Exploiting disagreement between high-dimensional variable selectors for uncertainty visualization

Christine Yuen
Department of Statistics, London School of Economics and Political Science
and
Piotr Fryzlewicz
Department of Statistics, London School of Economics and Political Science

March 5, 2020

Abstract

We propose Combined Selection and Uncertainty Visualizer (CSUV), which estimates the set of true covariates in high-dimensional linear regression and visualizes selection uncertainties by exploiting the (dis)agreement among different base selectors. Our proposed method selects covariates that get selected the most frequently by the different variable selection methods on subsampled data. The method is generic and can be used with different existing variable selection methods. We demonstrate its variable selection performance using real and simulated data. The variable selection method and its uncertainty illustration tool are publicly available as R package `CSUV` (`https://github.com/christineyuen/CSUV`). The graphical tool is also available online via `https://csuv.shinyapps.io/csuv`.

Keywords: high-dimensional data, variable selection, uncertainty visualization

1 Introduction

Model and variable selection in high-dimensional regression settings have been widely discussed in the past decades. In the context of the linear model, the best subset selection (dated back to at least Beale et al., 1967) is computationally infeasible when the number of covariates p is large. Regularization methods with convex penalties, such as the Lasso (Tibshirani, 1996), are capable of performing variable selection in large-p settings and yet they are computationally efficient. Elastic Net (Zou and Hastie, 2005) is believed to be particularly suitable for designs with a high degree of correlation between the covariates. Group Lasso (Yuan and Lin, 2006) is designed for situations in which the covariates are best considered in groups. Regularized regression methods with non-convex penalties such as the smoothly clipped absolute deviation (SCAD, Fan and Li, 2001) and minimax concave penalty (MCP, Zhang et al., 2010) methods are designed to reduce estimation bias. The theoretical evaluation of the properties of these and many others variable selection methods has been the subject of intense research effort. For example, the irrepresentable condition (Zhao and Yu, 2006) is sufficient and almost necessary for the Lasso to be sign consistent. Fan and Lv (2010) provide a detailed review of different variable selection methods in high-dimensional settings.

There has also been a growing focus on post-selection inference. Van de Geer et al. (2014), Zhang and Zhang (2014) and Javanmard et al. (2018) advocate the de-biasing approach, which constructs confidence intervals for covariates by de-sparsifying the Lasso estimators. Lee et al. (2016), Tibshirani et al. (2016) and Tibshirani et al. (2018) propose a conditional approach which provides confidence intervals for the selected covariates using the distribution of a post-selection estimator conditioning on the selection event. Chatterjee and Lahiri (2011) and Liu et al. (2013) suggest using bootstrapping on some existing variable selection methods.

In this paper we focus on identifying the true set of covariates and illustrating the selection uncertainty in the linear model. We assume that the observed data are the realization of:

$$Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j X_i^j + \epsilon_i, \quad i = 1, ..., n, \tag{1}$$

where p is the number of covariates, n is the number of observations, and we potentially have $p > n$. X_i^j is the j^{th} covariate of the i^{th} observation of $\boldsymbol{X}$ and $\boldsymbol{X}$ is a fixed $n \times p$ design matrix. $\boldsymbol{X}$ is standardized with each covariate X^j has $\sum_{i=1}^n X_i^j/n = 0$ and $\sum_{i=1}^n (X_i^j)^2/n = 1$. ϵ is i.i.d. noise with mean zero and variance σ^2. Furthermore, the model is assumed to be sparse with the set of true covariates $S = \{j \in \{1, ..., p\} : \beta_j \neq 0\}$, $s = |S| \ll p$.

Less effort has been devoted in the literature to *selecting the best variable selection method* for the data at hand. Various theoretical performance guarantees are available for a range of methods, but many of them are not testable in practice; for instance, checking the irrepresentable condition usually requires knowing the true set of covariates. Therefore, this type of theory can be of limited use in method selection. How to select a method remains an open and yet very important question to ask, as it affects our selection of the set of relevant variables. To illustrate this impact, let us consider two real-life datasets in Examples 1 and 2.

Example 1 (Riboflavin data)**.** The riboflavin dataset concerns the riboflavin (vitamin B2) production by bacillus subtilis. The response is the logarithm of the riboflavin production rate by bacillus subtilis and the $p = 4088$ covariates are the logarithms of the expression levels of 4088 genes. The number of samples is $n = 71 \ll p$. The dataset is available in the R package `hdi`.

Example 2 (Prostate cancer data, Stamey et al., 1989)**.** The prostate cancer dataset comes from a study that examined the relationship between the level of prostate-specific antigen and $p = 8$ clinical measures (logarithm of weight, age, Gleason score, among others) in men who were about to receive a radical prostatectomy. The sample size is $n = 97$. The dataset is available in the R package `lasso2`.

We process the datasets using five different variable selection methods: the Lasso, Elastic Net, relaxed Lasso (Meinshausen, 2007), MCP and SCAD in R with default tuning in the corresponding R packages (see Section 5.1.1 for more details). We justify the choice of these particular methods in Section 3.3.5. Working with default parameters would be a commonly used starting point for the non-expert applied user. The selection results are shown in Figures 1 and 2.

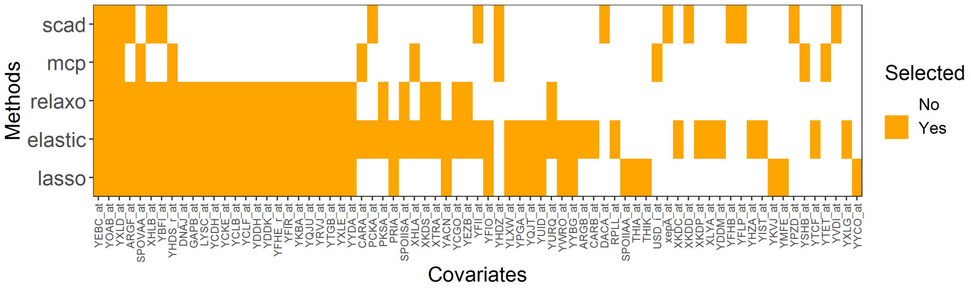

Figure 1: Graphical illustration of selections by different variable selection methods (Lasso, Elastic Net, relaxed Lasso, MCP and SCAD) with default tuning using the riboflavin dataset from Example 1. Covariates that are not selected by any methods are not shown in the graph for readability.

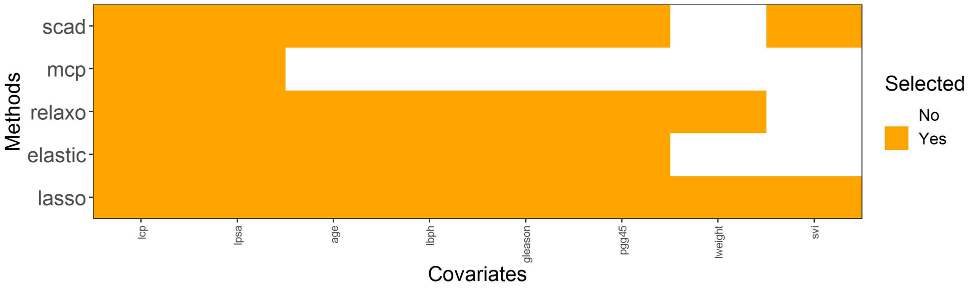

Figure 2: Graphical illustration of selections by different variable selection methods (Lasso, Elastic Net, relaxed Lasso, MCP and SCAD) with default tuning using the prostate dataset from Example 2.

