A. Proofs

A.1. Proof of Lemma 1

Proof. Let $(\mathcal{H}(z), \mathcal{S}(z)) = \mathcal{A}_2(\boldsymbol{y}(z))$ for all $z \in \mathbb{R}$ and write $\mathcal{H}(z) = (\mathcal{M}_1(z), \ldots, \mathcal{M}_K(z))$ and $\mathcal{S}(z) = (\mathcal{S}_1(z), \ldots, \mathcal{S}_K(z))$. By conditioning on the histories $\mathcal{H}(z) = \mathcal{H}'$,

$$\mathcal{M}_1(z) = \mathcal{M}'_1, \ldots, \mathcal{M}_K(z) = \mathcal{M}'_K$$

for all $z \in \mathbb{R}$ such that $\mathcal{H}(z) = \mathcal{H}'$. This indicates that

$$\left| \boldsymbol{x}_{j'_k}^\top \boldsymbol{r}(\boldsymbol{y}(z), X_{\mathcal{M}'_{k-1}}) \right| \geq \pm \boldsymbol{x}_j^\top \boldsymbol{r}(\boldsymbol{y}(z), X_{\mathcal{M}'_{k-1}}) \tag{18}$$

for all $(k, j) \in [K] \times ([p] \setminus \mathcal{M}'_{k-1})$ and $z \in \mathbb{R}$ such that $\mathcal{H}(z) = \mathcal{H}'$, where $(j'_1, \ldots, j'_K)$ is the sequence of the selected features when the K-step SFS algorithm is applied to the response vector $\boldsymbol{y}'$. By further conditioning on the signs $\mathcal{S}(z) = \mathcal{S}'$, (18) is written as

$$\mathcal{S}'_k \boldsymbol{x}_{j'_k}^\top \boldsymbol{r}(\boldsymbol{y}(z), X_{\mathcal{M}'_{k-1}}) \geq \pm \boldsymbol{x}_j^\top \boldsymbol{r}(\boldsymbol{y}(z), X_{\mathcal{M}'_{k-1}})$$

for all $(k, j) \in [K] \times ([p] \setminus \mathcal{M}'_{k-1})$ and $z \in \mathbb{R}$ such that $\mathcal{H}(z) = \mathcal{H}'$ and $\mathcal{S}(z) = \mathcal{S}'$. By restricting on a line $\boldsymbol{y}(z) = \boldsymbol{a} + \boldsymbol{b}z, z \in \mathbb{R}$, the range of z is written as

$$\max_{\substack{k \in [K], \\ j \in [p] \setminus \mathcal{M}'_{k-1}, \\ d_{(k,j)} > 0}} \frac{e_{(k,j)}}{d_{(k,j)}} \leq z \leq \min_{\substack{k \in [K], \\ j \in [p] \setminus \mathcal{M}'_{k-1}, \\ d_{(k,j)} < 0}} \frac{e_{(k,j)}}{d_{(k,j)}}. \tag{19}$$

□

A.2. The proof of Lemma 2

Proof. For a set of features $\mathcal{M} \subseteq [p]$ and a response vector $\boldsymbol{y}(z) = \boldsymbol{a} + \boldsymbol{b}z, z \in \mathbb{R}$ in a line, let us denote the AIC as a function of $\mathcal{M}$ and z as $\mathrm{AIC}(\mathcal{M}, z)$. Subsequently, by substituting $\boldsymbol{y} = \boldsymbol{a} + \boldsymbol{b}z$ into (14), it is written as a quadratic function of z as

$$\begin{aligned} \mathrm{AIC}(\mathcal{M}, z) = (\boldsymbol{b}^\top A_{\mathcal{M}} \boldsymbol{b}) z^2 + 2(\boldsymbol{a}^\top A_{\mathcal{M}} \boldsymbol{b}) z + (\boldsymbol{a}^\top A_{\mathcal{M}} \boldsymbol{a}) \\ + 2|\mathcal{M}|. \end{aligned} \tag{20}$$

