absolute Deviations (LAD) loss function and the ℓ_1 penalty. This led to the following formulation:

$$\hat{\boldsymbol{\beta}}^{\text{LAD-Lasso}} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \left\{ \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_1 + \lambda \|\boldsymbol{\beta}\|_1 \right\}. \tag{3.5}$$

Here, the key difference with the standard Lasso lies in the loss function: $\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_1 = \sum_{i=1}^{n} \left| y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta} \right|$ which measures the sum of absolute residuals instead of their squared values.

This change reduces the influence of large residuals, as they are not exaggerated by squaring. As a result, the estimator is less sensitive to outliers in the response variable and demonstrates improved robustness compared to Lasso. However, its effectiveness is limited to scenarios where contamination occurs primarily in the responses, in line with the rowwise contamination setting.

The LAD-Lasso is also shown by Wang et al. (2007) to satisfy the oracle property. This means it can correctly identify the set of non-zero (active) coefficients by setting irrelevant ones to zero. Additionally, because it minimises the sum of absolute deviations, the LAD-Lasso is robust to heavy-tailed error distributions. While the regularisation parameter can be selected using common techniques (e.g., cross-validation), the authors, aiming to establish theoretical properties, chose the BIC criterion for model selection. However, Alfons et al. (2013) demonstrated that the LAD-Lasso inherits a breakdown point of only $\frac{1}{n}$, identical to that of the standard Lasso.

3.2.3 Sparse Least Trimmed Squares (SLTS)

Alfons et al. (2013) propose the Sparse Least Trimmed Squares (SLTS) estimator, which combines the robustness of Least Trimmed Squares (LTS) Rousseeuw (1984) with the sparsity-inducing ℓ_1-penalty of the Lasso. Formally, it minimizes the sum of the h smallest squared residuals plus the regularization penalty on β:

$$\hat{\beta}_{\text{SLTS}} = \arg\min_{\beta} \sum_{i=1}^{h} r^2_{(i)}(\beta) + h\lambda \sum_{j=1}^{p} |\beta_j|, \tag{3.6}$$

where $r^2_{(1)}(\beta) \leq \cdots \leq r^2_{(n)}(\beta)$ are the ordered squared residuals, and $h \leq n$ determines the level of trimming. Note that when $h = n$, the SLTS estimator reduces to the standard

Lasso estimator of Tibshirani (1996), as no trimming is applied.

The finite-sample breakdown point of the SLTS estimator is given by $\varepsilon^* = \frac{n-h+1}{n}$, representing the proportion of contamination the estimator can tolerate before becoming unreliable. A smaller value of h increases robustness but may reduce model accuracy. To balance robustness and model fit, Alfons et al. (2013) recommend setting $h = \alpha n$ with $\alpha = 0.75$, which yields a breakdown point of approximately 25%. While in theory the breakdown point can exceed 50% for smaller h, the goal of robust statistics remains to fit the majority of the data.

3.2.4 GR-ALasso

The Gaussian Rank Adaptive Lasso estimator (GR-ALasso), introduced by Su et al. (2023), aims to be robust against cellwise contamination by combining the adaptive ℓ_1 penalty of Zou (2006) with the Gaussian-rank correlation estimator of Boudt et al. (2012).

The approach begins by using the Gaussian Rank (GR) correlation to estimate a positive semi-definite correlation matrix. From this correlation matrix, a robust covariance matrix is constructed by incorporating robust scale estimates for each variable. This covariance matrix is then used to reformulate the regression loss function. Unlike the classical OLS loss, which minimises the sum of squared residuals, this loss is expressed in terms of the robust covariance components, enabling more reliable coefficient estimation.

To achieve sparse variable selection, an ℓ_1-penalty with adaptive weights is incorporated into the loss function, yielding the GR-ALasso estimator. Due to the length and complexity of the full derivation, we refer the reader to Appendix A for details. For completeness, the final form of the estimator is presented here:

$$\hat{\boldsymbol{\beta}}^{\text{GR-ALasso}} = \arg\min_{\mathbf{b}} \left\{ \|\mathbf{v} - \mathbf{W}\mathbf{b}\|_2^2 + \lambda \sum_{j=1}^{p} \omega_j |b_j| \right\}, \tag{3.7}$$

where $\omega_j = \frac{1}{|\tilde{\beta}_j|}$ are the adaptive weights based on initial estimates $\tilde{\boldsymbol{\beta}}$. In high-dimensional settings such as ours, ridge regression (Hoerl and Kennard, 1970) is typically used to obtain these initial estimates. Moreover, the authors demonstrate that GR-ALasso satisfies the oracle property, as defined for the adaptive lasso by Zou (2006).

3.2.5 DetectDeviatingCells (DDC)

Rousseeuw and Van den Bossche (2018) introduced the *DetectDeviatingCells* method to handle contaminated data, offering advantages such as applicability to datasets with more than 50% contaminated rows and usability beyond high-dimensional data. Unlike the other estimators discussed in this study, this method is a preprocessing technique rather than a direct estimator based on linear regression with an ℓ_1 penalty. Therefore, we then fit a Lasso on the cleaned data. A brief outline of the method is given.