Figure 1 shows that for the riboflavin dataset the sets of covariates selected vary significantly among the methods, which makes it difficult to justify the validity of the set of covariates selected using any one method. For the prostate cancer dataset, even though there are only eight covariates to choose from, there is still selection disagreement among the methods (Figure 2).

Such disagreement among methods as shown in Figures 1 and 2 is not an exception but a common observation. The distance heat maps in Appendix A.3 show that selection disagreement manifests itself across different simulation settings (see Section 5 for more details on the simulation settings). Having observed disagreement, one possible way to proceed would be to rank the different models considered (e.g. using cross-validation or an information criterion) and select the highest-ranked one. In this paper, we consider eBIC (Chen and Chen, 2008) and delete-$n/2$ cross-validation (Zhang and Yang, 2015) as they are suitable for high-dimensional settings. Further details of these two methods are discussed in Section 2.1. Our simulation results show that in general eBIC performs better than the delete-$n/2$ cross-validation in terms of variable selection (see Tables 1-15 in Appendix A.4). In fact, eBIC in many simulation settings performs very similarly to the best performing individual variable selection method.

Although eBIC seems to be able to select a single good model fit, can more be said regarding the uncertainty of variable selection, based on the disagreement between the methods tested? The similarities and disagreements among the different variable selectors, which is a piece of information not typically used by any one of them, may provide us with some useful insight. For example, in Figure 1 all of the methods select the first three covariates whereas the remaining covariates are selected by some of the methods only. Does it mean that the first three covariates are more likely to be the true covariates? This question is central to this paper, and motivates our main development, described next. In this paper, we propose a new tool for variable selection with uncertainty visualization, termed Combined Selection and Uncertainty Visualizer (CSUV). CSUV combines, in a particular way, a number of different base variable selection methods into a new variable selector, and illustrates the output of this new selector together with a graphical representation of its uncertainty. It makes use of sets of covariates selected on different subsamples of the data with different variable selection methods. A full description of the proposed method is in Section 3 and 4. The variable selection part of the proposed procedure can be summarized as follows: first, split the data into the training and test sets and fit different variable selection methods on the training set over a grid of tuning parameter values. Estimate the performance of the fitted models on the test set, and retain only the k best-performing models. Repeat the process a number of times and select the covariates that appear the most frequently in the collection of the retained fitted models.

The other component of CSUV is a graphical tool designed to visualize the selection uncertainty by using disagreement among the different model fits. See Figure 3 as an example of a graphical output of CSUV. The plot shows the frequency with which each covariate is selected and the variability of the non-zero estimated coefficients. As we will see in Section 4.2, the graphical tool can be used to assist variable selection.

Our numerical experience (see Figure 4 for a summary) suggests that the fitted models selected by CSUV tend to be distributed fairly uniformly over the entire range of the base variable selection methods used. This shows CSUV generally makes use of most of the base variable selection methods to get the final fitted model.

The paper is organized as follows. In Section 2, we describe some related work. In Section 3, we discuss the main ideas behind CSUV, and we present the variable selection and coefficient estimation part of CSUV. In Section 4, we introduce the graphical tool of CSUV to illustrate the disagreement in variable selection and the variability in coefficient estimation, and demonstrate its capability in assisting variable selection. In Section 5, we present the simulation results. We conclude the paper with a discussion in Section 6.

2 Related work

2.1 Model selection procedures

One possibility open to analysts when faced with competing fitted models is to select one of them. For example, Chen and Chen (2008) propose eBIC, an extension of BIC to high-dimensional data which takes into account both the number of unknown parameters and the complexity of the model space. Zhang and Yang (2015) advocate the use of the delete-$n/2$ cross-validation to select a method among all the candidate methods. For each iteration, delete-$n/2$ cross-validation uses half of the data for fitting and half for evaluation. The authors argue that in order to consistently identify the best variable selection procedure by cross-validation, the evaluation part has to be sufficiently large so that there are (1) more observations in the testing part to provide better evaluation and (2) fewer observations in