Equation (20) represents the range of $z \in \mathbb{R}$ such that when $\boldsymbol{y}(z)$ is fed into the algorithm, the same history $\mathcal{H} = \mathcal{A}_2(\boldsymbol{y}')$ is obtained. Let the sequence of the selected models corresponding to the history $\mathcal{H}' = \psi(\boldsymbol{y}')$ be

$$\mathcal{H}' = (\mathcal{M}'_1, \ldots, \mathcal{M}'_K),$$

where K is the number of steps in the history $\mathcal{H}$. Next, the event of the history can be fully characterized by comparing AICs as follows:

$$\begin{aligned} &\mathrm{AIC}(\mathcal{M}'_k, z) \leq \mathrm{AIC}(\mathcal{M}'_{k-1} \cup \{j\}, z) \; \forall j \in [p] \setminus \mathcal{M}_{k-1}, \\ &\mathrm{AIC}(\mathcal{M}'_k, z) \leq \mathrm{AIC}(\mathcal{M}'_{k-1} \setminus \{j\}, z) \; \forall j \in \mathcal{M}_{k-1}, \\ &\mathrm{AIC}(\mathcal{M}'_k, z) \leq \mathrm{AIC}(\mathcal{M}'_{k-1}, z), \end{aligned} \tag{21}$$

for $k = 1, 2, \ldots, K$ and

$$\begin{aligned} &\mathrm{AIC}(\mathcal{M}'_K, z) < \mathrm{AIC}(\mathcal{M}'_K \cup \{j\}, z) \; \forall j \in [p] \setminus \mathcal{M}_K, \\ &\mathrm{AIC}(\mathcal{M}'_K, z) < \mathrm{AIC}(\mathcal{M}'_K \setminus \{j\}, z) \; \forall j \in \mathcal{M}_K. \end{aligned} \tag{22}$$

Here, the first and second inequalities in (21) indicate that the selected model at step k has the smallest AIC among all possible choices, the third inequality in (21) indicates that the AIC of the selected model at step k is smaller than that at the previous step, and two inequalities (22) indicate that the AIC of the selected model at the final step K cannot be decreased anymore. Because the AIC is written as a quadratic function of z as in (20) under the fixed history $\mathcal{H}'$, all these conditions are written as quadratic inequalities of z. This means that the range of $z \in \mathbb{R}$ that satisfies these conditions is represented by a finite set of quadratic inequalities of $z \in \mathbb{R}$. □

B. Details of Experiments

We executed the experiments on Intel(R) Xeon(R) CPU E5-2687W v3 @ 3.10GHz.

Comparison methods in the case of Forward SFS. In the case of forward SFS, we remind that $\mathcal{A}(\boldsymbol{y})$ results a set of selected featured $\mathcal{M}$ when applying forward SFS algorithm $\mathcal{A}$ to $\boldsymbol{y}$. With a slight abuse of notations, let $\mathcal{H}(\boldsymbol{y})$ and $\mathcal{S}(\boldsymbol{y})$ respectively denote the history and signs obtained when applying algorithm $\mathcal{A}$ to $\boldsymbol{y}$. We compared the following five methods:

- `Homotopy`: conditioning on the selected features $\mathcal{M}$ (minimal conditioning), i.e.,

$$\boldsymbol{\eta}^T\boldsymbol{Y} \mid \{\mathcal{A}(\boldsymbol{Y}) = \mathcal{A}(\boldsymbol{y}), \boldsymbol{q}(\boldsymbol{Y}) = \boldsymbol{q}(\boldsymbol{y})\}.$$

- `Homotopy-H`: additionally conditioning on the history $\mathcal{H}$, i.e.,

$$\boldsymbol{\eta}^T\boldsymbol{Y} \mid \{\mathcal{H}(\boldsymbol{Y}) = \mathcal{H}(\boldsymbol{y}), \boldsymbol{q}(\boldsymbol{Y}) = \boldsymbol{q}(\boldsymbol{y})\}.$$

  Here, we note that $\mathcal{H}(\boldsymbol{Y}) = \mathcal{H}(\boldsymbol{y})$ already includes the event $\mathcal{A}(\boldsymbol{Y}) = \mathcal{A}(\boldsymbol{y})$.