The method assumes that the rows of the data matrix are generated from an unknown p-variate mean vector $\boldsymbol{\mu}$ and a positive semidefinite covariance matrix $\boldsymbol{\Sigma}$. After the data is generated, some individual cells may become contaminated. The original paper notes that the algorithm still performs well even if the data are not generated from a multivariate Gaussian distribution. Here, we focus mainly on the intuition behind the algorithm rather than its exact theoretical setup.

Using robust estimators for location, scale, correlation, and linear regression, the

method aims to transform a contaminated data matrix $\boldsymbol{X}_{\text{cont}}$ into a "clean" version $\boldsymbol{X}_{\text{clean}}$, where outliers and missing values are replaced by predicted values. It begins by examining the data matrix $\boldsymbol{X}_{\text{cont}}$ one column at a time, flagging cells that appear anomalous. Each flagged cell is then predicted based on the non-flagged cells in the same row, focusing on those in columns that are correlated with the column under consideration. If the observed value of a cell deviates significantly from its predicted value, it is classified as an outlier. The algorithm then produces the cleaned data matrix $\boldsymbol{X}_{\text{clean}}$ by replacing these outliers and any missing values with their corresponding predictions.

For our study, we fit a Lasso estimator using the cleaned data matrix produced by this algorithm. Formally, this can be expressed as

$$\hat{\boldsymbol{\beta}}^{\text{DDC}} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \left\{ \left\| \boldsymbol{y} - \boldsymbol{X}_{\text{clean}}\,\boldsymbol{\beta} \right\|_2^2 + \lambda \left\| \boldsymbol{\beta} \right\|_1 \right\}, \tag{3.8}$$

where the only difference with Equation (3.4) is the data matrix used. It is worth noting that, given DDC's strategy of predicting flagged cells based on other cells in the same row, having more predictors provides more information for these predictions. Therefore, in high-dimensional settings, the method may perform even better, as observed by Rousseeuw and Van den Bossche (2018). Furthermore, although DDC was designed for cellwise outlier detection, it can still flag numerous contaminated cells within a single observation, providing measures against casewise contamination.

4 Methodology

4.1 Simulation Parameters and Data Generating Process

To compare the estimators and methods of outlier detection introduced in Section 3, we perform a Monte Carlo simulation study in a high-dimensional data setting. Of interest are both predictive performance on clean data and variable selection capabilities in a sparse setting, where only a subset of predictors is active.

In each Monte Carlo replication (200 replications per contamination level ε), we generate a high-dimensional dataset with $n = 100$ observations and $p = 300$ predictors, of which exactly $p_r = 10$ are truly active ($\beta_j = 1$ for $j = 1, \ldots, 10$ and $\beta_k = 0$ for $k = 11, \ldots, 300$). We set the outlier magnitude $\gamma = 6$ and consider two contamination schemes—rowwise and cellwise—at levels $\varepsilon \in \{0, 0.05, 0.10, 0.15, 0.20\}$. First, we draw a "clean" design matrix from the multivariate normal distribution

$$X_{\text{clean}} \sim N\big(0, \Sigma\big), \tag{4.1}$$

where Σ has an AR(1) correlation structure with $\Sigma_{ij} = r^{|i-j|}$ and $r = 0.5$, so that the correlation between any two predictors is high if close together but decays when the distance between the predictors increases. Independently we draw the clean error vector $\varepsilon_{\text{clean}} \sim N(0, 1)$, and form the baseline response

$$y_{\text{baseline}} \;=\; X_{\text{clean}}\,\beta \;+\; \varepsilon_{\text{clean}}. \tag{4.2}$$

After forming the clean baseline response, we introduce outliers in one of two ways. Under *rowwise* contamination, we randomly select εn entire observations and, for each selected row i, shift every predictor by $\zeta_{X,ij} \sim N(\gamma, 1) \times \{\pm 1\}$ and the corresponding error by $\zeta_{\varepsilon,i} \sim N(\gamma, 1) \times \{\pm 1\}$, so that $X_{ij} \leftarrow X_{\text{clean, ij}} + \zeta_{X,ij}$ for all j and $\varepsilon_i \leftarrow \varepsilon_{\text{clean, i}} + \zeta_{\varepsilon,i}$.

Under *cellwise* contamination, we independently select εn rows in each column j and add $\zeta_{X,ij} \sim N(\gamma, 1) \times \{\pm 1\}$ to each marked cell (i, j), while separately choosing εn rows at random to contaminate the error via $\zeta_{\varepsilon,i} \sim N(\gamma, 1) \times \{\pm 1\}$. In both schemes, letting X and ε denote the contaminated design and error, the observed response is then

$$y \;=\; X\,\beta \;+\; \varepsilon. \tag{4.3}$$

Intuitively, under rowwise contamination, both the design matrix X and the error term ε are affected for the same set of rows, meaning the response y is contaminated through both its predictors and its noise component. Under cellwise contamination, individual cells in the design matrix are corrupted independently across columns, and the error term is contaminated separately in a randomly selected subset of rows. As a result, in the cellwise case, the design matrix is always partially contaminated, while the response y is only contaminated if its associated error term is affected.