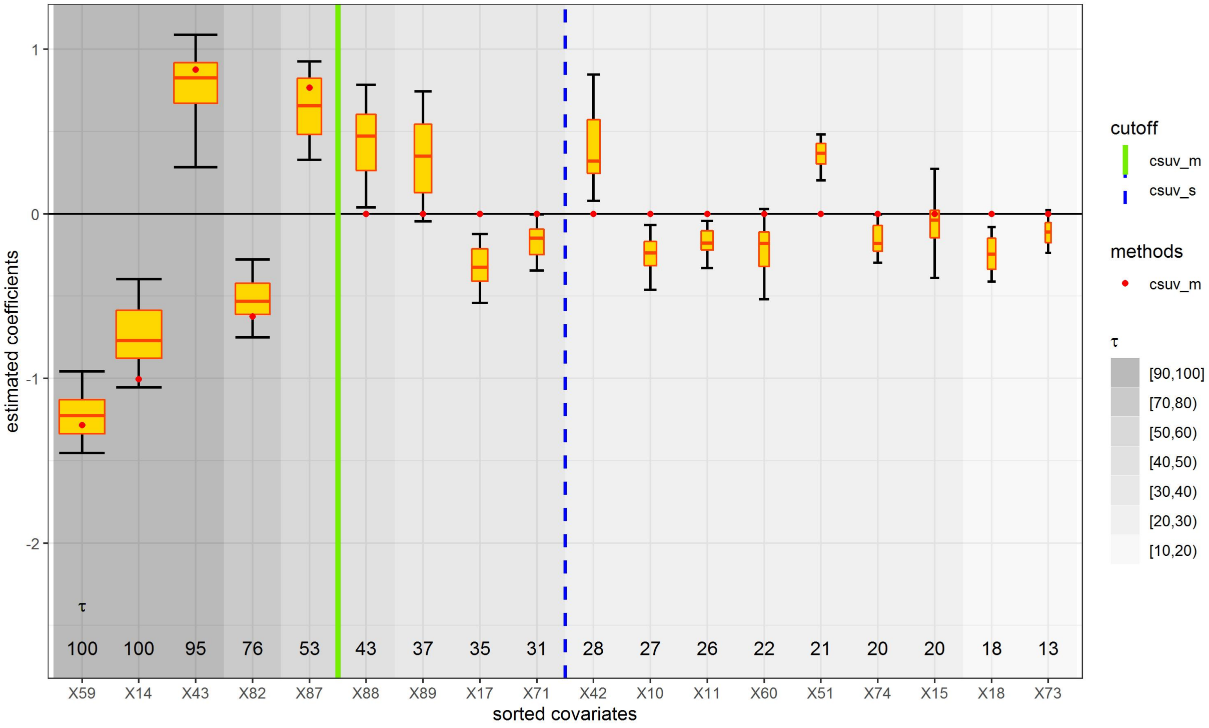

Figure 3: Example of the CSUV graphical tool with simulated data from model 2 parameter setting 5 (see Section 5.1.4 for more details on the simulation setting). Box plots illustrate the empirical distributions of the estimated coefficients conditional on them being non-zero, and the whiskers represent their 5% and 95% percentiles. The ordering of the covariates is according to the CSUV solution path (see Definition 3 in Section 3.4) and the width of each box plot along the x-axis is proportional to the level of the relative same sign frequency τ_j (see Definition 1 in Section 3.2; heuristically, the higher the value of τ_j, the higher the frequency with which the corresponding variable has been selected with the same positive or negative sign). The numbers at the bottom of the graph show the actual values of τ_j times 100 and the shade in the background corresponds to the level of τ_j with ranges as shown in the legend. Dots (red in the color version) are the estimated coefficients by CSUV-m (see Definition 2 in Section 3.3.4). The solid vertical line (green in the color version) represents the cut-off of CSUV-m, and the dotted vertical line (blue in the color version) represents the cut-off of CSUV-s (see Definition 4 in Section 3.4). Covariates with $\tau_j < 0.1$ are not shown for readability.

the training part to magnify the difference in performance between methods.

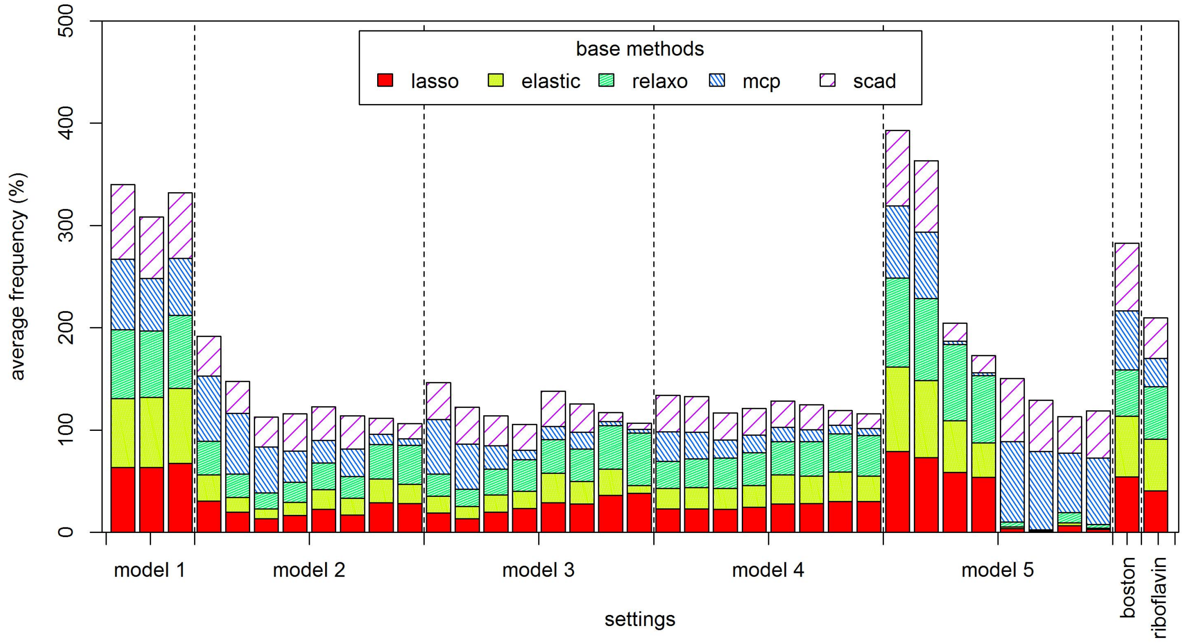

Figure 4: Average relative frequency of the constituent methods selecting the same set of covariates as the fitted models retained by CSUV when the Lasso, Elastic Net, relaxed Lasso, MCP and SCAD are used as the constituent variable selection methods for CSUV in our simulations (see Section 5.1.4 for more details on the simulation settings). The sum of the average frequency of methods can be more than 100% as multiple methods can select the same set of covariates.

2.2 Model combination with a single method

Model combination with subsampling has been used to improve variable selection performance of a single variable selection method. For example, Bolasso (Bach, 2008) fits the Lasso on each bootstrap sample and takes the intersection of all the selections. Wang et al. (2014) propose the median selection subset aggregation estimation (MESSAGE) algorithm which aims to perform variable selection on large-n datasets. It runs a variable selection method (e.g. the Lasso) in parallel on each subset of the data and selects the set of covariates whose median is non-zero. The ranking-based variable selection (Baranowski et al., 2020) algorithm uses subsampling to identify the set of consistently highly-ranked covariates. Stability selection (Meinshausen and Bühlmann, 2010 and Shah and Samworth,

2013) provides control over the finite sample familywise type I errors via subsampling. The stability selection procedure repeatedly samples observations and fits the sampling data using a variable selection method (e.g. the Lasso). It then keeps the covariates with selection frequency higher than a certain threshold.

Similarly to the methods above, CSUV, our proposal, fits variable selection methods on subsampled data and selects the covariates that appear the most frequently. Unlike these other approaches, however, CSUV makes use of different variable selection methods as we observe that no one method outperforms all other methods in all settings. This brings various advantages, including obtaining access to good model fits from different variable selection methods, and being able to exploit disagreement between the selectors to evaluate selection uncertainty. We elaborate on these points later.

2.3 Model combination with multiple methods

Adaptive regression by mixing (ARM, Yang, 2001) and its variation, adaptive regression by mixing with screening (ARMS, Yuan and Yang, 2005), aggregate fits from different methods by estimating weights through subsampling. ARM uses half of the data to fit some candidate models/procedures (e.g. smoothing splines with cross-validation tuning) and estimate σ. The remaining data is used to evaluate the prediction loss. The weight for each candidate model/procedure is calculated using $\hat{\sigma}$ and prediction loss. ARM gets the final weights by averaging the weights from different iterations. Finally it fits the full set of data using all the candidate models/procedures and obtains the final model by averaging the fits using the estimated weights. ARMS is similar to ARM except it uses half of the data to calculate AIC or BIC and retains only the models that have low AIC or BIC. The final fitted model from ARM or ARMS is not necessarily sparse as it is a weighted average of a number of models. Variable selection deviation measures (VSD, Nan and Yang, 2014) aim to provide a sense of how trustworthy a set of selected covariates is. The VSD of a target model m is the weighted cardinality of the symmetric difference between m and each candidate model. Nan and Yang (2014) suggests using the sets of fitted models on the solution paths from the Lasso, SCAD and MCP as candidate models and the weight of each candidate model is calculated based on information criteria or ARM. The simulation results in Nan and Yang (2014) show that a large VSD compared to the size of the target model means that the target model is not trustworthy, but a small VSD does not necessarily mean that the target model is close to the true model. Yang and Yang (2017) propose to select a set of covariates that minimizes the total Hamming distance with all the candidate models in terms of VSD (we refer to this method as VSD-minimizing in the remainder of the paper). The authors also propose using different thresholds, where the threshold of 0.5 is equivalent to minimizing the standard Hamming distance. For variable selection method combinations that do not involve subsampling, Tsai and Hsiao (2010), Mares et al. (2016) and Pohjalainen et al. (2015) provide empirical results on combining sets of selected covariates from different variable selection methods by intersection, union and/or some other set operations.