- `Homotopy-S`: additionally conditioning on the signs, i.e.,

$$\boldsymbol{\eta}^T\boldsymbol{Y} \mid \{\mathcal{A}(\boldsymbol{Y}) = \mathcal{A}(\boldsymbol{y}), \mathcal{S}(\boldsymbol{Y}) = \mathcal{S}(\boldsymbol{y}), \boldsymbol{q}(\boldsymbol{Y}) = \boldsymbol{q}(\boldsymbol{y})\}.$$

- `Polytope` (Tibshirani et al., 2016): additionally conditioning on both history $\mathcal{H}$ and signs $\mathcal{S}$, i.e.,

$$\boldsymbol{\eta}^T\boldsymbol{Y} \mid \{\mathcal{H}(\boldsymbol{Y}) = \mathcal{H}(\boldsymbol{y}), \mathcal{S}(\boldsymbol{Y}) = \mathcal{S}(\boldsymbol{y}), \boldsymbol{q}(\boldsymbol{Y}) = \boldsymbol{q}(\boldsymbol{y})\}.$$

  This definition is equivalent to

$$\boldsymbol{\eta}^T\boldsymbol{Y} \mid \{\mathcal{A}_2(\boldsymbol{Y}) = \mathcal{A}_2(\boldsymbol{y}), \boldsymbol{q}(\boldsymbol{Y}) = \boldsymbol{q}(\boldsymbol{y})\},$$

  where $\mathcal{A}_2(\cdot)$ is defined in §3.

- `DS`: data splitting is the commonly used procedure for the purpose of selection bias correction. In this approach, the data is randomly divided in two halves — first half used for model selection and the other for inference.

Comparison methods in the case of Forward-Backward SFS. In the case of forward-backward SFS, $\mathcal{A}(\boldsymbol{y})$ results a set of selected featured $\mathcal{M}$ when applying forward-backward SFS algorithm $\mathcal{A}$ to $\boldsymbol{y}$, and $\mathcal{A}_2(\boldsymbol{y})$ results the history. We compared the following two methods:

- `Homotopy`: conditioning on the selected features $\mathcal{M}$ (minimal conditioning), i.e.,

$$\boldsymbol{\eta}^T\boldsymbol{Y} \mid \{\mathcal{A}(\boldsymbol{Y}) = \mathcal{A}(\boldsymbol{y}), \boldsymbol{q}(\boldsymbol{Y}) = \boldsymbol{q}(\boldsymbol{y})\}.$$

- `Quadratic`: additionally conditioning on the history $\mathcal{H}$ (implemented by using quadratic inequality-based conditional SI in (Loftus & Taylor, 2015)), i.e.,

$$\boldsymbol{\eta}^T\boldsymbol{Y} \mid \{\mathcal{A}_2(\boldsymbol{Y}) = \mathcal{A}_2(\boldsymbol{y}), \boldsymbol{q}(\boldsymbol{Y}) = \boldsymbol{q}(\boldsymbol{y})\}.$$

Definition of TPR. In SI, we only conduct statistical testing when there is at least one hypothesis discovered by the algorithm. Therefore, the definition of TPR, which can be also called *conditional power*, is as follows:

$$\mathrm{TPR} = \frac{\#\ \text{detected \& rejected}}{\#\ \text{detected}},$$

where $\#$ detected is the number of truly positive features selected by the algorithm (e.g., SFS) and $\#$ rejected is the number of truly positive features whose null hypothesis is rejected by SI.

Demonstration of confidence interval (forward SFS). We generated $n = 100$ outcomes as $y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \varepsilon_i,\ i = 1, ..., n$, where $\boldsymbol{x}_i \sim \mathcal{N}(0, I_p)$ and $\varepsilon_i \sim \mathcal{N}(0, 1)$. We set $p = 10$, $K = 9$ and $\boldsymbol{\beta} = [0.25, 0.25, 0.25, 0.25, 0.25, 0, 0, 0, 0, 0]^\top$. We note that the number of selected features between the four options of conditional SI methods (`Homotopy`, `Homotopy-H`, `Homotopy-S`, `Polytope`) and DS can be different. Therefore, for a fair comparison, we only consider the features that are selected in both cases. Figure 5 shows the demonstration of CIs for each selected feature. The results are consistent with Fig. 1 (b).