4.2 Practical Estimator Fitting and Performance Metrics

The estimators are fitted following the recommendations found in the relevant literature. For the Lasso, we adopt the procedure suggested by Tibshirani (1996) but later also supported by Bühlmann and van de Geer (2011) for practical uses, cross-validation. Therefore, we make use of the `glmnet` package (Friedman et al., 2010) with 10-fold cross-validation to select the regularisation parameter λ that minimises the mean cross-validated error. For the LAD–Lasso, Wang et al. (2007) use the Bayesian Information Criterion (BIC) (Schwarz, 1978) rather than cross-validation or the Akaike Information Criterion (AIC) (Akaike, 1974) for tuning, consistent with their theoretical objectives. However, we instead employ 10-fold cross-validation using the `quantreg` package Koenker (2023), aligning with our goal of evaluating both predictive performance and variable selection practically.

To fit the SLTS estimator, we follow the procedure outlined by Alfons et al. (2013). While they recommend trimming approximately 25% of the data, we opt for a 10% trimming procedure due to our relatively small sample size of 100 observations, retaining the remaining 90% for fitting the estimator. Unlike the other estimators in our study, the regularisation parameter λ is selected using the Bayesian Information Criterion (BIC). The estimator is then fitted using the `sparseLTS` function from the `robustHD` package (Alfons, 2021), with reweighting enabled to improve statistical efficiency and robustness.

For the DDC with then using the Lasso procedure, we first apply the DDC algorithm from the `cellWise` package (Hubert, 2023) to detect and adjust outliers in the design matrix, and then fit a Lasso model using the same cross-validation strategy as for the standard Lasso.

The GR-ALasso is fitted by first estimating a robust covariance matrix using the Gaussian rank correlation, followed by a ridge-regularised initial estimator to compute adaptive weights. The final estimated coefficients are obtained by solving a weighted Lasso problem using the `glmnet` package once again, where the regularisation parameter λ is selected via 5-fold cross-validation on a pseudo-dataset derived from rank-transformed variables. The code for the estimator is available in the author's GitHub repository as the `robcovsel` package [1].

The data used in the simulations is generated using the `genevar` function from the `regcell` package (Su et al., 2024), which is available from the main author's GitHub repository[2]. This data-generating function (and package containing other methods) allows for

[1] `https://github.com/PengSU517/robcovsel`
[2] `https://github.com/PengSU517/regcell`

flexible specification of contamination type, outlier magnitude, and covariance structure.

To evaluate the performance of the estimators, we record metrics that capture both predictive accuracy and variable-selection performance. The latter is particularly important given the sparsity-inducing nature of the ℓ_1 penalty employed by Lasso-type methods.

Following Yu et al. (2014), who studied the performance of robust M-estimators, we track the following metrics: the root mean squared prediction error (RMSPE), the robust bias (RB), and the median absolute deviation (MAD). The RMSPE is a standard measure of predictive accuracy and assesses how well an estimator generalises to new data. It is computed by fitting the model on contaminated training data and evaluating predictions on clean test data:

$$\text{RMSPE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\big(y_i - \hat{y}_i\big)^2}, \tag{4.4}$$

where y_i is the true response and $\hat{y}_i$ is the predicted value. The RMSPE quantifies the average squared deviation between predictions and actual outcomes, providing a summary of the model's out-of-sample predictive performance.

The robust bias and median absolute deviation for each coefficient β_i are defined as:

$$RB_i = \text{median}\big(\hat{\beta}_i - \beta_i\big), \tag{4.5}$$

$$MAD_i = \text{median}\big(|\hat{\beta}_i - \beta_i|\big). \tag{4.6}$$

Here, $\hat{\beta}_i$ represents the estimate of the i-th coefficient obtained from repeated Monte Carlo simulations. The definition of robust bias used in this study differs from that in Yu et al. (2014). Instead of computing the robust bias (RB) separately for each coefficient as they did, we calculate a single summary measure: the median bias across all active coefficients. Specifically, RB_i reflects the median deviation of the estimator from the true value for the i-th coefficient, while MAD_i (Median Absolute Deviation) captures the typical magnitude of absolute error around the true coefficient. Both metrics are computed only for the active (non-zero) coefficients. Together, RB and MAD provide insight into the accuracy and robustness of the estimates under different contamination scenarios.

Finally, to evaluate variable-selection performance, we employ the F1 score. Originally proposed by Van Rijsbergen (1979) and subsequently used in studies like Bleich et al. (2014) and Su et al. (2024), the F1 score is well suited for assessing sparsity-inducing estimators because it balances precision and recall.

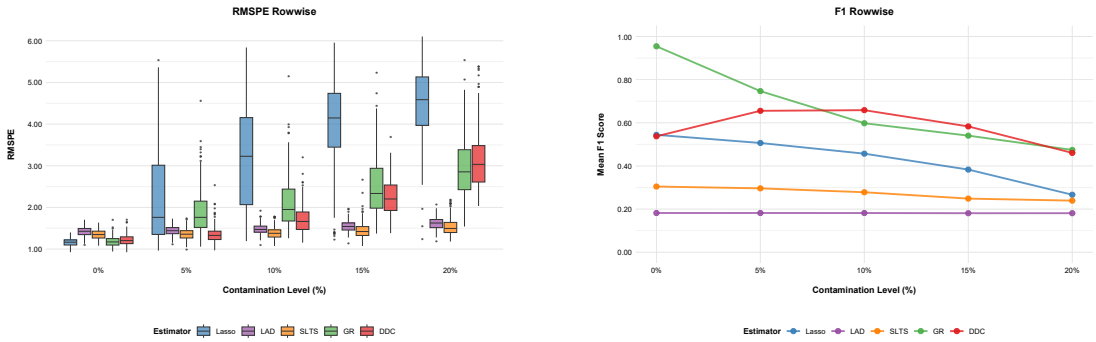
Precision (P) is defined as the proportion of selected variables that are truly relevant, and recall (R) is the proportion of relevant variables that are correctly selected. The F1 score is their harmonic mean:

$$F_1 = \frac{2PR}{P+R} = \frac{2TP}{2TP + FP + FN}, \tag{4.7}$$

where TP, FP, and FN denote true positives, false positives, and false negatives, respectively. By combining these components, the F1 score provides a single, balanced measure of variable-selection effectiveness, reflecting both the accuracy (precision) and completeness (recall) of the selected model.

5 Results

5.1 Casewise Contamination



(a) RMSPE distribution across contamination.

(b) Mean F_1 score across contamination.

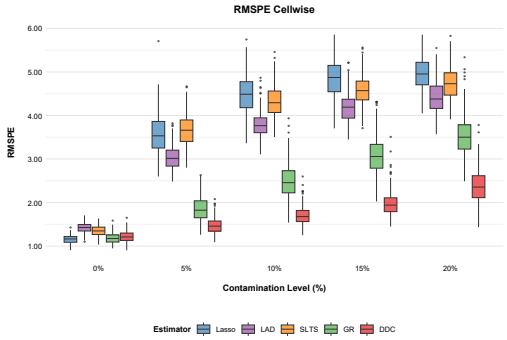
Figure 5.1: Performance comparison of methods under casewise contamination. Subfigure (a) shows the distribution of RMSPE values, and subfigure (b) shows the mean F_1 score, both computed over the MC iterations and plotted against the contamination level. Colors indicate the method: blue = Lasso, purple = LAD-Lasso, orange = Sparse LTS, green = GR-ALasso, and red = DDC.

Figure 5.1a reports the distribution of RMSPE values across the MC simulations. At 0 % contamination, Lasso outperforms all of the robust methods. However, even a slight amount of contamination causes its error to explode, producing unreliable estimates. In contrast, LAD-Lasso and SLTS both maintain a low mean RMSPE and a tight spread, even under severe contamination. Moreover, the DDC procedure tolerates mild contamination (around 5 %) reasonably well, but its performance worsens as the outlier proportion grows. Finally, GR-ALasso performs well on clean data, yet at just 5 % contamination it incurs the highest median RMSPE among the robust methods.

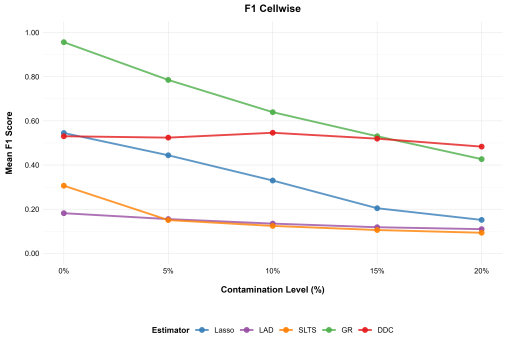
Figure 5.1b, which plots the F_1 score at each contamination level, provides insights into variable-selection performance. Unlike the RMSPE results, GR-ALasso and DDC achieve the highest F_1 scores overall. In particular, DDC's F_1 score initially improves slightly before following the same downward trend as the other estimators at higher contamination levels. Both SLTS and LAD-Lasso exhibit consistently low F_1 scores, with SLTS performing marginally better than LAD-Lasso. Lasso also shows a decline in F_1 value as contamination increases, though it remains above the row-wise robust methods at moderate contamination levels.

5.2 Cellwise Contamination

For the cell-wise contamination scenario, Figure 5.2a shows the RMSPE distributions. Even at just 5 % contamination, Lasso, LAD-Lasso, and SLTS exhibit a sharp jump in RMSPE, indicating that their estimates are severely affected by cell-wise outliers. Consequently, these three methods have higher median RMSPE values overall and perform worse under cell-wise than under row-wise contamination. Although DDC and GR-Alasso are also affected, their error increases are much smaller. Overall, DDC achieves the lowest median RMSPE under non-zero values of contamination, producing the most reliable estimates.



(a) RMSPE distribution across contamination.



(b) Mean F_1 score across contamination.

Figure 5.2: Performance comparison of methods under cellwise contamination. Subfigure (a) shows the distribution of RMSPE values, and subfigure (b) shows the mean F_1 score, both computed over the MC iterations and plotted against the contamination level. Colors indicate the method: blue = Lasso, purple = LAD-Lasso, orange = Sparse LTS, green = GR-ALasso, and red = DDC.

Figure 5.2b plots the mean F_1 scores under cell-wise contamination. Consistent with the good RMSPE results, DDC maintains an almost constant F_1 score even at high contamination levels, demonstrating robustness in variable selection. GR-Alasso achieves the highest F_1 score at most contamination levels, but its performance declines as contamination increases. Lasso, LAD-Lasso, and SLTS continue to exhibit lower F_1 scores overall; once any contamination is introduced, SLTS and LAD-Lasso perform almost identically, with Lasso outperforming both.