Both our method and VSD use resampling and different variable selection methods to provide an assessment of how good the final set of covariate selection is. VSD focuses on the whole model fit. Our method focuses on the uncertainty of individual covariates and a graphical tool is designed to illustrate these uncertainties. In terms of methodology detail, our method combines the sets of covariates selected in resampling fits whereas VSD combines sets of covariates selected on the solution path when fitting using all the data. Resampling data is only used in VSD for calculating the weight of each set of covariates. In our simulation study we compare the variable selection performance of our method to the VSD-minimizing method proposed by Yang and Yang (2017), as it is the method the most similar to CSUV. The simulation results in Section 5.2.2 show that in general our method outperforms the VSD-minimizing model.

3 CSUV variable selection methodology

3.1 Simple aggregation

The first goal of this paper is to use the similarity of fits from different methods to obtain the final set of covariates. One naive way to do so would be as follows.

- Step 1: fit the data using different variable selection methods.

- Step 2: record the percentage of times a covariate X_j is selected among the different methods. Denote it by θ_j.

- Step 3: get the final set of covariates by selecting covariates with high θ_j's. For example, select the set of covariates $\{X_j : \theta_j \geq 0.5\}$.

Different variable selection methods optimize different objective functions. In the case of regularized regression, the difference among methods is usually in terms of the penalty. If a covariate is selected by the majority of methods, it means the covariate is chosen to minimize many different objective functions. We expect that a true covariate j should have a high θ_j, i.e. it should frequently be chosen regardless of the objective function used. This simple procedure, however, suffers from the following drawbacks.

- Some variable selection methods can be similar in terms of selection regardless of the data as their objective functions are similar. Taking an extreme example, if we include two equivalent variable selection methods, such as the constrained and the penalized forms of the Lasso with equivalent regularization parameters, the sets selected by both methods will be the same. Such set is selected twice not because it maximizes two different object functions but merely because two equivalent methods are considered. This issue can cause an uneven "sampling" of methods and the corresponding fitted models.

- The above procedure assigns the same weight to all the base methods. When the performance across methods is very different (for example one method is substantially better than the others), such equal weight assignment is not ideal.

- Several methods can be wrong at the same time. For example, a false covariate can be wrongly selected by most methods if it has a spuriously high sample correlation with the response. When all methods are not performing well, a false covariate may have a high θ_j.

In the next section, we discuss how to overcome these drawbacks.

3.2 CSUV variable selection

Motivated by the above discussion, the variable selection in CSUV uses the general simple aggregation principles introduces in the previous section, but is also supplemented with the additional principles below:

- Only include the fitted models that exhibit good performance, in the sense specified in Section 3.3.2.
- Repeat the fitting on subsampled data, to incorporate the variability in selection caused by the variability in data.

The variable selection procedure of CSUV can be summarized as follows. First, randomly split the data into training and test sets, and fit different variable selection methods on the training set over a grid of regularization parameters without tuning (see Section 3.3.5 and 5.1.1 for more the details on the grid of regularization parameters considered). Then, use the test set to calculate the performance of the fitted models and retain only the first k fitted models that have the best performance (see Section 3.3.2 for more details on performance measure). Repeat the process many times to record a list of retained fitted models. Finally, select the covariates that appear the most frequently with the same positive or negative sign in the retained fitted models. The pseudo-code in Algorithm 1 provides a more detailed description of the variable selection part of CSUV. Coefficient estimation on the selected set is discussed in Section 3.3.1.

Before we present Algorithm 1, we define the relative same sign frequency τ_j, which measures the percentage of times that the jth covariate is selected with the same sign.

Definition 1 (Relative same sign frequency τ_j)**.** *Assume we have a set of fitted models $\mathcal{M}$. The relative same sign frequency of covariate X_j is defined as:*

$$\tau_j = \frac{1}{|\mathcal{M}|} \max \left(\sum_{M_k \in \mathcal{M}} \mathbb{1}_{\hat{\beta}_j^{M_k} > 0}, \sum_{M_k \in \mathcal{M}} \mathbb{1}_{\hat{\beta}_j^{M_k} < 0} \right)$$

where $\hat{\beta}_j^{M_k}$ is the estimated coefficient of the j^{th} covariate on the fitted model $M_k \in \mathcal{M}$, and $\mathbb{1}_x$ is the indicator function.

Algorithm 1 Select a set of covariates in CSUV

```
Input: variable selection methods $\mathcal{A}_1, ..., \mathcal{A}_R$ with the corresponding generation of the grid of regularization parameters; $n$ observations with $p$ covariates $\boldsymbol{X}$ and response $Y$; number of repetitions $B$, percentile parameter $q$; frequency threshold $t$; percentage of data used in training set $w\%$; performance measure.
Output: set of selected covariates $\hat{S}$.
 1: for $b$ in $\{1, ..., B\}$ do
 2:     randomly assign $w\%$ of the observations as training data with labels $I^b_{train}$ and the rest as test data with label $I^b_{test}$. Fit data with label $I^b_{train}$ using $\mathcal{A}_1, ..., \mathcal{A}_R$ over grids of $K'_r$ different values of the corresponding regularization parameters, $r \in \{1, ..., R\}$. For each method $\mathcal{A}_r$, denote the fitted models as $\tilde{M}^b_{r,1}, ..., \tilde{M}^b_{r,K'_r}$ and the set of covariates selected by each fitted model as $S^{\tilde{M}^b_{r,k}} = \{j : \tilde{\beta}_j^{\tilde{M}^b_{r,k}} \neq 0\}, k \in \{1, ..., K'_r\}$.
 3:     remove any duplication *within each method* in terms of variable selection to get $S^{\tilde{M}^b_{r,1}}, ..., S^{\tilde{M}^b_{r,K_r}}$ such that for each $r, S^{\tilde{M}^b_{r,k}} \neq S^{\tilde{M}^b_{r,k'}}$ $\forall k \neq k' \in \{1, ..., K_r\}$. Record the sets of covariates selected by each fitted model $S^{\tilde{M}^b_{1,1}}, ..., S^{\tilde{M}^b_{1,K_1}}, ..., S^{\tilde{M}^b_{R,1}}, ..., S^{\tilde{M}^b_{R,K_R}}$ and re-index as $S^{\tilde{M}^b_1}, ..., S^{\tilde{M}^b_{K^b}}$, where $K^b$ is the number of fitted models recorded.
 4:     if the number of selected covariates $|S^{\tilde{M}^b_k}| < |I^b_{train}|$, refit the selected set of covariates $S^{\tilde{M}^b_k}$ using ordinary least squares (OLS), to get the fitted models $\hat{M}^b_1, ..., \hat{M}^b_{K^b}$ with the estimated coefficients $\hat{\beta}_j^{\hat{M}^b_k}$. Otherwise, set $\hat{\beta}_j^{\hat{M}^b_k} = \tilde{\beta}_j^{\tilde{M}^b_k}$.
 5:     use data with label $I^b_{test}$ to estimate the performance of each fitted model $\hat{M}^b_k$ from Step (4). Order the models $\hat{M}^b_1, ..., \hat{M}^b_{K^b}$ by the performance measure calculated from the best to the worst to obtain $\hat{M}^b_{(1)}, ..., \hat{M}^b_{(K^b)}$.
 6:     retain the first $q\%$ of the fitted models $\hat{M}^b_{(1)}, ..., \hat{M}^b_{(K^b_q)}$, where $K^b_q = \text{round}(K^b \times q/100)$.
 7: end for
 8: denote the set of retained fitted models by $\mathcal{M} = \{\hat{M}^1_{(1)}, ..., \hat{M}^1_{(K^1_q)}, ..., \hat{M}^B_{(1)}, ..., \hat{M}^B_{(K^B_q)}\}$.
 9: calculate the relative same sign frequency $\tau_j$ for each variable $j$ according to Definition 1 and select the covariates such that:
        $$\hat{S} = \{j : \tau_j \geq t\}$$
10: return $\hat{S}$.
```