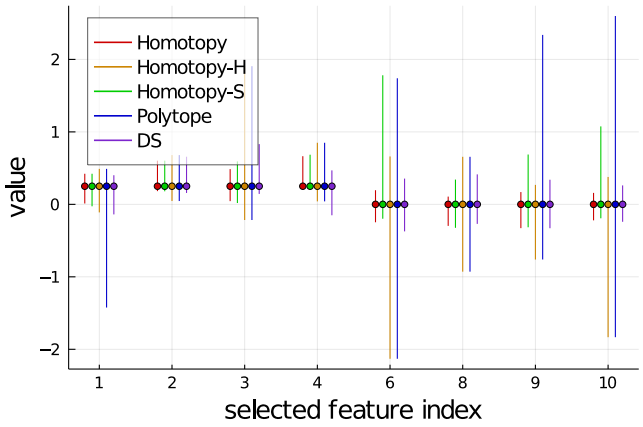

Figure 5. Demonstration of confidence interval.

C. Experiments on Computational Aspects (Forward SFS)

We demonstrate the computational efficiency of the proposed `Homotopy` method. We generated n outcomes as $y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \varepsilon_i,\ i = 1, ..., n$, where $\boldsymbol{x}_i \sim \mathcal{N}(0, I_p)$ and $\varepsilon_i \sim \mathcal{N}(0, 1)$. We set $n = 50$ and $p = 10$. In Figure 6, we show the results of comparing the computational time between the proposed `Homotopy` method and the existing method. For the existing study, if we want to keep high statistical power, we have to enumerate a huge number of all the combinations of histories and signs $2^K \times K!$, which is only feasible when the number of selected features is fairly small. We observe in blue plots in Fig. 6 that the computational cost of existing method is exponentially increasing with the number of selected features. With the proposed method, we are able to significantly reduce the computational cost while keeping high power.

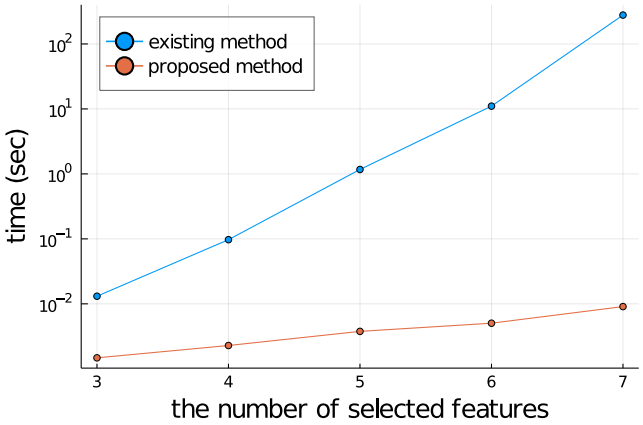

Figure 6. The result of comparing the computational time between the proposed method and the existing method with an $n = 50,\ p = 10$ artificial dataset.

One might wonder how we can circumvent the computational bottleneck of exponentially increasing number of polytopes. Our experience suggests that, by focusing on the line along the test-statistic in data space, we can skip majority of the

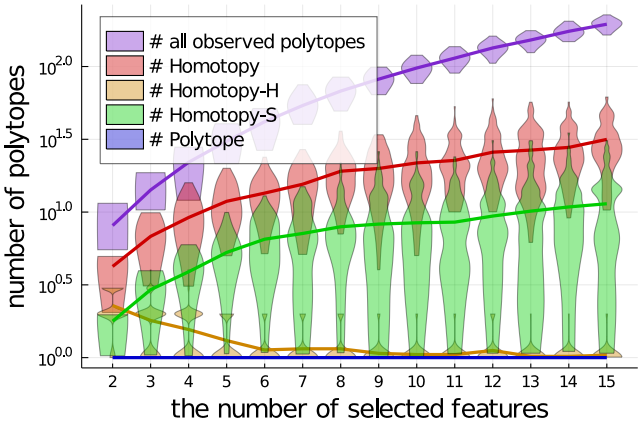

Figure 7. The number of polytopes intersecting the line z that we need to consider. The solid lines are shown the sample averages.

polytopes that do not affect the truncated Normal sampling distribution because they do not intersect with this line. In other words, we can skip majority of combinations of histories and signs that never appear. We show the violin plot of the actual numbers of intervals of the test statistic z that involves in the construction of truncated sampling distribution in Figure 7. Here, we set $n = 250$ and $= 50$. Regarding `Homotopy` and `Homotopy-S`, the number of polytopes linearly increases when increasing K. This is the reason why our method is highly efficient. In regard to `Homotopy-H`, the number of polytopes decreases because the *history* constraint is too strict when K is increased.

