Regression with right-censored high-dimensional data: An application with different imputation techniques

Ersin Yilmaz[1,*], Dursun Aydin[1], S. Ejaz Ahmed[2]
[1]*Dept. of Statistics, Mugla Siki Kocman University, Mugla, Turkey*
[2]*Dept. of Mathematics & Statistics, Brock University, St. Catharines, Canada*
**Corresponding author: ersinyilmaz@mu.edu.tr*

Abstract

This study aims to introduce four modified linear estimators for the right-censored high-dimensional data. Obviously, data of interest involves two important problems to be solved that are censorship and high dimensionality. This paper can be distinguished from other studies in the literature with that it achieves to handle these two problems simultaneously. The main contribution of the paper is merging weighted-ridge method with the imputation techniques to obtain more efficient estimators than its alternatives. To solve the censorship problem, four imputation techniques are considered based on machine learning algorithms kNN, sliding-windows, regression and support vector machines. The high-dimensionality problem is handled by the weighted ridge approach which provides estimator with less risk than its alternatives because it detects the covariates with a weak contribution via the post-selection procedure. To show the empirical performance of the introduced estimators, a simulation study is made and comparative results are presented. Results show that kNN and regression imputation basis WR esitmators show satisfying performances on estimation of the high-dimensional right-censored model.

Keywords: High-dimensional data; kNN imputation; machine learning; right-censored data; sliding-windows

1. Introduction

High-dimensional data (HDD), which is one of the important sub-titles of Big Data phenomenon, has recently attracted great attention in many fields of science, depending on technological developments. For instance, in Biology or Bioinformatics, new sequencing techniques allow extracting the data for all molecular levels, such as mRNA or DNA sequences (see (Dehmer *et al.*, 2011); (Rao *et al.*, 2019)). In addition, HDD can be encountered in other research areas such as signal processing ((Gavish *et al.*, 2010)), Finance and economy ((Dang *et al.*, 2015); (Abonazel & Rabie, 2019)) and especially in medical studies ((Goh *et al.*, 2019); (Dondelinger *et al.*, 2020)). The most interested data type in the medical applications is gene expression microarray data. This kind of data involve much larger number of variables (p) than the sample size (n). Note that the key idea in analyzing the microarray data is to consider the number of "important" variables (genes) that are assumed as smaller than n. Usually, gene expression data include orthologous genes that have high sequence similarity because of repeated runs of amino-acids, pseudo genes and so on (Keith, 2008). In addition, to extract the expression data is an expensive procedure which makes hard to repeat it. Therefore, the extracted datasets involve the incomplete (censored) or missing data points mostly. Thus, modelling the gene expression data may cause the biased and widely variated estimations. Accordingly, researchers across the datasets with two issues to be solve that are high-dimensionality in explanatory variables and the censorship in the response variable. This matter indicates the crucial importance of the variable selection in high-dimensional data estimation and to solve the censorship problem in regression models.

The main objective of the variable selection or regularization methods (penalty functions) is to decide which variables are important and which are not. Thus, more stable, and interpretable model and estimators can be obtained. Let us consider the high-dimensional linear model as follows for completely observed response variable:

$$y_i = \sum_{j=1}^{p_n} \chi_{ij}\beta_j + \varepsilon_i, 1 \le i \le n \tag{1}$$

where y_i's are the uncensored response values, x_{ij}'s are the values of p_n predictors that form high-dimensional ($p_n >> n$) explanatory variables ε_i's have zero mean and constant variance σ_{ε^2}. Note that p_n denotes that change of p may be dependent to n which affects the asymptotic properties of estimators (see (Lei *et al.*, 2018). In this paper, we are interested in estimating the regression coefficients of model (1) when the observations of response variable are incompletely observed and right-censored by a random censoring variable c_i, but x_{ij}'s are completely observed. Therefore, instead of observing the values of response variable y_i, we observe the dataset (z_i, δ_i) with

$$z_i = \min\left(y_i, c_i\right), \delta = \begin{cases} 1 & y_i < c_i \\ 0 & y_i > c_i \end{cases} \tag{2}$$

where z_i's are incomplete response observations and δ carries the censorship existence information. If data point is censored $\delta = 0$ and $\delta = 1$ otherwise. In this case, model (1) transforms into a linear model with right-censored data, which can also be updated in terms the values of new response variable z_i. By using (2), right-censored high-dimensional model is given in (3):

$$z_i = \sum_{j=1}^{p_n} x_{ij}\beta_j + \varepsilon_i, 1 \le i \le n \tag{3}$$

Note that, censorship problem is mostly ignored by the researchers by eliminating them from the dataset or assuming all data points are completely observed. However, in medical research which is a highly sensitive field because it focuses on the human health, bias, and high-variance due to the right-censored data cause the unreliable estimates and interpretations.

There is a rich literature on estimating model (1) based on gene expression data (Segal *et al.*, 2004). Due to high-dimensional $p_n >> n$ nature of the data, they introduced regularized linear regression procedure based on Lasso (Tibshirani, 1996). Note that there are number of penalty functions that have high potential on estimating regression models for the microarray data such as Elastic Net (Zou, 2005), smoothly clipped absolute deviation (SCAD) proposed by (Fan, J., & Li, R., 2011), minimum concave penalty (MCP) defined by (Zhang, 2010) and their modifications. Some of these methods have been adapted to microarray data applications. For example, (Zou, 2005) used ElasticNet approach, (Kim *et al.*, 2009) used SCAD function, (Huang *et al.*, 2011) used MCP function to analyze the microarray data under high-dimensional settings. Note that the mentioned studies provide the linear estimators based on commonly used variable selection methods such as Lasso, SCAD or MCP. Although the mentioned penalties such as SCAD, MCP and Lasso-type functions have widely used in modelling the HDD owing to their feasible performances and computational easiness, they have a restrictive assumption in model design which affects the consistency and accuracy of the estimated model. This assumption is that regression coefficients of p_n predictors are formed by two subsets S_1 and S_2 where S_2 involves sparse part of the model with $\{\beta_j = 0\}_{j=1}^{p_{n_0}}$ (no signal) and S_1 involves the non-zero coefficients with $\{\beta_j \neq 0\}_{j=1}^{p_{n_1}}$ (strong signal). This restriction brings some other assumptions about the consistency of estimators obtained based on the penalty functions. On the other hand, (Gao *et al.*, 2016) introduced a new approach called the weighted ridge method (WR), which includes an important innovation for the estimation of the high-dimensional linear model. Unlike existing variable selection methods, it divides estimators into three subgroups, S_1 (strong signals) , S_2 (weak signals) and S_3 (no signals-sparse). As can be seen, WR takes into account weak signals, which provide a less risky prediction. In addition, this is one of main contributions of the paper using the advantage of WR in right-censored dataset.

On the other hand, to solve censorship problem, they preferred the cox hazard model or synthetic data transformation method. Solutions provide also satisfying estimates, but they manipulate the data structure. For instance, synthetic data transformation gives the right-censored data points zero and changes the magnitude of remaining data points (see (Aydin & Yilmaz, 2018)). This study aims to avoid this issue by using four imputation techniques based on kNN, sliding-windows (SW), regression (RI) and support vector machine-basis (SVMI) algorithms. Note that the mentioned imputation techniques are recently used in the literature (see (Malarvizhi & Thanamani, 2012) for kNN imputation, (Emmanuel *et al.*, 2021) for SVMI, (Doreswamy & Manjunatha, 2017) for RI) and developed by the compilation of the missing data in general. In this paper, those methods are adapted to the right-censored data and the modelling procedure. Thus, raw data can be directly on resolving censorship.

The main purpose of this paper is introducing the four linear estimators to estimate the right-censored high-dimensional linear model (1). To achieve this purpose, weighted-ridge (WR) approximation is used as a solution of high dimensionality. Thusly, using ridge penalty and lasso-type penalty, new estimators are introduced for components of (1) where ridge penalty provides to construct a "*data-adaptive post selection shrinkage estimator (PSE)*" as in mentioned by (Gao *et al.*, 2016). Also, four different imputation techniques that are kNN and sliding-windows (SW), regression and support vector machine-based imputations are considered to handle the right-censored data. Note that, the most important motivation of this paper is reducing the risk in high-dimensional data modeling which has a crucial importance in medical studies. From our knowledge high-dimensional right-censored data has no been modelled yet by the mentioned estimators in the literature.

Remain of the paper is arranged as follows. Section 2 introduces the four imputation techniques for the right-censored data. Imputation techniques are described with details. Then, linear estimators based on WR approach are explained. Sections 3 involves the simulation study and the obtained results. Finally, conclusions are given in Section 4.