In Step (3) of Algorithm 1, duplicated sets of covariates selected within each method from Step (2) are removed. This is because while multiple selections of a set of covariates in Step (2) may suggest that the covariates in the set are likely to be the true covariates, it can also just be because many similar regularization parameter values have been used in Step (2). In order to reduce the dependency of the frequency of the appearance of a set of covariates on the choice of the grid of regularization parameters, duplicated sets of covariates selected within each method from Step (2) are removed. Note that duplicated sets of covariates selected across methods are not removed.

Algorithm 1 involves repeated fits on subsamples of data, and this can be computationally expensive. Fortunately, the algorithm can easily be parallelized by running iterations on different cores/machines. This makes the algorithm feasible for high-dimensional data analysis. For example, on a 3-core machine, Applying CSUV on the riboflavin dataset of Example 1 takes less than 2 minutes when following the specifications recommended in Section 3.3 ($B = 100$, the constituent methods = {Lasso, MCP, SCAD}, etc.). Applying CSUV with the recommended specification on each simulated dataset in Section 5.1.4 with $p = 300$ takes less than 30 seconds.

3.3 Specifications for CSUV variable selection

Algorithm 1 provides a general framework for the CSUV variable selection approach. Here we discuss how the various parameters should or may be set for practical use.

3.3.1 Coefficient estimation

Algorithm 1 only selects a set of covariates without estimating the β coefficients. In our implementation we use ordinary least squares (OLS) to estimate the β coefficients on the selected set $\hat{S}$ using the full set of data to form the final fitted model. If the number of covariates selected is larger than the number of observations, we use ridge regression to estimate the coefficients by cross-validation (in this case, we use the default cross-validation setting from the `glmnet` R package).

3.3.2 Performance measure

Step (5) of Algorithm 1 aims to rank the fitted models based on their variable selection performance. As we do not know the true covariates, we are not able to measure variable selection performance directly. In general, in attempting to select fitted models or methods with good variable selection performance, it is common to use prediction measures such as MSE or information criteria such as BIC or eBIC. Theoretically, BIC is consistent in model identification when p is fixed and eBIC is consistent in high-dimensional settings (Chen and Chen, 2008). Our empirical experiments, however, show that when using BIC or eBIC as performance measures in Algorithm 1, the resulting fitted models tend to select too few covariates so the final selection by CSUV omits too many true covariates. By contrast, using MSE as the performance measure in Algorithm 1 in our simulation settings provides good variable selection performance. Although MSE measures prediction rather than variable selection performance, MSE is often used for variable selection methods such as in selecting tuning parameter λ for SCAD (Fan and Li (2001)).

3.3.3 Percentage of data used in training set $w\%$

Following Yang (2001), Yuan and Yang (2005) and Zhang and Yang (2015), we use 50% of the data for fitting and the remaining 50% for testing; this splitting ratio attempts to ensure a sufficiently large sample size for both. Stability selection (Meinshausen and Bühlmann, 2010 and Shah and Samworth, 2013) also uses the same splitting ratio although their rationale is that subsampling with such a ratio behaves similarly to bootstrapping.

Empirically, our simulations show that using a smaller training set (25% of the data) results in the selecting fitted models with too few covariates. When using a large training set (75% of the data), the selected fitted models are too similar to each other, which causes CSUV to select too many false covariates.

3.3.4 Frequency threshold t

The frequency threshold t features in Step (9) of Algorithm 1. In this paper, we set $t = 1/2$, which means that covariates with $\tau_j \geq 1/2$ are selected. We have the following definition, in which the "m" stands for median, because selecting covariates with $\tau_j \geq 1/2$ is equivalent to selecting covariates with a non-zero median in $\mathcal{M}$.

Definition 2 (CSUV-m). *The CSUV method described by Algorithm 1 and using $t = 1/2$ is denoted by CSUV-m.*

The following results hold.

Proposition 1. *If the signs for the non-zero $\hat{\beta}_j^k$'s for all k for which $M_k \in \mathcal{M}$ are the same, i.e.*

$$\tau_j = \frac{\sum_{\{k|M_k \in \mathcal{M}\}} \mathbb{1}_{\beta_j^k \neq 0}}{|\mathcal{M}|},$$

then selecting a covariate j when $\tau_j \geq 1/2$ is equivalent to minimizing the average Hamming distance between the final selected sets of covariates $\hat{S}$ and all the fitted models $M_k \in \mathcal{M}$.

Proposition 2. *Let $\tau_j^+ = \frac{1}{|\mathcal{M}|}\sum_{\{k|M_k\in\mathcal{M}\}} \mathbb{1}_{\hat{\beta}_j^k>0}$ and $\tau_j^- = \frac{1}{|\mathcal{M}|}\sum_{\{k|M_k\in\mathcal{M}\}} \mathbb{1}_{\hat{\beta}_j^k<0}$ (note $\tau_j = max(\tau_j^+, \tau_j^-)$). Consider the following distance function between a model M and all the fitted models $M_k \in \mathcal{M}$:*

$$dist(M, \mathcal{M}) = \sum_{j=1}^{p} \sum_{\{k|M_k\in\mathcal{M}\}} |s_j^M - sign(\hat{\beta}_j^k)|$$

where s_j^M is the sign of the coefficient of the covariate j in model M which can take the value -1, 0 or 1. Selecting a covariate j when $\tau_j \geq 1/2$ and setting $s_j^M = +1$ when $\tau_j^+ \geq 1/2$ and -1 when $\tau_j^- \geq 1/2$ minimizes $dist(M, \mathcal{M})$.