D. Experiments on Robustness (Forward-Backward SFS)

We applied our proposed method to the cases where the noise follows Laplace distribution, skew normal distribution (skewness coefficient 10), and t_{20} distribution. We also conducted experiments when σ^2 was also estimated from the data. We generated n outcomes as $y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \varepsilon_i$, $i = 1, ..., n$, where $p = 5, \boldsymbol{x}_i \sim \mathbb{N}(0, I_p)$, and ε_i follows Laplace distribution, skew normal distribution, or t_{20} distribution with zero mean and standard deviation was set to 1. In the case of estimated σ^2, $\varepsilon_i \sim \mathbb{N}(0, 1)$. We set all elements of $\boldsymbol{\beta}$ to 0, and set $\lambda = 0.5$. For each case, we ran 1,200 trials for each $n \in \{100, 200, 300, 400\}$. The FPR results are shown in Figure 8. Although we only demonstrate the results for the case of forward-backward SFS algorithm, the extension to forward SFS algorithm with similar setting is straightforward.

E. Homotopy-based SI for Cross-Validation (Forward SFS)

In this section, we introduce a method for SI conditional also on the selection of the number of selected features K via cross-validation. Consider selecting the number of steps K in the SFS method from a given set of candidates $\mathcal{K} = \{K_1, \ldots, K_L\}$ where L is the number of candidates. When conducting cross-validation on the observed dataset $(X, \boldsymbol{y})$, suppose that $\mathcal{V}(\boldsymbol{y}) = K_{\text{selected}} \in \mathcal{K}$ is the event that K_{selected} is selected as the best one. The test-statistic for the selected feature j when applying the SFS method with K_{selected} steps to $(X, \boldsymbol{y})$ is then defined as

$$\boldsymbol{\eta}^\top \boldsymbol{Y} \mid \{\mathcal{A}(\boldsymbol{Y}) = \mathcal{A}(\boldsymbol{y}), \mathcal{V}(\boldsymbol{Y}) = K_{\text{selected}}, \boldsymbol{q}(\boldsymbol{Y}) = \boldsymbol{q}(\boldsymbol{y})\}. \tag{23}$$

We note that $\mathcal{A}(\cdot)$ and $\boldsymbol{q}(\cdot)$ depend on K_{selected} but we omit the dependence for notational simplicity. The conditional data space in (9) with the event of selecting K_{selected} is then written as

$$\mathcal{Y} = \{\boldsymbol{y}(z) = \boldsymbol{a} + \boldsymbol{b}z \mid z \in \mathcal{Z}_{\text{CV}}\}, \tag{24}$$

where $\mathcal{Z}_{\text{CV}} = \{z \in \mathbb{R} \mid \mathcal{A}(\boldsymbol{y}(z)) = \mathcal{A}(\boldsymbol{y}), \mathcal{V}(\boldsymbol{y}(z)) = K_{\text{selected}}\}$. The truncation region $\mathcal{Z}_{\text{CV}}$ can be obtained by the intersection of the following two sets:

$$\mathcal{Z}_1 = \{z \in \mathbb{R} \mid \mathcal{A}(\boldsymbol{y}(z)) = \mathcal{A}(\boldsymbol{y})\} \quad \text{and} \quad \mathcal{Z}_2 = \{z \in \mathbb{R} \mid \mathcal{V}(\boldsymbol{y}(z)) = K_{\text{selected}}\}.$$

Since the former $\mathcal{Z}_1$ can be obtained by using the method described in §3, the remaining task is to identify the latter $\mathcal{Z}_2$.