5.3 Direct Comparison: Rowwise vs. Cellwise Contamination

	RMSPE			F_1			TPR			FPR			RB			MAD		
	0%	R10	C10	0%	R10	C10	0%	R10	C10	0%	R10	C10	0%	R10	C10	0%	R10	C10
Lasso	**1.16**	3.16	4.50	0.54	0.46	0.33	1.00	0.67	0.49	0.07	0.05	0.05	-0.09	-0.65	-0.94	**0.13**	0.66	0.94
LAD	1.42	1.47	3.80	0.18	0.18	0.13	1.00	**1.00**	0.74	0.31	0.31	0.32	**-0.05**	**-0.11**	-0.81	0.14	0.20	0.81
SLTS	1.35	**1.38**	4.32	0.30	0.28	0.12	1.00	0.99	0.60	0.18	0.19	0.28	-0.13	-0.12	-0.92	0.17	**0.18**	0.92
GR	1.19	2.13	2.49	**0.95**	0.60	**0.64**	1.00	0.84	0.86	**0.00**	**0.04**	**0.04**	-0.09	-0.41	-0.49	0.16	0.47	0.49
DDC	1.22	1.71	**1.71**	0.54	**0.66**	0.55	1.00	0.96	**0.98**	0.07	**0.04**	0.07	-0.09	-0.24	**-0.20**	0.17	0.29	**0.32**

Table 5.1: Summary of global performance metrics for each method under 0% and 10% contamination levels. R10 and C10 denote 10% rowwise and cellwise contamination, respectively. The six metrics reported—RMSPE, F_1 score, TPR, FPR, Robust Bias, and MAD—are averaged over the Monte Carlo simulations. For each metric, the best-performing value (per column) is highlighted in bold.

By observing the values in Table 5.1, it becomes clear that cellwise contamination poses at least as great a threat, if not a greater one, than casewise contamination. This is evident when comparing the metric values against the uncontaminated base case and observing the extent to which they change under the two contamination models. The worse performance of the methods can be seen by the higher RMSPE values and generally lower F1 scores, except for GR-ALasso, which achieves a higher F1 score under cellwise contamination.

The True Positive Rate (TPR) suffers noticeably under cellwise contamination for Lasso, LAD-Lasso, and SLTS, while GR-Alasso and DDC maintain or even improve their

TPR. False Positive Rate (FPR) values are the same or greater when considering cellwise contamination, indicating worse performance. Robust bias is generally worse with cellwise contamination, with the notable exception of DDC, where it improves. Additionally, the median absolute deviation (MAD) increases more sharply under cellwise contamination, indicating greater variability in coefficient estimates.

6 Conclusion

We evaluated robust, ℓ_1-penalised methods in a high-dimensional data setting with synthetic casewise and cellwise contamination. Not only did the study compare predictive accuracy, but, crucially, each estimator's ability to select the correct coefficients by leveraging the automatic feature-selection mechanism to recover the sparse underlying signal. As expected, the different contamination structures posed a challenge for the estimators.

Under row-wise contamination, SLTS minimised prediction error while GR-ALasso maximised the F_1 score, with DDC having a similar performance. Under cell-wise contamination, DDC dominated, achieving the lowest prediction error, the highest true-positive rate, and minimal false positives. GR-ALasso retained an F_1 advantage only when contamination remained low.

Overall, no single estimator performed best across both contamination types. DDC provided the most consistent trade-off between prediction accuracy and variable-selection stability, making it a strong general-purpose choice. However, if low contamination ($<$ 10%) is expected and variable selection is the primary objective, GR-ALasso is preferable for its superior F_1 score.

7 Discussion

We confirm that classical robust methods break down under cellwise contamination—an issue that can and does arise in practice. Newer approaches, such as DDC, handle both row- and cell-level outliers. Yet real-world contamination often follows more complex patterns than these two extremes. The contamination model by Alqallaf et al. (2009) allows for structured outlier patterns that are neither purely rowwise nor purely cellwise. It remains an open question whether existing robust estimators can adapt to these hybrid contamination models.

The strong performance of DDC (Rousseeuw and Van den Bossche, 2018) was not surprising: by leveraging correlations among predictors, it turns high dimensionality into an asset rather than a curse. However, DDC is limited to continuous variables—it cannot process nominal or binary data, which are common in genomics studies (Bühlmann and van de Geer, 2011). In mixed-type datasets, this limitation must be addressed, perhaps by extending DDC or by developing analogous methods for categorical features.

If prediction is our sole concern, Alfons et al. (2013) argued that failing to detect true signals (i.e., a low true positive rate, TPR) is more detrimental than falsely detecting predictors with no effect (i.e., a high false positive rate, FPR). This argument is supported by Table 5.1, where a decrease in TPR is accompanied by a larger RMSPE. It is therefore worth questioning whether maximising TPR—if it comes at the cost of selecting noise—is justified. When predictive accuracy is the primary objective, an estimator need not be overly strict in variable selection: omitting a true coefficient risks losing important information, whereas including a coefficient with no real impact merely adds noise.