The proofs are in Appendix A.1 and A.2. Selecting covariates via thresholding τ_j in CSUV-m (Definition 2) is not the only option. In Section 3.4 we introduce CSUV-s, which uses information provided by the sizes of the retained models.

3.3.5 Constituent variable selection methods $\mathcal{A}_1, ..., \mathcal{A}_R$

CSUV is designed to be generic so that any variable selection methods can be used as the constituent methods $\mathcal{A}_1, ..., \mathcal{A}_R$ in CSUV. Ideally, all the methods $\mathcal{A}_r$ should have good variable selection performance, and there should be some variability among the methods in terms of false selection. The constituent methods should also be computationally efficient as Algorithm 1 fits the constituent methods on subsampled data multiple times. In this paper, we choose the Lasso, MCP and SCAD to be the default constituent methods as they are optimizing different objective functions. Methods like Elastic Net or relaxed Lasso are not selected as the default constituent methods as they are relatively similar to Lasso. The default constituent methods we choose are also computationally feasible in high-dimensional settings with efficient fitting algorithms available, and there is also a default way to compute the grid of regularization parameters to consider. For example, the R package `ncvreg` for MCP and SCAD by default computes a sequence of parameters λ with equal spacing on the log scale and of length 100, starting from the smallest value 0.001. See Section 5.1.1

for more details on the R packages used. We do not consider some two-stage methods such as the adaptive Lasso (Zou, 2006) as they are relatively slow. We also do not consider methods without default parameter tuning in R (e.g. the Dantzig selector, Candes and Tao, 2007) as it makes the comparison with other methods like delete-$n/2$ cross-validation more complicated.

CSUV can also tolerate duplicated or very similar methods, although it is not recommended due to the computational time. Including duplicated or very similar methods, though not preferable as it extends the computation time, in our experience it does not affect the variable selection performance much when the percentage parameter q is small. In our simulation, when methods that usually select similar sets as the Lasso (such as the Elastic Net or relaxed Lasso) are included, the performance of CSUV is close to when these similar methods are not included.

3.3.6 Percentile parameter q

In our simulation $q = 0$ and $q = 5$ are used with MSE as the performance measure recommended in Section 3.3.2, with $q = 0$ corresponds to selecting one single fitted model with the lowest MSE. The performance is similar with $q = 0$ and $q = 5$, although $q = 0$ provides slightly better results. When the larger percentile $q = 20$ is used, again the performance is still close to that of $q = 0$. With $q = 50$, CSUV performs poorly as it includes too many fitted models.

3.3.7 Number of repetitions B

The number of repetitions B should be large enough to stabilize the value of τ_j and at the same time it should not be too large so that Algorithm 1 can be run within a reasonable time. $B = 100$ is used in our simulation when $n = 100$ and $p = 100, 300$ and it provides a good compromise between stability and computational time.

3.4 Solution path and selection with other thresholds

The CSUV-m uses $t = 1/2$, which is equivalent to selecting the covariates for which $\tau_j \geq 1/2$. Empirically, based on our simulation results, CSUV-m provides good variable selection results by striking a good balance between false inclusion and false omission. Comparing to other variable selection methods, CSUV-m usually includes many fewer false covariates, with the trade off being that it occasionally omits some true covariates. When the analyst's focus is on performance criteria other than variable selection, for example on prediction, they may want to select more covariates. This can be done by considering other thresholds t on the sign frequency τ_j, or a threshold on the model size as described in Algorithm 2.

Algorithm 2 generates a solution path (see Definition 3) by ordering covariates from the highest to the lowest relative same sign frequency τ_j. This solution path can be regarded as a series of nested sets of covariates with increasing model sizes. Given a fixed model size s as the size threshold, Algorithm 2 selects the first s covariates on the solution path and returns them as the final selection set.

Definition 3 (CSUV solution path)**.** *The CSUV solution path orders covariates so that*

$$R_j < R_{j'} \textit{ if } \tau_j > \tau_{j'} \textit{ or } (\tau_j = \tau_{j'} \textit{ and } |\bar{\hat{\beta}}_j| > |\bar{\hat{\beta}}_{j'}|)$$

where R_j is the position of covariate j on the solution path, τ_j is the relative same sign frequency calculated in Step (9) of Algorithm 1 and $\bar{\hat{\beta}}_j$ is the average of the estimated coefficients in $\mathcal{M}$ in Step (8) of Algorithm 1.

Algorithm 2 CSUV with a given model size

```
Input: relative same sign frequency $\tau_j$ calculated in Step (9) and $\mathcal{M}$ in Step (8) of Algorithm 1; size threshold $s$.
Output: set of selected covariates.
1: obtain the solution path (Definition 3) using $\tau_j$ and $\mathcal{M}$.
2: return the first $s$ covariates ordered in Step (1), i.e.
       $\{j|R_j \leq s\}$
```

The standardization of the design matrix in Equation (1) ensures the comparison of the size of the estimated coefficients is meaningful. In the particular implementation of CSUV described in this paper, we set the size threshold s equals to the median size of the selected sets in $\mathcal{M}$ in Step (8) of Algorithm 1 and we define CSUV with this threshold as CSUV-s.

Definition 4 (CSUV-s)**.** *The CSUV method in Algorithm 2 with size threshold* $s = median(|S^{\tilde{M}^1_{(1)}}|, ..., |S^{\tilde{M}^1_{(K)}}|, ..., |S^{\tilde{M}^B_{(1)}}|, ..., |S^{\tilde{M}^B_{(K)}}|)$*, i.e. s equal to the median size of the selected sets in $\mathcal{M}$ in Step (8) of Algorithm 1 is denoted by CSUV-s, where s stands for size.*

4 CSUV visualization of uncertainty

4.1 Graphical component of CSUV

In this section, we introduce the graphical component of CSUV, which is a tool designed to illustrate the variable selection and estimation uncertainty. An example of a plot is shown in Figure 3 and the graphical tool is available interactively via a Shiny app at `https://csuv.shinyapps.io/csuv` and in the R package `CSUV`. It has the following ingredients.

- Box plots that visualize the estimated coefficient uncertainty: each box plot correspond to a covariate X_j and it shows the lower and the upper quartiles of the empirical distributions of the estimated coefficients conditional on them being non-zero, i.e. only take into account of the non-zero coefficients $\{\hat{\beta}_j^{M_k} | \hat{\beta}_j^{M_k} \neq 0, \mathrm{M}_k \in \mathcal{M}\}$ from Step (8) of Algorithm 1. Its whiskers corresponding to the 5% and 95% percentile of the non-zero estimated coefficients (default, level can be changed in the `CSUV` R package). The width of each box is proportional to the relative same sign frequency τ_j (Definition 1). The median value of the non-zero estimated coefficients is shown as a horizontal line in each box (red in the color version). The box plots are ordered according to the solution path (Definition 3). Together, the width and the vertical aspect of each box plot visually describe the variability of the corresponding estimated coefficient over the different data subsamples drawn.

- Shaded background representing τ_j: the background behind each box plot is shaded according to the relative same sign frequency τ_j of the corresponding covariate. The darker the color, the higher the value of $\lfloor 100\%\tau_j/10 \rfloor$. The actual value of τ_j is displayed in black underneath the box plots.