2. Material and methods

2.1 Right-censored data

Let assume that F, G and J are the conditional distribution functions of variables y, c and $z \in R^+$ or given value of fixed covariate $X = x$, respectively. Let r be a positive constant, from that the mentioned distributions can be written as

$$\begin{gathered} F(r \mid X = x) = P(z \leq r \mid X = x), \quad G(r \mid X = x) = P(c \leq r \mid X = x) \\ J(r \mid X = x) = P(y \leq r \mid X = x) \text{ for } r \in \mathbb{R}^+ \end{gathered} \tag{4}$$

Due to right-censored response variable y, data pairs to be analyzed $(x_i, y_i)_{i=1}^n$ turn into data triplets $(x_i, z_i, \delta_i)_{i=1}^n$. It is necessary to add the censorship effect on the estimation process. Also, relationship between the survival functions of the mentioned variables is given by:

$$[1 - J(r \mid X = x)] = [1 - F(r \mid X = x)].[1 - G(r \mid X = x)] \tag{5}$$

To make the estimated model identifiable, there are two critical assumptions to be ensured related with (5) that are given as follows:
A1. Censoring variable c is independent from (x, y)
A2. $P(y \leq c \mid y, x) = P(y \leq c \mid y)$

Note that A1 and A2 are known as general assumptions in the random right-censored models (see, (Stute, 1993) for details). Because of the censoring, the ordinary estimation methods (such as least squares or maximum likelihood etc.) for estimating model (3) cannot applied directly. In the literature, to overcome the censored observations, different data transformation techniques ((Koul *et al.*, 1981)) or weighted least squares ((Orbe *et al.*, 2003)) are considered. As mentioned before, these methods touch the structure of all the data points. On the other hand, this study aims to impute the censored observations without manipulating the data. Accordingly, four different machine learning algorithms are considered to achieve this purpose that are explained in the following subsections.

2.2 kNN imputation

This section describes the kNN imputation method. It provides reasonable estimates for the right-censored data points without theoretical restrictions. In this paper, kNN imputation summarized and provides an algorithm. All the details about the method can be seen in (Ahmed *et al.*, 2019).

The kNN imputation has an advantage which is it can be used for both discrete and continuous variables. For discrete variables, the most frequently used value among k-nearest neighbors is determined as an imputed value. Mean value of k-nearest neighbors is used if the variable of interest is continuous. This is one of the important advantages of the method. Basically, the kNN is a similarity-based machine learning method which depends on the distance between data points. Therefore, similartiy measure affects the results seriously. In the litereature, generally, the Euclidean norm is used to evaluate the distances proposed by (Strike, 2001). The Euclidean norm can be computed as follows:

$$m_E(x,z) = \sqrt{\sum_{i=1}^{n} (x_i - z_i)^2} \tag{6}$$

where $m_E(x,z)$ represents the function of distance measure. To obtained the imputed values of the right-censored data points in the response variable y_i, this paper considers the algorithm proposed by (Ahmed *et al.*, 2019) which is given in Algorithm 1:

Algorithm 1 Algorithm for imputed kNN

Input: Right-censored dataset z_i, Censoring indicator δ_i, umber of nearest neigbours k, Values of predictor variable x_i (high-correlated one with y_i)
Output: Imputed dataset $\mathbf{y}^{knn} = \left(y_1^{knn}, \ldots, y_n^{knn}\right)^T$
Begin
for $(i = 1 : n)$ **do**
If $(\delta_i = 0)$ **do** (if data point is censored)
for $(j = 1 : n)$ **do**
Find the Euclidean distances given in (4) between x_j and x_i for each censored data point
Sort the distances from small to large
for $(j = 1 : k)$ **do**
end
Take the first uncensored k values of z_i associated to sorted distances
Calculate the i^{th}imputed value (y_i^{knn}) with average of nearest k records of y_i
Replace the imputed values (y_i^{knn}) with censored data points $(z_i, \delta_i = 0)$ in censored data set $\mathbf{Z} = (z_1, \ldots, z_n)$
end
Return $\mathbf{y}^{knn} = \left(y_1^{knn}, \ldots, y_n^{knn}\right)^T$

It should be emphasized that neighbours of the instances may be right-censored which makes critical to determine both the number of neighbours "k" and their locations. (Cartwright *et al.*, 2004) suggested a low k (i.e., 1 or 2). However, to choose more efficient "k" it is selected from between interval of $[2, 10]$ that minimizes the mean squared error (MSE) score.

2.3 Imputation based on sliding-windows

In this section, the sliding-windows (SW) imputation method proposed by (Ahmed *et al.*, 2020) is introduced which is another censorship solution method. SW imputes the right-censored observations by a sliding window method based on predictive model. Note that in data science SW is of the important machine-learning methods especially in data mining applications. SW includes a fixed window size on the data points, and it works locally with the data points placed in the specified window then moves to the next window. The main advantage of the SW from the other imputation methods is its local operation feature which makes it superior the SW for the datasets with unstable variances. It is works together with the linear regression model by OLS to estimate the right-censored data point with in-sample prediction. SW imputation is summarized as follows.

Let assume the following notations.

- w: window size for SW
- t: window of interest (t^{th} window)
- $\mathbf{Z}_t^*$: Vector of response variable for t^{th} window
- $\mathbf{X}_t^*$: Matrix of explanatory variables for t^{th} window

The number of windows (n_w) changes depends on the window size (w) which is computed by $n_w = (n - w + 1)$. Note that it is substantial to determine the accurate window size (w). (Ahmed *et al.*, 2020) suggests that w changes according to censoring level. "*When the censoring level increases, "w" gets small values and takes large values otherwise*" they mentioned. When the necessary parameters are decided for the SW, OLS can be applied by the subsets in each window. Accordingly, SW model can be given by:

$$\mathbf{Z}_t^* = \mathbf{X}_t^{*T}\boldsymbol{\theta}_t + \boldsymbol{\varepsilon}_t, t = 1, 2, \ldots, n_w \tag{7}$$

where $\boldsymbol{\theta}_t = (\theta_{1t}, \theta_{2t}, \ldots, \theta_{pt})^T$ vector of coefficients t^{th} window and $\boldsymbol{\varepsilon}_t \sim N\left(0, \sigma_t^2\right)$. Hence, estimation of $\boldsymbol{\theta}_t$ is obtained in the equation (8)

$$\widehat{\boldsymbol{\theta}}_t = \left(\mathbf{X}_t^{*T}\mathbf{X}_t^*\right)^{-1}\mathbf{X}_t^{*T}\mathbf{Z}_t^* \tag{8}$$

and the fitted values are given by:

$$\widehat{\mathbf{Z}}_t^* = \mathbf{X}_t^{*T}\widehat{\boldsymbol{\theta}}_t = \mathbf{H}_t\mathbf{Z}_t^* \tag{9}$$

where $\mathbf{H}_t = \mathbf{X}_t^*\left(\mathbf{X}_t^{*T}\mathbf{X}_t^*\right)^{-1}\mathbf{X}_t^{*T}$. Thusly, imputation for the right-censored observations placed in the t^{th} window can be estimated by using (9). Detailed information can be seen from the attached algorithm.

Algorithm 2 SW imputation for right-censored data

1: **Input:** Right-censored data points z_i (obtained from equation (2)) Corresponding $\delta_i = I(y_i < c_i)$, Values of predictor variable x_i (high-correlated one with z_i), window size parameter w
2: **Output:** Imputed dataset $\hat{\mathbf{y}}^{\mathrm{sw}} = \left(y_1^{\mathrm{sw}}, y_2^{\mathrm{sw}}, \ldots, y_{n_c}^{\mathrm{sw}}\right)^T$
3: **Begin**
4: **for** $(i = 1 : n)$ **do**
5: **If** $(\delta_i = 1)$ **do**
6: obtain z_i^* with $z_i^* = z_i$
7: obtain x_i^* with $x_i^* = x_i$
8: **end**
9: Determine the number of windows (n_w) with $(n - w + 1)$
10: **for** $(j = 1 : n_w)$
11: Estimate the $\boldsymbol{\theta}_j^*$ for j^{th} window
12: Obtain the fitted values for the j^{th} window by $\hat{y}_j^{\mathrm{sw}} = \mathbf{X}_j^* \widehat{\boldsymbol{\theta}}_j$
13: **end**
14: **for** $(i = 1 : w)$ **do**
15: **if** $(\delta_i = 0)$ **do**
16: $z_i = y_i^{sw}$ (replacing the censored ones by the imputed ones)
17: **else** $(\delta_i = 1)$ **do**
18: $z_i = z_i^*$
19: **end** (for loop in Step 12)
20: **Return** $\hat{\mathbf{y}}^{\mathrm{sw}} = \left(y_1^{\mathrm{sw}}, y_2^{\mathrm{sw}}, \ldots, y_{n_c}^{\mathrm{sw}}\right)^T$
21: **end**