This study remains limited by its use of synthetic data. While ideal for controlled comparisons, these datasets may not fully reflect real-world settings, where variables can

experience varying degrees of contamination, coefficients can differ in scale, and correlation structures can be more intricate. Furthermore, we kept the outlier magnitude constant. Future studies could explore varying outlier magnitudes or introduce contamination through more complex synthetic schemes. Another direction worth exploring could be to consider merging data-cleaning and robust penalised-regression approaches. For example, combining the DDC method with GR-ALasso's adaptive weighting might retain the predictive accuracy of DDC followed by Lasso on the cleaned data, while boosting F1 scores for variable selection. Such hybrids could offer the best of both worlds at the possible cost of additional computational time.

Today, data availability is no longer an issue—datasets are vast and growing, so automated preprocessing is essential to ensure data quality before analysis. This is especially critical in fields such as health care, where errors can have fatal consequences. Our study underscores the importance of choosing the right tools for each task: whether that is selecting a method for its prediction accuracy or to uncover the true signals from the data.

ChatGPT (OpenAI, 2025) was employed during the preparation of this study to support writing tasks (including grammar, spelling, punctuation, mathematical notation, and LaTeX), and to assist with programming.

References

Claudio Agostinelli, Andrew Leung, Victor J. Yohai, and Ruben H. Zamar. Robust estimation of location and scatter with cellwise and casewise contamination. *Computational Statistics & Data Analysis*, 83:109–121, 2015. doi: 10.1016/j.csda.2014.07.014.

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

Andreas Alfons. robusthd: An r package for robust regression with high-dimensional data. *Journal of Open Source Software*, 6(67):3786, 2021. doi: 10.21105/joss.03786. URL `https://cran.r-project.org/package=robustHD`.

Andreas Alfons, Christophe Croux, and Sarah Gelper. Sparse least trimmed squares regression for analyzing high-dimensional large data sets. *The Annals of Applied Statistics*, 7(1):226–248, 2013. doi: 10.1214/12-AOAS575.

F. Alqallaf, S. Van Aelst, V.J. Yohai, and R.H. Zamar. Propagation of outliers in multivariate data. *The Annals of Statistics*, 37(1):311–331, 2009.

Peter J. Bickel, Ya'acov Ritov, and Alexandre B. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009. doi: 10.1214/08-AOS620.

J. Bleich, A. Kapelner, E.I. George, and S.T. Jensen. Variable selection for bart: An application to gene regulation. *The Annals of Applied Statistics*, 8(3):1750–1781, September 2014. doi: 10.1214/14-AOAS755.

Kris Boudt, Jonathan Cornelissen, and Christophe Croux. The gaussian rank correlation estimator: robustness properties. *Statistics and Computing*, 22(2):471–483, 2012. ISSN 1573-1375. doi: 10.1007/s11222-011-9237-0. URL `https://doi.org/10.1007/s11222-011-9237-0`.

P. Bühlmann and S. van de Geer. *Statistics for High-Dimensional Data: Method, Theory and Applications*. Springer, 2011. ISBN 978-3-642-20191-2. doi: 10.1007/978-3-642-20192-9.

J.H. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010. doi: 10.18637/jss.v033.i01. URL `https://www.jstatsoft.org/article/view/v033i01`.

F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley & Sons, New York, 1986. ISBN 978-0-471-73577-9. doi: 10.1002/9781118186435.

Frank R. Hampel. A general qualitative definition of robustness. *Annals of Mathematical Statistics*, 42(6):1887–1896, 1971.

A.E. Hoerl and R.W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

P.J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964. doi: 10.1214/aoms/1177703732.

P.J. Huber and E.M. Ronchetti. *Robust Statistics*. John Wiley & Sons, Hoboken, NJ, 2 edition, 2009. ISBN 978-0-470-12990-6.

M. Hubert. *cellWise: Cellwise Outlier Detection and Robust Analysis*, 2023. URL `https://CRAN.R-project.org/package=cellWise`. R package version X.X.

Bartosz Kaszuba. Correlation of outliers in multivariate data. In Myra Spiliopoulou, Lars Schmidt-Thieme, and Ruth Janning, editors, *Data Analysis, Machine Learning and Knowledge Discovery*, Studies in Classification, Data Analysis, and Knowledge Organization. Springer, Cham, 2014. doi: 10.1007/978-3-319-01595-8_29. URL `https://doi.org/10.1007/978-3-319-01595-8_29`.

Roger Koenker. *quantreg: Quantile Regression*, 2023. URL `https://CRAN.R-project.org/package=quantreg`. R package version 5.99.

J.T. Leek, R.B. Scharpf, H. Corrada Bravo, D. Simcha, B. Langmead, W.E. Johnson, D. Geman, K. Baggerly, and R.A. Irizarry. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 11(10):733–739, 2010. doi: 10.1038/nrg2825.

OpenAI. Chatgpt (june 16 version). `https://chat.openai.com`, 2025. Large language model.

P. Pfeiffer and P. Filzmoser. Robust statistical methods for high-dimensional data, with applications in tribology. *Analytica Chimica Acta*, 1279:341762, 2023. doi: 10.1016/j.aca.2023.341762. URL `https://www.sciencedirect.com/science/article/pii/S0003267023009832`.