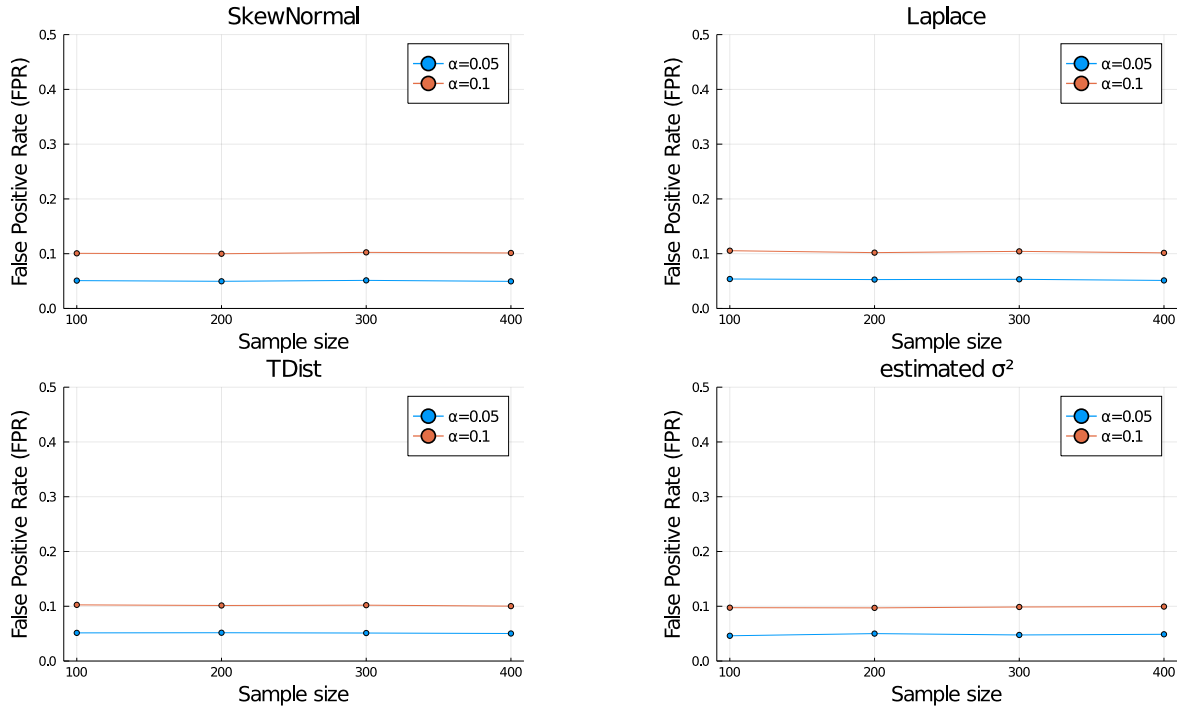

Figure 8. The robustness of the proposed method in terms of the FPR control.

For notational simplicity, we consider the case where the dataset $(X, \boldsymbol{y})$ is divided into training and validation sets, and the latter is used for selecting K_{selected}. The following discussion can be easily extended to cross-validation scenario. Let us re-write

$$(X, \boldsymbol{y}) = \left((X^{\text{tr}}\ X^{\text{va}})^\top \in \mathbb{R}^{n\times p}, (\boldsymbol{y}^{\text{tr}}\ \boldsymbol{y}^{\text{va}})^\top \in \mathbb{R}^n\right).$$

With a slight abuse of notation, for $K \in \mathcal{K}$, let $\mathcal{M}_K(\boldsymbol{y}^{\text{tr}}(z))$ be the set of selected features by applying K-step SFS method to $(X^{\text{tr}}, \boldsymbol{y}^{\text{tr}}(z))$. The validation error is then defined as

$$E_K(z) = \|\boldsymbol{y}^{\text{va}}(z) - X^{\text{va}}_{\mathcal{M}_K(\boldsymbol{y}^{\text{tr}}(z))}\hat{\boldsymbol{\beta}}_K(z)\|_2^2, \tag{25}$$

where $\hat{\boldsymbol{\beta}}_K(z) = \left(X^{\text{tr}}_{\mathcal{M}_K(\boldsymbol{y}^{\text{tr}}(z))}{}^\top X^{\text{tr}}_{\mathcal{M}_K(\boldsymbol{y}^{\text{tr}}(z))}\right)^{-1} X^{\text{tr}}_{\mathcal{M}_K(\boldsymbol{y}^{\text{tr}}(z))}{}^\top \boldsymbol{y}^{\text{tr}}(z)$. Then, we can write

$$\mathcal{Z}_2 = \{z \in \mathbb{R} \mid E_{K_{\text{selected}}}(z) \le E_K(z) \text{ for any } K \in \mathcal{K}\}.$$