2.4 Regression imputation (RI)

Regression imputation uses the classical linear regression model estimated by the ordinary least squares (OLS) to make imputation Let assume that "m" be the number of the uncensored observations and $(z_i^{RI})_{i=1}^m$ be the value(s) of them. From that, regression model for the imputation can be given in equation (10):

$$z_i^{RI} = (\mathbf{X}_i^{RI})^T \boldsymbol{\eta} + \varepsilon_i^{RI}, i = 1, \ldots, m \tag{10}$$

where z_i^{RI} is the i^{th} value of the response variable, $\mathbf{X}_i^{RI}$ denotes the predictor variables, $\boldsymbol{\eta} = (\eta_0, \ldots, \eta_{p_{RI}})^T$ is the vector of regression coefficients and ε_i^{RI} $N(0, 1)$ is the random error terms for the RI model. The key idea of the RI method is to estimate $\boldsymbol{\eta}$ and *making in-sample predictions* for the right-censored data points. In this manner, similar to SW imputation method, equation (10) is estimated by OLS as follows:

$$argmin \left(\boldsymbol{\eta}^{RI}\right) = \sum_{i=1}^{n} \left(z_i^{RI} - \left(\mathbf{X}_i^{RI}\right)^T \boldsymbol{\eta}_i^{RI}\right)$$

$$\widehat{\boldsymbol{\eta}}^{RI} = \left(\left(\mathbf{X}^{RI}\right)^T \mathbf{X}^{RI}\right)^{-1} \left(\mathbf{X}^{RI}\right)^T \mathbf{z}^{RI} \tag{11}$$

Thus, by using $\boldsymbol{\eta}^{RI}$ and $\mathbf{X}_i^{RI}$, the imputed values can be obtained based on the censorship information provided by δ_i. Then, the imputed ones are replaced with the censored ones. Note that RI brings some extras about the imputed values with the magnitude and signs of the regression coefficients. As with other two imputation methods, an algorithm for RI is given in Algorithm 3.

Algorithm 3 RI imputation for right-censored data

Input: Right-censored data points z_i. Censoring indicator $\delta_i = I(y_i < c_i)$, Values of predictor variable x_i.
Output: Imputed dataset $\hat{\mathbf{y}}^{\mathrm{RI}} = \left(\hat{y}_1^{\mathrm{RI}}, \hat{y}_2^{\mathrm{RI}}, \ldots, \hat{y}_{n_c}^{\mathrm{RI}}\right)^T$
Begin
for $(i = 1 : n)$ **do**
If $(\delta_i = 1)$ **do**
obtain z_i^{RI} with $z_i^{RI} = z_i$
obtain x_i^{RI} with $x_i^{RI} = x_i$
end
Obtain the $\boldsymbol{\eta}$ from equation (11)
for $(i = 1 : n)$ **do**
If $(\delta_i = 0)$ **do**
Estimate i^{th} right-censored observation using estimated model
Replace the fitted value $(\hat{y}_i^{RI})$ with censored data point z_i.
end
Return $\hat{\mathbf{y}}^{\mathrm{RI}} = \left(\hat{y}_1^{\mathrm{RI}}, \hat{y}_2^{\mathrm{RI}}, \ldots, \hat{y}_n^{\mathrm{RI}}\right)^T$
end

2.5 Support Vector Machine-based imputation (SVMI)

SVM is one of the most commonly used machine learning algorithms to complete the missing data (see (Stewart *et al.*, 2018)). Note that SVMI is generally used to make imputation for the missing categorical variables not continuous data. Thusly, SVM classifier is preferred as a imputation tool. However, this paper modifies the SVM for the continuous right-censored data imputation by using censorship information and SVM regression estimator. Then it is integrated with the high-dimensional data modelling. Imputation procedure is explained with details in Algorithm 4. To make imputations, SVM regression is used and as in RI method, right-censored data points are imputed by in-sample predictions. In this manner, SVM regression is summarized as follows.

Let consider the training dataset of pairs as $\{(x_i, z_i)\}_{j=1}^{n} \in \mathbb{R}^n \times \mathbb{R}$ where x_i^* is the high-correlated covariate among p_n covariates in model (3) with right-censored response variable z_i to make more accurate imputation. As known, SVM considers the linear relationship between by solving the following regression function:

$$\mathbf{z} = \langle \mathbf{w}_s \cdot \mathbf{X}^* \rangle + \mathbf{b} \tag{12}$$

where $\mathbf{w_s}$ is the vector of gradient and $\mathbf{b}$ is the intercept term. In model (12), the objective function minimizes some error on the training set for a determenied loss function. Even if there other loss functions such as absolute loss, here, square loss function is used which can be given in equation (13):

$$L\left(\mathbf{z}_i, \hat{\mathbf{z}}_i\right) = \left(\mathbf{z}_i - \hat{\mathbf{z}}_i\right)^2 \tag{13}$$

By using (13) the objective function to minimized for the SVM regression can be written as:

$$J(\mathbf{w}_s) = \frac{1}{2}\mathbf{w}_s'\mathbf{w}_s + C_{svm}\sum_{j=1}^{n}\left(\xi_j + \xi_j^*\right) \tag{14}$$

where C is called as a box contraint, a positive numeric value that controls the penalty term which prevents the overfitting problem. ξ's are the slack variables to make possible the optimization. To save the space, all details of SVM regression cannot be given here. For further details see (Stewart *et al.*, 2018).

By using estimated model via minimizing equation (14) and obtaining gradients $\hat{\mathbf{w}}_s$, right-censored observations can be imputed by using the following algorithm. Thusly, imputed data set can be constructed.

Algorithm 4 SVMI imputation for the right-censored data

1: **Input:** Right-censored data points z_i. Censoring indicator $\delta_i = I(y_i < c_i)$, Values of predictor variable x_i, a training dataset without censored data points, a tolerance threshold and a maximum iteration number for the iterative process that are 10^{-5} and 200 respectively.
2: **Output:** Imputed dataset $\hat{\mathbf{y}}^{\mathrm{SVMI}} = \left(\hat{y}_1^{\mathrm{SVMI}}, \hat{y}_2^{\mathrm{SVMI}}, \ldots, \hat{y}_{n_c}^{\mathrm{SVMI}}\right)^T$
3: **Begin**
4: Estimate the SVM model by using training model from (14)
5: **for** $(i = 1 : n)$ **do**
6: **If** $(\delta_i = 0)$ **do**
7: Make in-smaple prediction for i^{th} right-censored observation by using $\hat{\mathbf{w}}_s$ and the SVM model.
8: Replace the fitted value $(\hat{y}_i^{SVMI})$ with censored data point z_i.
9: **end**
10: **Return** $\hat{\mathbf{y}}^{\mathrm{SVMI}} = \left(\hat{y}_1^{\mathrm{SVMI}}, \hat{y}_2^{\mathrm{SVMI}}, \ldots, \hat{y}_n^{\mathrm{SVMI}}\right)^T$
11: **end**

2.6 Procedure of weighted-ridge method

In this section, WR approach is summarized firstly with some important details and then, its integration to the right-censored responses is explained. Linear estimators based on the kNN and SW imputation techniques are obtained. At first, WR procedure is summarized. Details can be found in (Gao *et al.*, 2016).

As mentioned before, in WR approach works with three subsets $S \subset \{S_1, S_2, S_3\}$ that involve regression coefficients β_j's according to their signal strength. Basically, WR uses two penalties to obtain the estimators gradually. Firstly, using with Lasso, sparse and non-zero β_j's are obtained that can be expressed as $S \subset \hat{S}_1, \hat{S}_L$ where $\hat{S}_L \subset \hat{S}_2, \hat{S}_3$. Secondly, the post selection shrinkage is made for $\hat{S}_L$ to separate the weak signals ($\hat{S}_2$) from the sparse ones ($\hat{S}_3$) by ridge regression. Due to WR method, obtained estimators are expected to be more sensitive to taking account for the data structure.

To understand clearly and to make sense the separating the signals (regression coefficients) into three subsets S_1, S_2 and S_3 as mentioned above, some conditions need to be assumed that are explained by (Gao *et al.*, 2016) with details. Here, these conditions are summarized as follows:

(i) for given $\omega > 0, |\beta_j| > \omega\sqrt{(\log(p_n)/n}$ if $j \in S_2$

(ii) $\boldsymbol{\beta}$ should ensure that $\left\|\boldsymbol{\beta}_{S_3}\right\| = O\left(n^{\ell}\right)$, for $0 < \ell < 1$

(iii) $\beta_j = 0$ if $j \in S_1$

As usual in all variable selection methods in the literature, WR is adopted to the regression analysis with a penalty function which can be expressed briefly with "*Loss function + penalty function*". In this paper, loss function is determined as objective function of the penalized least squares (PLS) and the penalty function is chosen as WR. Basically, general minimization criterion can be given by:

$$\left\{\widehat{\beta}_J\right\} = \arg\min_{\beta\in\mathbb{R}^p}\left\{z_i - \sum_{j=1}^{p_n} x_{ij}\beta_j\right\}^2 + \sum_{j=1}^{p_n} p_{\lambda_r}(\beta_j) \tag{15}$$

where $\lambda_r > 0$ is a shrinkage parameter for the WR penalty which controls the shrinkage level. Note that in this section, we focused on the WR penalty function $\sum_{j=1}^{p_n} p_{\lambda_r}(\beta_j)$. As known $\sum_{j=1}^{p_n} p_{\lambda_r}(\beta_j)$ is a quite common notation to show the penalty function. For instance, it takes $\lambda_{\text{lasso}} \sum_{j=1}^{p_n} |\beta_j|$ for Lasso which is used in this paper to obtain the subset of strong signals $\hat{S}_1$. The selection of the shrinkage parameter λ_r has a crucial importance on estimation procedure. Therefore, cross-validation (CV) criterion is used to decide λ_r which is one of the most widely used methods for the regularization parameter selection (see (Lukas, 1993); (Jung *et al.*, 2018)).