Peter J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984.

P.J. Rousseeuw and W. Van den Bossche. Detecting deviating data cells. *Technometrics*, 60(2):135–145, 2018. doi: 10.1080/00401706.2017.1405975.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

P. Segaert, M.B. Lopes, S. Casimiro, S. Vinga, and P.J. Rousseeuw. Robust identification of target genes and outliers in triple-negative breast cancer data. *Statistical Methods in Medical Research*, 28(10–11):3042–3056, 2019. doi: 10.1177/0962280218794722. URL `https://doi.org/10.1177/0962280218794722`. Epub 2018 Aug 27.

A. Seheult and P. Green. Review of: Robust regression and outlier detection by p.j. rousseeuw and a.m. leroy. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 152:133–134, 1989. doi: 10.2307/2982847.

P. Su, G. Tarr, and S. Müller. Robust variable selection under cellwise contamination. *Journal of Statistical Computation and Simulation*, 94(6):1371–1387, 2023. doi: 10.1080/00949655.2023.2286316.

P. Su, G. Tarr, S. Müller, and S. Wang. Cr-lasso: Robust cellwise regularized sparse regression, 2024. URL `https://arxiv.org/abs/2307.05234`.

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

Sara van de Geer and Peter Bühlmann. On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics*, 3:1360–1392, October 2009. doi: 10.1214/09-EJS506.

C.J. Van Rijsbergen. *Information Retrieval*. Butterworth, Stoneham, MA, 2 edition, 1979. ISBN 0-408-70929-6.

M. J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (lasso). *IEEE Transactions on Information Theory*, 55(5):2183–2202, 2009. doi: 10.1109/TIT.2009.2016018.

H. Wang, G. Li, and G. Jiang. Robust regression shrinkage and consistent variable selection through the lad-lasso. *Journal of Business & Economic Statistics*, 25(3): 347–355, 2007. doi: 10.1198/073500106000000251. URL `https://doi.org/10.1198/073500106000000251`.

C. Yu, W. Yao, and X. Bai. Robust linear regression: A review and comparison. *Communications in Statistics - Simulation and Computation*, 46(8):4346–4367, 2014. doi: 10.1080/03610918.2016.1202271.

Bahadır Yüzbaşı, S. Ahmed, M. Arashi, and Mina Norouzirad. Lad, lasso and related strategies in regression models. In Fabrizio Ruggeri, Ron S. Kennett, and Silvia Salini, editors, *Statistical Learning and Data Sciences*, pages 429–444. Springer, Cham, 2020. ISBN 978-3-030-21247-6. doi: 10.1007/978-3-030-21248-3_32.

Peng Zhao and Bin Yu. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.

Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006. doi: 10.1198/016214506000000735. URL `https://doi.org/10.1198/016214506000000735`.

A Appendix A

A.1 Assumptions for the Classical Linear Regression Model

Assumption 1 (Linearity)**.** *The relationship between y and $\mathbf{X}$ is linear:*

Assumption 2 (Strict Exogeneity)**.**

$$E[\boldsymbol{\varepsilon} \mid \mathbf{X}] = \mathbf{0},$$

i.e. the (random) errors are uncorrelated with the predictors.

Assumption 3 (No Perfect Multicollinearity)**.** *The predictors are not perfectly correlated, ensuring that $\mathbf{X}^\top \mathbf{X}$ is invertible.*

Assumption 4 (Spherical Error Variance)**.**

$$\mathrm{Var}\big(\boldsymbol{\varepsilon} \mid \mathbf{X}\big) = E\left[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^t \mid \mathbf{X}\right] = \sigma^2 I_n$$

where I_n denotes the identity matrix of order n.
i.e. The errors are homoscedastic and uncorrelated across observations.

A.2 Weak Irrepresentable Condition

Let the design matrix $X \in \mathbb{R}^{n\times p}$ be partitioned as

$$X = \begin{bmatrix} X_{(1)} & X_{(2)} \end{bmatrix},$$

and define the sample covariance (Gram) matrix

$$\Sigma \;=\; \frac{1}{n} X^\top X \;=\; \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix},$$

where Σ_{11} corresponds to the true support and $\Sigma_{21} = \Sigma_{12}^\top$. If $\beta_{(1)}$ are the true nonzero coefficients, then the (weak) Irrepresentable Condition is

$$\left|\Sigma_{21}\, \Sigma_{11}^{-1} \operatorname{sign}\big(\beta_{(1)}\big)\right| \;<\; 1,$$

interpreted element-wise.

A.3 Derivation of the GR-Adaptive Lasso Loss Function

A.3.1 Robust Estimation of the Covariance Matrix

Let $\mathbf{Z} = (\mathbf{y}, \mathbf{X}) \in \mathbb{R}^{n\times(p+1)}$ denote the data matrix, where $\mathbf{y}$ is the response vector and $\mathbf{X}$ is the design matrix with p predictors. We denote by $\mathbf{z}_j$ the j-th column of $\mathbf{Z}$. The

covariance matrix of $\mathbf{Z}$ is $\mathbf{\Sigma}$, with $\mathbf{R}$ the corresponding correlation matrix, and $\mathbf{S} = \text{Diag}(\sigma_{z_1}, \ldots, \sigma_{z_{p+1}})$ the diagonal scale matrix containing the scale parameters of each variable z_j. Their estimates are denoted by $\hat{\mathbf{\Sigma}}$, $\hat{\mathbf{R}}$, and $\hat{\mathbf{S}}$, respectively.