- Lines showing the cut-off points for variable selection by the various versions of CSUV: CSUV-m (Definition 2) selects all covariates to the left of the solid vertical line. CSUV-s (Definition 4) selects all those to the left of the dotted vertical line.

Covariates with $\tau_j < 0.1$ are not included in the plot for readability. Users wishing to have a more detailed look into the empirical distribution of the non-zero estimated coefficients can superimpose the corresponding violin plots on the box plots in the `CSUV` package. See Figure 5 as an example of such a plot.

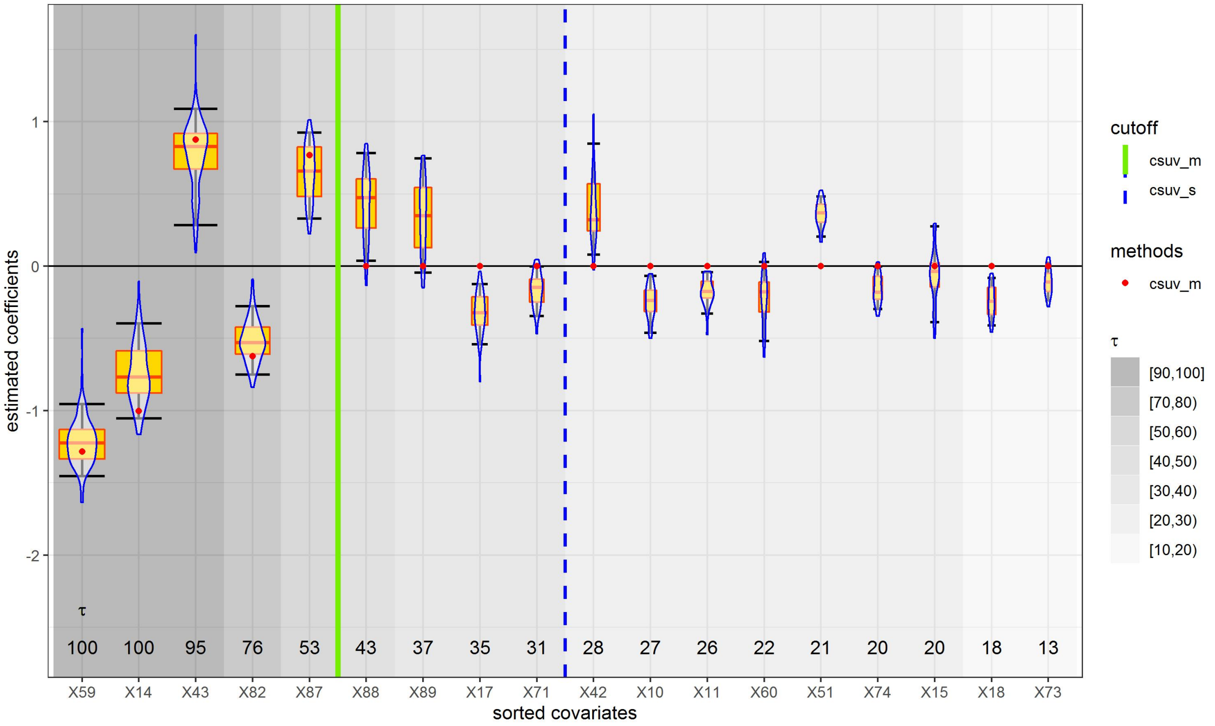

Figure 5: Same as Figure 3 but with violin plots superimposed to show the conditional kernel density.

The default plot such as the one shown in Figures 3 and 5 only considers the empirical distributions of the estimated coefficients conditional on them being non-zero (we refer to them as "conditional box plots"). This is because box plots that use all the estimated coefficients in $\mathcal{M}$ in Step (8) of Algorithm 1 that are both the zero and non-zero ones ("unconditional box plots", see Figure 6 for an example) hardly provide useful information beyond that already provided in the value of τ_j, the latter also being reflected in the width of the conditional boxes. Nevertheless, the `CSUV` package allows users to create the unconditional box plots as well.

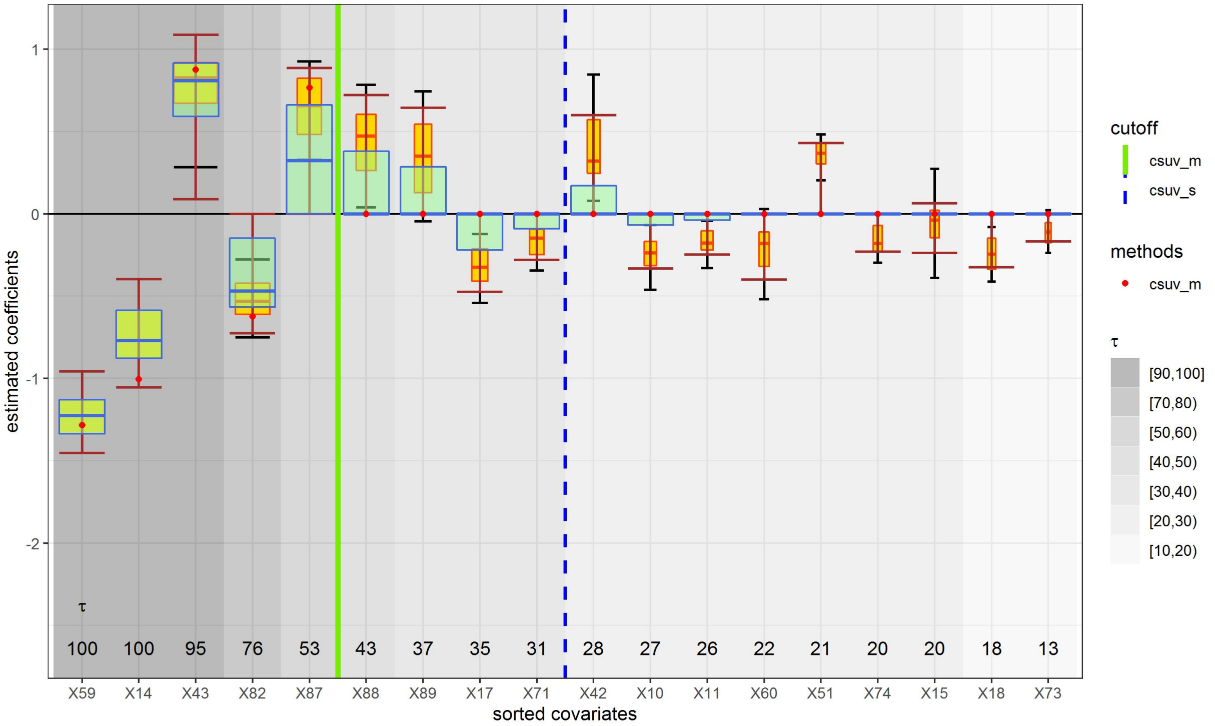

Figure 6: Same as the box plot in Figure 3 but with the semi-transparent boxes (green in the color version, usually they are wider than the conditional boxes underneath them) which represent all the estimated coefficients in $\mathcal{M}$ in Step (8) superimposed on top of it.