WR works with two stages. At the first place, the subset of weak signals S_2 is ignored and a model is obtained which includes only the strong signals ($\beta_j \in S_1$) and if $j \notin S_1$, it is decided $hatbeta_j = 0$. Accordingly, by using penalized least squares (PLS), restricted least squares estimator (RE) is obtained as follows:

$$\widehat{\boldsymbol{\beta}}_{S_1}^{RE} = \left(\mathbf{X}_{S_1}^T \mathbf{X}_{S_1}\right)^{-1} \mathbf{X}_{S_1}^T \mathbf{Z} \tag{16}$$

The focused point by using WR approach is to reduce the risk of the estimator given in (9). To achieve that it is needed to use information from the subset S_1^c that means to add some weak signals into the model. In this context, let assume that $\mathbf{X} = (\mathbf{X}_{S_1} | \mathbf{X}_{S_2} | \mathbf{X}_{S_3})$ and corresponding regression coefficients are $\boldsymbol{\beta} = \left(\boldsymbol{\beta}_{S_1} \left| \boldsymbol{\beta}_{S_2} \right| \boldsymbol{\beta}_{S_3}\right)^T$. To make simple to understand, the following notations are used; $s(S_1) = p_1, s(S_2) = p_2$ and $s(S_3) = p_3$. From that $p = p_1 + p_2 + p_3$. Also, it is important to mentioned the restriction of $h = p_1 + p_2 \leq n$ with $\mathbf{R} = (\mathbf{X}_{S_1}, \mathbf{X}_{S_2})$. Here, $\boldsymbol{\Sigma} = n^{-1}\mathbf{R}^T\mathbf{R}$ is an inversible matrix. By using the given information, three steps to obtain the WR estimator and $\widehat{\boldsymbol{\beta}}^{PSE}$ can be given below in three steps:

Table 1. Estimation steps of $\widehat{\boldsymbol{\beta}}^{PSE}$ based on WR approach

Step 1. Obtain the subset $\hat{S}_1$ by using Lasso and $\hat{\beta}_{\hat{S}_1}^{RE}$ given in (9).
Step 2. Obtain $\widehat{\boldsymbol{\beta}}^{WR} = \left(\widehat{\boldsymbol{\beta}}_{\hat{S}_1}^{WR}, \widehat{\boldsymbol{\beta}}_{\hat{S}_1^c}^{WR}\right)$ based on WR penalty threshold and $\hat{S}_1$ which found in Step 1.
Step 3. Obtain the post selection shrinkage estimator $\widehat{\boldsymbol{\beta}}^{PSE}$ by shrinking the $\widehat{\boldsymbol{\beta}}^{WR}$ estimated in Step 2.

Note that $\widehat{\boldsymbol{\beta}}^{PSE}$ can eliminates the three main problems in high-dimensional data modeling that are i) Extracting the sparse signals, ii) Eliminating the multi-collinearity, iii) Adding the weak signals to the model. WR approach can solve the mentioned (i-iii) problems with in Steps 1-3.

A short algorithm is provided in Algorithm 5 for WR approach and obtain the $\widehat{\boldsymbol{\beta}}^{PSE}$. To see further discussion, see (Gao *et al.*, 2016). After the algorithm, $\widehat{\boldsymbol{\beta}}^{PSE}$ is integrated with the imputed response variables obtained from kNN and SW techniques.

Algorithm 5 Estimate model (3) with WR approach

1: **Input:** Response variable z_i (y_i^{kNN} or y_i^{sw}), high-dimnesional covariate matrix $\mathbf{X} \in \mathbb{R}^{n \times p_n}, p_n \gg n$

2: **Output:** $\widehat{\boldsymbol{\beta}}^{PSE}$

3: Begin

4: Minimize the $\widetilde{\boldsymbol{\beta}}(r) = \arg\min_{\beta} \left\{ \|\mathbf{Z} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \left\| \boldsymbol{\beta}_{\hat{S}_1^c} \right\|^2 \right\}$ for obtained $\hat{S}_1$ and $\hat{S}_1^c$

5: Obtain the WR estimators based on λ and WR threshold α based on $\hat{S}_1$ as follows:

$$\hat{\beta}_j^{WR}(\lambda_r, \alpha) = \begin{cases} \tilde{\beta}_j(\lambda_r), & j \in \hat{S}_1 \\ \tilde{\beta}_j(\lambda_r) I\left(\tilde{\beta}_j(\lambda_r) > \alpha\right), & j \in \hat{S}_1^c \end{cases}$$

6: Based on the WR threshold α, obtain the subset of weak signals $\hat{S}_2$ by

$$\hat{S}_2 := \hat{S}_2\left(\hat{S}_1\right) = \left\{ j \in \hat{S}_1^c : \beta_j^{WR}(\lambda_r, \alpha) \neq 0 \right\}$$

7: Obtain the subset of sparse signals $\hat{S}_3$ as $\hat{S}_3 := \hat{S}_3\left(\hat{S}_1\right) = \left(\hat{S}_1 \cup \hat{S}_2\right)^c$

8: Compute the necessary arguments:

$$\widehat{T}_r = \left(\widehat{\boldsymbol{\beta}}_{\hat{S}_2}^{WR}\right)^T \left(\mathbf{X}_{\hat{S}_2}^T \mathbf{M}_{S_1} \mathbf{X}_{\hat{S}_2}\right) \widehat{\boldsymbol{\beta}}_{\hat{S}_2}^{WR} / \sigma^2 \text{ and } \mathbf{M}_{S_1} = \mathbf{I}_n - \mathbf{X}_{\hat{S}_1}\left(\mathbf{X}_{\hat{S}_1}^T \mathbf{X}_{\hat{S}_1}\right)^{-1} \mathbf{X}_{\hat{S}_1}^T$$

9: Obtained estimator $\widehat{\boldsymbol{\beta}}^{PSE}$ based on $\hat{\beta}_j^{WR}(\lambda_r, \alpha)$ and the matrices in step 6 as follows:

$$\widehat{\boldsymbol{\beta}}^{PSE} = \boldsymbol{\beta}_{\hat{S}_1}^{WR} - \left(\left[\left(\left|\hat{S}_2\right| - 2\right) / \hat{T}_r\right] \wedge 1\right)\left(\boldsymbol{\beta}_{\hat{S}_1}^{WR} - \boldsymbol{\beta}_{\hat{S}_1}^{RE}\right)$$

Note that, the threshold α which is used in step 3 of Algorithm 5, is needed to ensure the following condition:$\left|\hat{S}_2\right| = s\left(\hat{S}_2\right) > 2$ and $\left|\hat{S}_3^c\right| = s\left(\hat{S}_3^c\right) < n$ According to that its calculation is given by $\alpha = \vartheta n^{-d}, 0 < d \leq 0.5, \vartheta > 0$. Based on the $\widehat{\boldsymbol{\beta}}^{PSE}$ fitted values for the determined model are calculated as follows by using matrix $\mathbf{X}^* = [\mathbf{X}_{S_1} \mathbf{X}_{S_2}]$

$$\widehat{\boldsymbol{\mu}}_{S_3^c} = \widehat{\mathbf{Z}} = \mathbf{X}^* \hat{\boldsymbol{\beta}}^{PSE} \tag{17}$$

As can be seen, Algorithm 5 is used incomplete response variable z_i as input argument. However, as we mentioned before, z_i cannot be used directly in the estimation process. To overcome this issue, four imputation techniques are introduced in in Section 2. In this manner, instead of z_i imputed response variables should be used as input in Algorithm 5. If y_i^{knn} is used as a response variable, $\widehat{\boldsymbol{\beta}}^{PSE}$ is obtained based on kNN imputation method. Similarly, y_1^{sw} gives SW based estimator $\widehat{\boldsymbol{\beta}}^{PSE}$ and the procedure is same for the RI and SVMI methods.