Since the validation error $E_K(z)$ in (25) is a picecewise-quadratic function of z, we have a corresponding picecewise-quadratic function of z for each $K \in \mathcal{K}$. The truncation region $\mathcal{Z}_2$ can be identified by the intersection of the intervals of z in which the validation error $E_{K_{\text{selected}}}(z)$ corresponding to K_{selected} is minimum among a set of picecewise-quadratic functions for all the other $K \in \mathcal{K}$.

Loftus (2015) already discussed that it is possible to consider cross-validation event into conditional SI framework. However, his method is highly over-conditioned in the sense that additional conditioning on all intermediate models in the process of cross-validation is required. Our method described above is minimumly-conditioned SI in the sense that our inference is conducted based exactly on the conditional sampling distribution of the test-statistic in (23) without any extra conditions.

For the experiments on cross-validation, we demonstrate the TPRs and the CIs between the cases when $K = 9$ is fixed and K is selected from the set $\mathcal{K}_1 = \{3, 6, 9\}$, or $\mathcal{K}_2 = \{1, 2, \ldots, 10\}$ using 5-fold cross-validation. We set $p = 10$, only the first elements of $\boldsymbol{\beta}$ was set to 0.25, and all the rest were set to 0. We show that the TPR tends to decrease when increasing the size of $\mathcal{K}$ in Figure 9. This is due to the fact that when we increase the size of $\mathcal{K}$, we have to condition on more information which leads to shorter truncation interval and results low TPR. The TPR results are consistent with the CI results shown in Figure 10. In other words, when increasing the size of $\mathcal{K}$, the lower the TPR is, the longer the length of CI becomes.

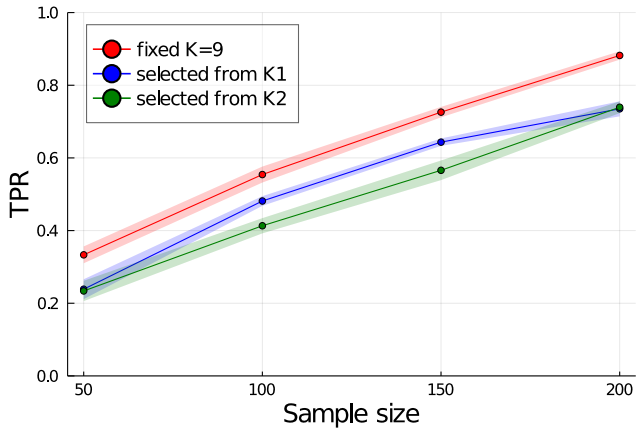

Figure 9. Demonstration of TPR when accounting cross-validation selection event.

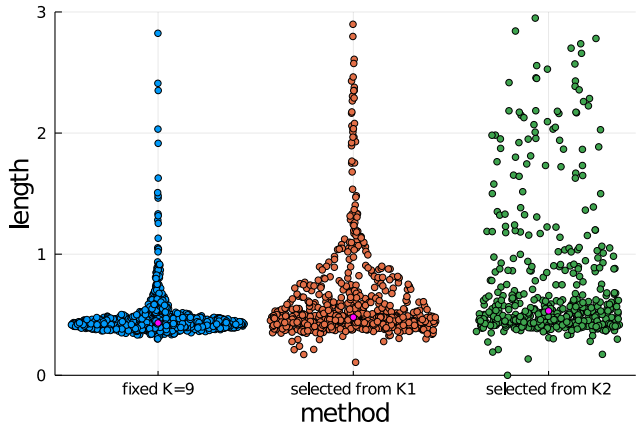

Figure 10. Demonstration of CI length when considering cross-validation selection event.