A robust and efficient estimator of correlation used here is the *Gaussian rank (GR) correlation*, computed as follows. For each observation z_{ij} of variable j, compute the normal scores:

$$\tilde{z}_{ij} = \Phi^{-1}\left(\frac{\text{Rank}(z_{ij})}{n+1}\right),$$

where $\text{Rank}(z_{ij})$ is the rank of z_{ij} among all n observations of variable j, Φ is the standard normal cumulative distribution function (CDF), and Φ^{-1} is its inverse (quantile function). This transformation produces a pseudo dataset $\tilde{\mathbf{Z}} = (\tilde{z}_{ij}) \in \mathbb{R}^{n\times(p+1)}$.

The GR correlation matrix $\hat{\mathbf{R}}$ is then obtained as the Pearson correlation matrix of $\tilde{\mathbf{Z}}$. The GR correlation matrix is positive semi-definite (PSD) by construction, even in high-dimensional settings, ensuring convexity in subsequent optimization problems.

Robust scale estimates $\hat{\sigma}_{z_j}$ are obtained using the Q_n-estimator, which provides both robustness and efficiency. Combining these components, we define the robust covariance estimator:

$$\hat{\mathbf{\Sigma}} = \hat{\mathbf{S}}\hat{\mathbf{R}}\hat{\mathbf{S}}.$$

A.3.2 Robust Regression and Variable Selection via GR-Adaptive Lasso

Consider the linear regression model:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

Using the robust covariance estimate $\hat{\mathbf{\Sigma}}$, partitioned as

$$\hat{\mathbf{\Sigma}} = \begin{pmatrix} \hat{\Sigma}_{yy} & \hat{\mathbf{\Sigma}}_{xy}^\top \\ \hat{\mathbf{\Sigma}}_{xy} & \hat{\mathbf{\Sigma}}_{xx} \end{pmatrix},$$

we express the robust quadratic loss function for estimating $\boldsymbol{\beta}$ as:

$$\hat{\boldsymbol{\beta}}^{\text{GR}} = \arg\min_{\mathbf{b}} \left\{ n\hat{\Sigma}_{yy} + n\mathbf{b}^\top \hat{\mathbf{\Sigma}}_{xx}\mathbf{b} - 2n\mathbf{b}^\top \hat{\mathbf{\Sigma}}_{xy} \right\}.$$

Since $\hat{\mathbf{\Sigma}}$ is PSD, there exists a square root $\hat{\mathbf{\Sigma}}^{1/2}$ such that

$$\hat{\mathbf{\Sigma}}^{1/2} = (\mathbf{v}, \mathbf{W}),$$

where $\mathbf{v} \in \mathbb{R}^m$ is the first column and $\mathbf{W} \in \mathbb{R}^{m\times p}$ contains the remaining columns, satisfying

$$\mathbf{v}^\top\mathbf{v} = \hat{\Sigma}_{yy}, \quad \mathbf{W}^\top\mathbf{v} = \hat{\mathbf{\Sigma}}_{xy}, \quad \mathbf{W}^\top\mathbf{W} = \hat{\mathbf{\Sigma}}_{xx}.$$

This allows rewriting the loss function as the classic least squares problem

$$\hat{\boldsymbol{\beta}}^{\text{GR}} = \arg\min_{\mathbf{b}} \|\mathbf{v} - \mathbf{W}\mathbf{b}\|_2^2.$$

To induce sparsity for variable selection, we introduce the adaptive Lasso penalty weighted

by initial estimates $\tilde{\beta}_j$:

$$\hat{\boldsymbol{\beta}}^{\mathrm{GR-ALasso}} = \arg\min_{\mathbf{b}} \left\{ \|\mathbf{v} - \mathbf{W}\mathbf{b}\|_2^2 + \lambda \sum_{j=1}^{p} \omega_j |b_j| \right\},$$

where the weights are set as

$$\omega_j = \frac{1}{|\tilde{\beta}_j|},$$

and $\tilde{\boldsymbol{\beta}}$ is an initial robust estimator, for example,

$$\tilde{\boldsymbol{\beta}} = \begin{cases} \hat{\boldsymbol{\Sigma}}_{xx}^{-1} \hat{\boldsymbol{\Sigma}}_{xy}, & p \ll n, \\ (\hat{\boldsymbol{\Sigma}}_{xx} + \kappa \mathbf{I})^{-1} \hat{\boldsymbol{\Sigma}}_{xy}, & \text{high-dimensional regime,} \end{cases}$$

with $\kappa > 0$ a regularization parameter.

Finally, the intercept is estimated robustly by

$$\hat{\beta}_0 = \hat{\mu}_y - \hat{\boldsymbol{\mu}}_x^\top \hat{\boldsymbol{\beta}},$$

where $\hat{\mu}_y$ and $\hat{\boldsymbol{\mu}}_x$ are robust location estimates of $\mathbf{y}$ and $\mathbf{X}$, respectively.
To return to the Problem Formulation, Section 3, click here.