3. Simulation study

This section provides a design and results of the detailed simulation experiments to show performances of the introduced four linear estimators for right-censored high-dimensional data. Note that simulation study is realized with R-software. Simulation design, data generation and model settings are summarized as follows:

Data Generation: Regarding to model (1), each element of the model obtained as follows:

$$\mathbf{X}_i \sim MN\left[\boldsymbol{\mu}_{p_n \times 1}, \boldsymbol{\Sigma}_{p_n \times p_n}\right] \text{ and } \varepsilon_i \sim N\left(\mu_\varepsilon = 0, \sigma_\varepsilon^2 = 0.5\right)$$

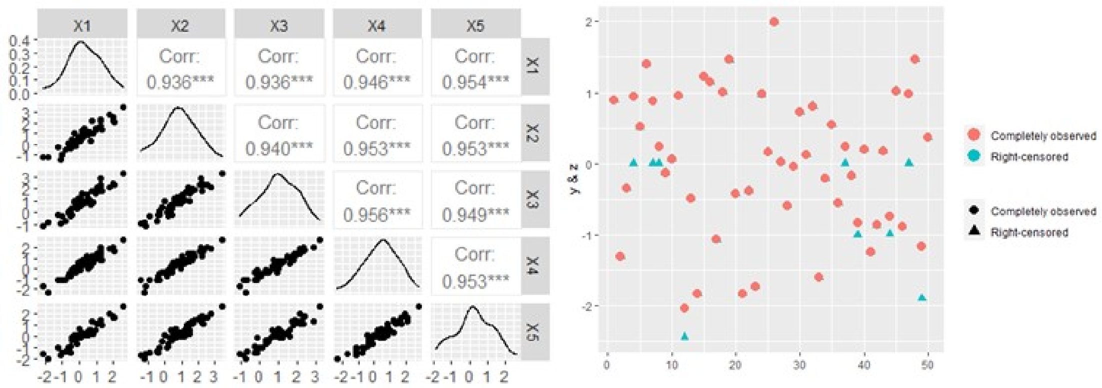
The true vector of regression coefficients are given by:

$$\beta_j = \begin{cases} -5 & \text{if } j = 1, 2, 3, 4, 5 \\ 5 & \text{if } j = 11, 12, 13, 14, 15 \\ 0.2 & \text{if } j = 21, 22, 23, 24, 25 \\ 0 & \text{otherwise} \end{cases}$$

Thus, it can be said that there are 15 signals to be estimated and for each sample size there are $(n-15)$ sparse signals.

Regarding the censoring data, censoring variable c_i is generated as $c_i \sim N\left(\mu_y, \sigma_y^2\right)$ independently of the initially observed variable y_i. Hence, partially observed responses are obtained with $z_i = min(y_i, c_i)$. An algorithm for censoring procedure is provided by (Aydin *et al.*, 2021).

To show the multicollinearity and censorship problems in the generated datasets, Figure 1 is presented which is formed by two panels. In panel (a), collinearity can be seen obviously and in panel (b), right-censored responses are indicated with blue "Δ" when $CL = 25\%$.



(a) Correlation plot for high-correlated 5 covariates

(b) Scatterplot for the right-censored response variable (z_i) and completely observed (y_i)

Fig. 1. Multicollinearity problem (a) and right-censored responses (b) in the generated data

Simulation Design:The sample size was determined as $n = 50, 100, 150$ and 200 and the number of variables as $(p_n = 200, 300, 500)$. Accordingly, it is planned to examine different $p >> n$ states. Data is produced by using the correlation coefficient $\rho = 0.95$ for explanatory variables in order to show the multicollinearity problem, which is frequently encountered in microarray data. However, these correlations were applied for a certain number of variables, not for each variable. This number is intuitively set to 5 which will be enough to emerge the multicollinearity problem the generated datasets. Also, all the simulations are realized for the two censoring levels $CL = 5\%, 10\%, 15\%$ and 25%. Each simulation was repeated 1000 times.

Performance of the methods are evaluated by mean square error (MSE) of the model based on the fitted values given in (10) and relative mean squared error (ReMSE) for estimated regression coefficients that can be computed as follows:

$$\begin{aligned} \text{MSE}(\widehat{\mathbf{z}}) &= n^{-1}\sum_{i=1}^{n}(z_i - \hat{z}_i)^2 = (\mathbf{z} - \widehat{\mathbf{z}})^T(\mathbf{z} - \widehat{\mathbf{z}}) \\ \text{ReMSE}\left(\widehat{\boldsymbol{\beta}}_{kNN}, \widehat{\boldsymbol{\beta}}_{SW}\right) &= \frac{\left(\widehat{\boldsymbol{\beta}}_{kNN} - \widehat{\boldsymbol{\beta}}_{SW}\right)'\left(\widehat{\boldsymbol{\beta}}_{kNN} - \widehat{\boldsymbol{\beta}}_{SW}\right)}{\left(\widehat{\boldsymbol{\beta}}_{SW} - \boldsymbol{\beta}\right)'\left(\widehat{\boldsymbol{\beta}}_{SW} - \boldsymbol{\beta}\right)} \end{aligned} \tag{18}$$

where $\widehat{\boldsymbol{\beta}}_{kNN}$ and $\widehat{\boldsymbol{\beta}}_{sw}$ are the estimated coefficients by the WR approach in Algorithm 2. Details are discussed in Section 2. If $ReMSE > 1$, it means $\widehat{\boldsymbol{\beta}}_{sw}$ gives better estimates than $\widehat{\boldsymbol{\beta}}_{kNN}$ and vice versa.

Also, to compare the imputation methods individually, averaged-bias (AvB) for the imputed values, and inaccuracy (IA) measure are used that are given by:

$$AvB = \frac{1}{n_{\text{cens}}} \sum_{i=1}^{n_{\text{cens}}} \left| y_i - y_i^{imp} \right|, IA = \frac{1}{n_{\text{cens}}} \sum_{i=1}^{n_{\text{cens}}} \frac{\left| y_i - y_i^{imp} \right|}{y_i} \tag{19}$$

where n_{cens} is the number of censored data points, y_i^{imp} denotes the imputed data points any of kNN or SW methods. Note that these scores can be computed for only the simulation experiments because both real and censored responses are known. Before the estimation of the model, imputed response variables are obtained from the kNN and SW imputation methods. Selection of window size for SW and number of neighbors for kNN imputations are determined by using mean squared error (MSE) imputed values. Example plots are given in Figure 2 for all possible simulation combinations.

Figure 2 shows the optimal No. neighbors for the kNN. In a similar manner, window size for SW imputation method is decided optimally as seen in Figure 2. Note that, optimum values of w and k are provided in Tables 2-7 and, for each configuration, optimal values of them are determined before the model estimation. Imputation performances are clearly seen in Tables 2-7 for all possible simula-

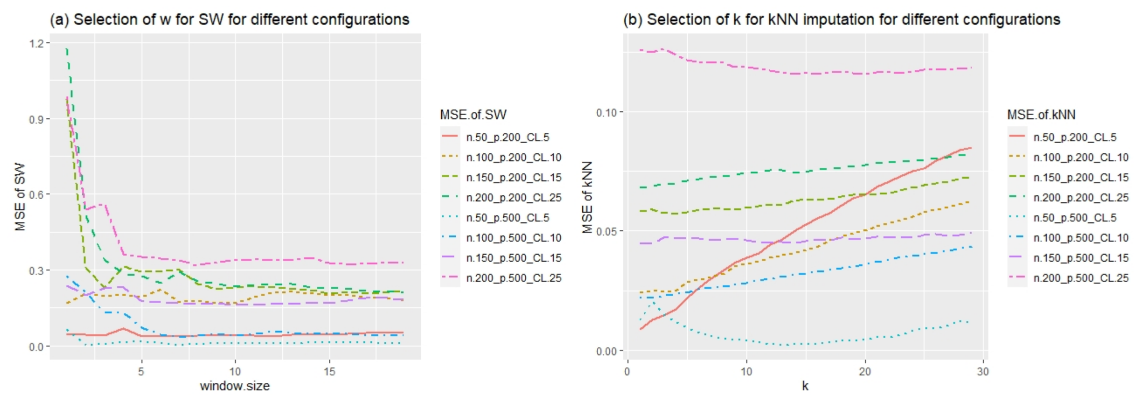


Fig. 2. Selection of window size (w) of SW imputation (panel (a)) and no. neighbors (k) of kNN imputation (panel (b)).

tion configurations. At the first look, kNN imputation method achieves the impute the right-censored observations quite better than other three imputation methods. In detail, for large number of covariate ($p_n = 500$) RI technique shows better performance than others regarding the AvB and IA scores. Because of there is no distributional assumption for kNN, its performance is not affected by the sample size which can be counted as both advantage and disadvantage. From a positive aspect, it can give satisfying results for small sample sizes even $CL = 25\%$ such as $n = 50, p = 500$ and $CL = 25\%$ configuration. On the other hand, it cannot be said that when the sample size is getting larger, the performance of kNN is getting better due to its nonparametric nature. Regarding the SW, RI and SVM, they work based on the least squares method. Therefore, as can be seen from the tables, their performances are getting better when sample size is getting large in contrary to kNN imputation. if IA scores in Tables 2-7 inspected carefully, it is obvious that in most of the cases, imputation methods give closer values. Although SW, RI and SVM cannot give good performances on this simulation study, They show more stable and predictable performances than kNN imputation After the performances of the imputation methods, estimation of the model based on WR can be applied by using obtained imputed response variables $\mathbf{y}_{kNN}$, $\mathbf{y}_{SW}$, $\mathbf{y}_{RI}$ and $\mathbf{y}_{SVMI}$. To achieve that estimation procedure given in Table 1 and Algorithm 5 are applied to the generated dataset.

At first, a candidate model and subset of strong signals $\hat{S}_1$ are obtained by Lasso. As known, Lasso is a shrinkage method and it shrinks the regression coefficients towards to zero by using a iterative

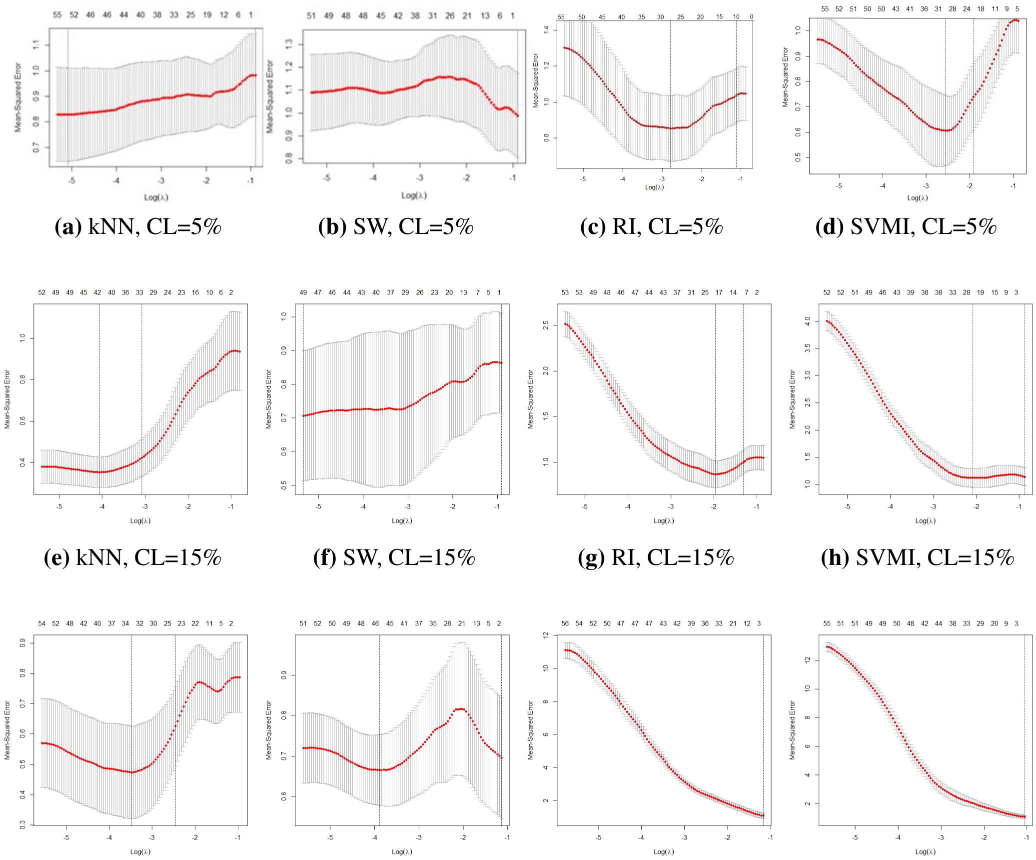
Table 2. AvB and IA scores for the imputation methods for CL=5% and CL=10%

CL	n		AvB				IA				Optimum	
		p_n	kNN	SW	RI	$SVMI$	kNN	SW	RI	$SVMI$	k	w
5%	50	200	**0.377**	0.591	0.717	0.777	**0.614**	0.921	1.038	1.307	11	12
		300	**0.374**	0.562	0.668	0.749	**0.558**	0.638	0.612	0.962	11	11
		500	**0.371**	0.548	0.612	0.722	**0.521**	0.580	0.593	0.963	10	11
	100	200	**0.443**	0.502	0.624	0.736	**0.435**	0.591	0.825	0.848	5	8
		300	**0.284**	0.476	0.489	0.587	0.624	**0.527**	0.730	0.808	16	12
		500	**0.433**	0.467	0.674	0.823	0.807	0.889	**0.708**	0.766	8	12
	150	200	**0.421**	0.425	0.423	0.575	**0.420**	0.532	0.693	0.819	11	13
		300	0.419	0.409	**0.345**	0.511	0.702	**0.409**	0.616	0.724	11	13
		500	0.412	0.402	**0.267**	0.447	0.677	0.786	**0.639**	1.729	11	13
	200	200	0.423	0.396	**0.188**	0.383	**0.378**	0.395	0.462	0.734	11	13
		300	0.429	0.384	**0.110**	0.319	0.505	0.408	**0.385**	0.730	11	13
		500	0.435	0.372	**0.032**	0.255	0.333	0.621	**0.208**	0.684	11	14
10%	50	200	**0.409**	0.615	0.802	0.821	**0.258**	1.174	1.774	1.478	5	8
		300	**0.351**	0.621	0.772	0.663	**0.341**	0.869	1.074	1.293	9	12
		500	**0.390**	0.604	0.727	0.665	**0.458**	0.773	1.049	1.028	5	13
	100	200	**0.481**	0.587	0.625	0.709	**0.310**	0.918	1.487	1.133	11	14
		300	**0.324**	0.582	0.521	0.670	**0.624**	0.833	0.991	1.015	12	15
		500	**0.487**	0.587	0.645	0.750	**0.495**	0.781	0.822	0.915	3	14
	150	200	**0.376**	0.495	0.535	0.652	**0.150**	0.732	1.067	1.042	8	14
		300	0.472	**0.421**	0.511	0.659	**0.468**	0.665	0.976	0.901	16	14
		500	**0.442**	0.468	0.621	0.733	**0.266**	0.639	0.951	0.965	10	18
	200	200	**0.458**	0.444	0.571	0.654	**0.551**	0.707	0.964	0.799	9	17
		300	**0.440**	0.441	0.601	0.685	**0.444**	0.644	0.817	0.787	14	17
		500	0.424	**0.401**	0.570	0.675	0.935	0.698	**0.688**	0.732	10	16

The best scores are indicated with bold color

process. In this process, shrinkage parameter of lasso $\lambda_{lasso} > 0$ has a crucial importance which is mentinoed before. As in selection of shrinkage parameter of WR, CV criterion is preferred to choose the λ_{lasso}. Figure 3 is drawn to show the implementation of the selection of the shrinkage parameter for two simulation configurations. Note that all configurations cannot be shown here due to space restrictions. For the remaining configurations, selection of λ_{lasso} are applied similarly.

After the determined the $\hat{S}_1$ and $\hat{S}_1^c$ via Lasso, as shown in Algorithm 5, estimated coefficients are seperated into three subsets by using both λ_r and α parameteres $\hat{S}_1, \hat{S}_2$ and $\hat{S}_3$. Table 4 provides the number of elements of the subsets $\hat{S}_1, \hat{S}_2$.



(a) kNN, CL=5% **(b)** SW, CL=5% **(c)** RI, CL=5% **(d)** SVMI, CL=5%

(e) kNN, CL=15% **(f)** SW, CL=15% **(g)** RI, CL=15% **(h)** SVMI, CL=15%

(i) kNN, CL=25% **(j)** SW, CL=25% **(k)** RI, CL=25% **(l)** SVMI, CL=25%

Fig. 3. Selection the regularization parameter of Lasso for kNN, SW, RI and SVMI imputations when $n = 50, p = 200$ and $CL = 5\%, 15\% and 25\%$.

Table 3. Outcomes obtained from the simulation configurations when CL=5% and CL=10%

CL	n	p_n	$ReMSE$ kNN	$ReMSE$ SW	$ReMSE$ RI	$ReMSE$ $SVMI$	MSE kNN	MSE SW	MSE RI	MSE $SVMI$
5%	50	200	0.022	**0.021**	0.024	0.024	**0.266**	0.315	0.315	0.318
		300	**0.050**	0.051	0.057	0.058	**0.779**	0.801	0.842	0.832
		500	**0.025**	**0.025**	0.027	**0.025**	0.985	0.964	1.044	**0.943**
	100	200	0.018	0.020	**0.016**	0.023	**0.167**	0.190	0.265	0.314
		300	0.011	0.012	**0.010**	0.011	0.275	**0.273**	0.305	0.316
		500	**0.006**	0.050	0.040	0.060	0.486	0.490	0.489	**0.484**
	150	200	0.093	0.093	**0.022**	0.037	**0.201**	0.262	0.391	0.400
		300	0.062	0.062	**0.020**	0.038	0.373	0.403	**0.348**	0.361
		500	0.032	0.032	**0.018**	0.039	0.740	0.748	**0.304**	0.323
	200	200	0.052	0.061	**0.016**	0.040	0.427	0.447	**0.261**	0.284
		300	0.055	0.065	**0.014**	0.041	0.418	0.437	**0.218**	0.246
		500	0.058	0.069	**0.012**	0.042	0.408	0.428	**0.174**	0.207
10%	50	200	0.019	0.014	**0.013**	0.016	**0.674**	0.717	0.725	0.809
		300	**0.009**	0.008	0.010	0.010	0.726	**0.537**	1.022	0.965
		500	0.003	**0.002**	**0.002**	**0.002**	0.933	**0.644**	1.332	1.291
	100	200	0.020	0.019	**0.016**	0.021	**0.429**	0.504	0.782	0.946
		300	**0.009**	**0.009**	**0.009**	0.016	**0.564**	0.578	0.715	0.881
		500	**0.055**	0.057	0.077	0.097	0.970	0.951	**0.939**	0.983
	150	200	**0.016**	0.016	0.017	0.023	**0.287**	0.385	0.788	1.068
		300	0.011	**0.009**	**0.009**	0.017	**0.414**	0.417	0.597	0.729
		500	**0.005**	**0.005**	**0.005**	0.006	**0.079**	0.082	0.087	0.091
	200	200	**0.076**	0.084	0.078	0.097	**0.161**	0.415	0.566	0.664
		300	0.011	**0.008**	**0.008**	0.011	**0.369**	0.455	0.749	0.910
		500	0.007	0.006	**0.004**	0.005	**0.605**	0.633	0.772	0.860

The best scores are indicated with bold color

4. Conclusions

In this paper, the right-censored high-dimensional model is estimated by four different linear estimators based on four imputation techniques and a weighted-ridge procedure. Also, the performance of these introduced estimators is inspected with the simulation study given in Section 3. Results are given in Tables 2-7 and Figures 1-3.

The obtained results that are provided in the tables and the figures can be interpreted individually for the imputation and the model estimation. Tables 2 and 7 present the performance of the imputation techniques that are kNN, SW, RI, and SVMI by using AvB and IA measures. Obviously, kNN and RI methods show more satisfying imputation performance than SW and SVMI methods. It is clear that kNN imputation is a more practical method and easy to compute. On the other hand, RI, SW, and SVMI are more predictable and reliable types of imputation methods than kNN because they make imputations based on least squares and they have a distributional background. Thus, the censorship problem is solved which is the first part of the study. In the second part, by using imputed response variables, the high-dimensional model is estimated by WR. The results of estimated models are given in Tables 3-6. Results show that kNN and RI-based estimators give smaller ReMSE and MSE scores and SW and SVMI. However, in most cases, four estimators provide closer performance scores.

The success of the kNN and RI methods can be explained by the linear data structure which makes it easy to impute the censored observations for the kNN and RI. On the other hand, although the SVMI is used in non-linear data mostly, in this paper, it is not the best but shows close performance to the best. In SW-based estimator is more reliable than kNN as in RI, but because it works with small partitions

Table 4. Numbers of selected covariates for strong and weak signal subsets when CL=5% and CL=10%

CL	n	$s(\hat{S}_1)$					$s(\hat{S}_2)$			
		p_n	kNN	SW	RI	$SVMI$	kNN	SW	RI	$SVMI$
5%	50	200	26	25	17	17	3	4	4	5
		300	24	22	17	14	2	3	7	10
		500	19	18	12	11	3	3	2	3
	100	200	21	21	19	19	2	2	3	3
		300	30	23	22	21	9	11	14	22
		500	26	25	24	21	2	3	4	8
	150	200	16	15	14	14	10	9	17	24
		300	17	16	14	16	18	18	12	20
		500	20	18	20	21	11	14	25	34
	200	200	14	12	13	14	5	7	7	8
		300	15	12	11	12	3	5	5	6
		500	12	15	13	13	2	2	4	6
10%	50	200	31	29	3	3	2	2	2	2
		300	21	16	11	9	6	12	9	13
		500	23	11	14	8	10	30	20	30
	100	200	20	17	20	24	17	32	38	40
		300	33	27	22	18	11	14	23	38
		500	23	23	24	16	3	6	16	13
	150	200	17	17	15	13	3	4	5	6
		300	17	17	18	20	3	3	4	5
		500	17	20	20	18	13	17	3	5
	200	200	13	14	14	15	7	8	8	8
		300	12	14	14	14	4	5	7	9
		500	18	14	14	13	23	4	7	8

of data due to its nature, sometimes it cannot catch the total dispersion of the data which diminishes its performance. From our knowledge, SW shows good performance when the dataset involves outliers.

Finally, as result, kNN and RI imputation-based WR estimators show better performance than the other two. In addition, a remarkable finding was obtained in this study. As pointed out in Section 3, as the censorship increases, four imputation methods include more weak signals in the model which is important merit brought by the WR approach. Thus, the information loss caused by the censorship is tried to be compensated by using more weak signals.

APPENDIX

A. Tables for remaining simulation configurations

Table 5. Outcomes obtained for CL=15% and CL=25%

CL	n	p_n	$ReMSE$ kNN	$ReMSE$ SW	$ReMSE$ RI	$ReMSE$ $SVMI$	MSE kNN	MSE SW	MSE RI	MSE $SVMI$
15%	50	200	0.018	0.014	0.013	**0.011**	0.622	**0.577**	0.872	0.933
		300	0.008	0.006	0.008	**0.005**	0.635	**0.566**	0.836	0.912
		500	**0.002**	**0.002**	**0.002**	**0.002**	**0.730**	0.910	0.863	0.840
	100	200	0.022	**0.020**	0.022	0.023	**0.461**	0.593	0.707	0.928
		300	0.012	**0.011**	0.016	0.017	**0.673**	0.708	0.690	0.844
		500	**0.005**	**0.005**	0.006	0.009	**0.666**	0.659	**0.648**	0.799
	150	200	0.038	**0.036**	**0.036**	0.053	**0.360**	0.584	0.625	0.894
		300	0.012	0.011	**0.010**	0.020	**0.440**	0.510	0.566	0.987
		500	0.010	0.010	**0.009**	0.014	0.797	0.892	**0.501**	0.879
	200	200	**0.011**	0.012	**0.011**	0.013	**0.183**	0.510	0.595	0.724
		300	0.012	0.012	**0.010**	0.013	**0.407**	0.496	0.513	0.716
		500	**0.006**	0.007	**0.006**	0.007	0.633	0.651	**0.466**	0.655
25%	50	200	0.166	0.167	**0.125**	0.128	**0.717**	0.865	1.174	1.380
		300	0.077	0.077	0.087	**0.068**	**0.943**	0.990	1.077	1.125
		500	0.026	0.026	**0.018**	0.024	1.055	1.042	**0.975**	1.090
	100	200	0.038	**0.034**	0.045	0.039	**0.523**	0.828	1.037	1.177
		300	0.012	**0.011**	0.015	0.017	**0.758**	0.816	0.916	1.002
		500	0.008	0.008	0.012	**0.007**	1.047	1.057	**0.851**	0.936
	150	200	**0.051**	**0.051**	0.053	0.077	**0.501**	0.768	0.962	1.067
		300	0.031	**0.027**	0.034	0.047	**0.562**	0.753	0.923	0.973
		500	0.008	**0.006**	0.008	0.008	**0.836**	0.969	0.871	0.989
	200	200	0.016	**0.015**	**0.015**	0.016	**0.162**	0.473	0.736	0.947
		300	0.023	**0.021**	0.023	0.027	**0.513**	0.803	0.687	0.750
		500	0.010	0.010	**0.009**	0.017	0.754	0.844	**0.685**	0.794

The best scores are indicated with bold color

Table 6. Numbers of selected covariates for strong and weak signal subsets when CL=15% and CL=25%

CL	n	$s(\hat{S}_1)$					$s(\hat{S}_2)$			
		p_n	kNN	SW	RI	$SVMI$	kNN	SW	RI	$SVMI$
15%	50	200	27	23	2	1	6	4	2	2
		300	13	8	5	3	2	2	5	3
		500	19	11	4	1	12	20	13	15
	100	200	21	19	20	18	2	3	4	8
		300	32	26	16	9	1	2	5	2
		500	24	25	16	7	5	11	12	23
	150	200	17	16	13	11	3	4	6	7
		300	18	17	21	19	3	4	5	8
		500	23	22	18	17	1	2	5	8
	200	200	14	16	15	17	7	8	8	8
		300	14	23	15	15	4	5	7	9
		500	17	21	15	14	16	14	28	10
25%	50	200	21	11	1	1	7	2	2	2
		300	18	9	2	1	6	5	3	3
		500	14	8	3	2	3	5	13	14
	100	200	23	22	11	11	2	4	8	9
		300	22	27	7	4	2	3	12	14
		500	22	24	9	1	6	16	17	15
	150	200	19	20	12	6	2	3	7	2
		300	20	21	14	7	3	3	8	4
		500	23	24	12	6	6	17	8	15
	200	200	15	18	16	17	7	7	8	9
		300	15	21	15	13	13	11	8	11
		500	19	25	15	11	26	15	10	14

Table 7. AvB and IA scores for the imputation methods for CL=15% and CL=25%

CL	n	AvB					IA				$Optimum$	
		p_n	kNN	SW	RI	$SVMI$	kNN	SW	RI	$SVMI$	k	w
15%	50	200	**0.398**	0.585	0.550	0.625	**0.868**	0.949	1.003	0.960	4	17
		300	**0.463**	0.699	0.550	0.634	**0.704**	0.830	0.948	0.773	8	14
		500	**0.322**	0.712	0.608	0.674	0.622	0.752	**0.538**	0.732	2	13
	100	200	**0.487**	0.778	0.569	0.652	0.664	**0.654**	0.768	0.732	10	13
		300	**0.357**	0.621	0.552	0.679	**0.566**	0.699	0.649	0.657	17	14
		500	**0.466**	0.843	0.642	0.744	0.558	**0.524**	0.555	0.544	4	13
	150	200	0.472	0.634	**0.455**	0.657	0.440	0.468	**0.408**	0.457	7	15
		300	0.443	0.784	**0.430**	0.682	**0.479**	0.524	**0.479**	0.481	8	16
		500	**0.432**	0.723	0.622	0.720	0.572	**0.372**	0.465	**0.372**	10	14
	200	200	**0.436**	0.747	**0.436**	0.604	**0.248**	0.288	0.295	0.309	12	18
		300	**0.436**	0.766	0.552	0.622	**0.225**	0.228	0.251	0.259	15	21
		500	0.461	0.774	**0.374**	0.666	**0.204**	0.260	0.205	0.243	11	17
25%	50	200	**0.432**	0.738	0.668	0.760	1.516	1.740	**1.103**	1.109	5	14
		300	**0.428**	0.739	0.532	0.610	**0.701**	0.713	0.811	0.751	5	14
		500	**0.427**	0.736	0.637	0.668	**0.383**	0.407	0.678	0.613	5	14
	100	200	0.472	0.814	**0.460**	0.635	**0.771**	0.828	0.931	0.946	10	17
		300	0.409	0.676	**0.398**	0.470	**0.700**	0.841	0.753	0.791	12	18
		500	0.442	0.793	**0.400**	0.689	0.565	**0.529**	0.698	0.588	9	13
	150	200	**0.432**	0.778	0.577	0.558	**0.728**	0.791	0.734	0.771	9	17
		300	**0.434**	0.785	0.469	0.784	0.567	0.699	0.451	**0.403**	12	17
		500	0.437	0.785	**0.412**	0.724	0.585	0.668	**0.267**	0.493	13	17
	200	200	**0.435**	0.762	0.591	0.640	**0.209**	0.332	0.247	0.250	12	17
		300	**0.419**	0.788	0.558	0.615	**0.186**	0.191	0.201	0.206	12	18
		500	**0.429**	0.757	0.590	0.679	0.362	0.189	**0.142**	0.173	10	18

The best scores are indicated with bold color

References

Abonazel, M., and Rabie, A. (2019). The impact of using robust estimations in regression models: An application on the Egyptian economy. *Journal of Advanced Research in Applied Mathematics and Statistics*,**4**(2), 8-16.

Ahmed, S. E., Aydin, D., & Yilmaz, E. (2019). Nonparametric regression estimates based on imputation techniques for right-censored data. *International Conference on Management Science and Engineering Management*, (pp. 109-120). Springer, Cham.

Ahmed, S. E., Aydin, D., & Yilmaz, E. (2020). Imputation Method Based on Sliding Window for Right-Censored Data. *International Conference on Management Science and Engineering Management*, (pp. 433-446). Springer, Cham.

Aydin, D., and Yilmaz, E. (2018). Modified spline regression based on randomly right-censored data: A comparative study. *Communications in Statistics-Simulation and Computation*, **47**(9), 2587-2611.

Aydin, D., Ahmed, S. E., & Yilmaz, E. (2021). Right-Censored Time Series Modeling by Modified Semi-Parametric A-Spline Estimator. *Entropy*, **23**(12), 1586.

Cartwright, M. H., Shepperd, M. J., and Song, Q. (2004). Dealing with missing software project data. *In Proceedings. 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry (IEEE Cat. No. 03EX717)*, 154-165.

Dang, D. M., Jackson, K. R., and Mohammadi, M. (2015). Dimension and variance reduction for Monte Carlo methods for high-dimensional models in finance. *Applied Mathematical Finance*, **22**(6), 522-552.

Dehmer, M., Emmert-Streib, F., Graber, A., & Salvador, A. (2011). *Applied statistics for network biology: methods in systems biology*. John Wiley & Sons.

Dondelinger, F., Mukherjee, S., and Alzheimer's Disease Neuroimaging Initiative (2020). The joint lasso: high-dimensional regression for group structured data. *Biostatistics*, **21**(2), 219-235.

Doreswamy, I. G., and Manjunatha, B. R. (2017) Performance evaluation of predictive models for missing data imputation in weather data. *In 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, **9**, 1327-1334.

Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B., and Tabona, O. (2021). A survey on missing data in machine learning. *Journal of Big Data*, **8**(1), 1-37.

Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, **96**(456), 1348-1360.

Gao, X., Ahmed, S. E., & Feng, Y. (2016).Post selection shrinkage estimation for high–dimensional data analysis. *Applied Stochastic Models in Business and Industry*, **33**(2), 97-120.

Gavish, M., Nadler, B., and Coifman, R. R. (2010).Multiscale wavelets on trees, graphs and high dimensional data. *Theory and applications to semi supervised learning in ICML.*

Goh, T. S., Lee, J. S., Kim, J., Park, Y. G., Pak, K., Jeong, D. C., ... and Kim, Y. H. (2019). Prognostic scoring system for osteosarcoma using network-regularized high-dimensional Cox-regression analysis and potential therapeutic targets. *Journal of cellular physiology*, **234**(8), 12851-13857.

Huang, J., Ma, S., Li, H., & Zhang, C. H. (2011). The sparse Laplacian shrinkage estimator for high-dimensional regression. *Annals of statistics*, **39**(4), 2021.

Jung, Y. (2018). Multiple predicting K-fold cross-validation for model selection. *Journal of Nonparametric Statistics*, **30**(1), 197-215.

Kim, S., Sohn, K. A., and Xing, E. P. (2009). A multivariate regression approach to association analysis of a quantitative trait network. *Bioinformatics*, **25**(12), 204-212.

Koul, H., Susarla, V., & Van Ryzin, J. (1981). Regression analysis with randomly right-censored data. *The Annals of statistics*, 1276-1288.

Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J., & Wasserman, L. (2018). Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, **113**(523), 1094–1111.

Lukas, M. A. (1993). Asymptotic optimality of generalized cross-validation for choosing the regularization parameter. *Numerische Mathematik*, **66**(1), 41–66.

Malarvizhi, R., and Thanamani, A. S. (2012). K-nearest neighbor in missing data imputation. *International Journal of Engineering Research and Development*, **5**(1), 5–7.

Orbe, J., Ferreira & Nunez-Anton, V. (2003). Censored partial regression. *Biostatistics*, **4**(1), 109–121.

Rao, H., Shi, X., Rodrigue, A. K., Feng, J., Xia, Y., Elhoseny, M., ... & Gu, L. (2019). Feature selection based on artificial bee colony and gradient boosting decision tree. *Applied Soft Computing*, **74**, 634-642.

Segal, E., Friedman, N., Koller, D., & Regev, A. (2004). A module map showing conditional activity of expression modules in cancer. *Nature genetics*, **36**(10), 1090-1098.

Stewart, T. G., Zeng, D., and Wu, M. C. (2018). Constructing support vector machines with missing data. *Wiley Interdisciplinary Reviews: Computational Statistics*, **10**(4), e1430.

Strike, K. El Emam and N. Madhavji (2001). Software cost estimation with incomplete data. *IEEE Transactions on Software Engineering*, **27**, 890-908.

Stute, W. (1993). Consistent estimation under random censorship when covariables are present. *Journal of Multivariate Analysis*, **45**(1), 89-103.

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

Zhang, C. H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, **38**(2), 894-942.

Zhang, C. H., & Zhang, S. S. (2014). Confidence intervals for low dimensional parameters in high dimensional linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **76**(1), 217-242.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, **67**(2), 301-320.

Submitted: 18/02/2022
Revised: 11/05/2022
Accepted: 12/05/2022
DOI: 10.48129/kjs.splml.18961