The `CSUV` package users wishing to compare the results returned by CSUV with any individual variable selection procedures of their choice (as long as their outputs are in a compatible format stated in the R package documentation) are also able to produce an enhanced CSUV plot, showing all of the above, and with addition of the items below.

- Graphical representation of the selection by a group of user-provided variable selection methods: the number (blue in the color version) in the bottom part of the graph shows the percentage of user-provided methods that have selected the corresponding covariate when fitting with all the observations.

- Graphical representation of the selection by any single user-provided method: the coefficient estimates by the given method are shown as empty circles (white circles with a blue outline in the color version).

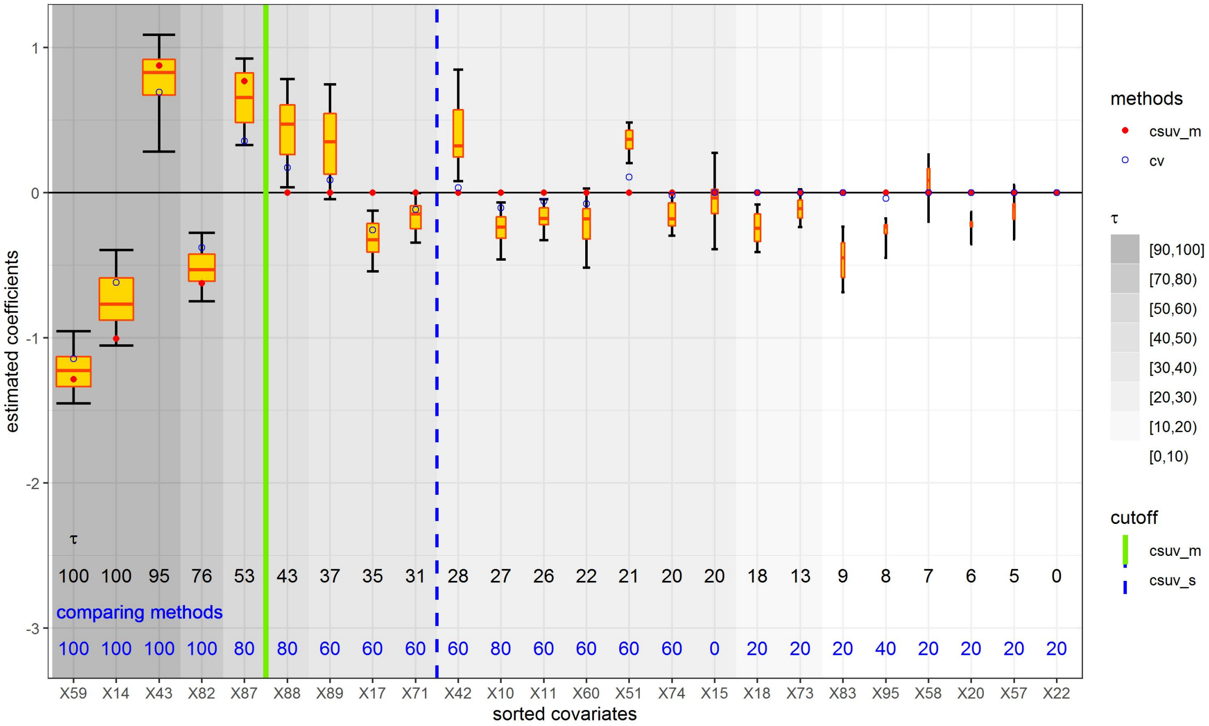

Figure 7: Example of the CSUV graphical tool with additional information of the fitting results from five individual variable selection methods (Lasso, Elastic Net, relaxed Lasso, MCP and SCAD) and delete-$n/2$ cross-validation, using simulated data from model 2 parameter setting 5 (see Section 5.1.4 for more details on the simulation setting). The plot is the same as Figure 3 with the following extra information: Empty circles (white circles with blue outline in the color version) represent the coefficients estimated by a single method (here is delete-$n/2$ cross-validation). Numbers at the bottom (blue in the color version) represent the relative percentage proportion of the group of the individual methods that select the corresponding covariates. Covariates that are not selected by any methods and $\tau_j < 0.1$ are not shown for readability.

See Figure 7 for an example for such a plot. The user can decide if a covariate should be selected by considering if the corresponding CSUV box plot, the coefficient estimated by a single method and the percentage of selection by a group of comparing methods agree to some extent.

Note that it is common that CSUV and other model selection procedures agree to some extent. For example, in Figure 7, CSUV, cross-validation and all the individual variable selection methods select the first four covariates. The methods, however, have some disagreements over the other covariates. For example, the fifth covariate is selected by both versions of CSUV, cross-validation and 80% of the individual variable selection methods, but one of the individual variable selection methods does not select the covariate. The next ten covariates are selected by the majority of the individual methods and cross-validation, but they are not chosen by CSUV-m. These non-selection decisions taken by CSUV-m are correct, as in this particular simulation setting only the first five covariates have non-zero coefficients.

4.2 CSUV assessment of uncertainty

The CSUV plot provides a graphical tool to illustrate both the selection and the estimation uncertainty in the coefficients. The uncertainty illustrated by the CSUV plot should be interpreted to originate from the randomness of ϵ. This is similar to the classical confidence intervals in fixed-p, fixed-design regression.

In this section, our focus is on the uncertainty illustration by the default conditional boxes and whiskers, and on whether and how the information they carry can be used to assess the uncertainty in selection and estimation. Therefore our mentions of "boxes" or "whiskers" in this section refer to the conditional boxes and whiskers. Roughly speaking, the selection uncertainty is represented by the width of the boxes along the x-axis, and the estimation uncertainty is represented by the range of the boxes and whiskers along the y-axis. The plot provides a graphical aid to help users to decide whether to select a covariate by considering both dimensions of the corresponding box. The following similarities between the CSUV boxes and confidence intervals can be identified.

- Both provide intervals that likely cover the value of the true coefficient.
- Both aid the users in deciding if a covariate should be selected.

However, we also highlight the following differences between the two.

- *Information content.* Unlike the classical confidence interval, which is one-dimensional, the CSUV box is two-dimensional: both its width and its range should be used in deciding whether or not to include the corresponding covariate. This is because the ranges of CSUV boxes only contain information on non-zero estimated coefficients (i.e. any zero estimates for the coefficient are not reflected in the range of the box, but only in its width). For this reason, a covariate that is rarely chosen (and in particular, is not selected by CSUV-m) may have a box plot that does not cross 0. Therefore, the width of the box plot, which is directly proportional to the same-sign frequency with which the corresponding coefficient is selected, should also be considered in deciding whether or not to include the corresponding covariate in the model.

- *Covering percentiles.* The boxes in the CSUV plot represent the upper and the lower quartiles (i.e. 25% and 75% percentile) of the non-zero estimated coefficients. By contrast, classical confidence intervals are often considered in the context of much larger coverage; frequently, 90 or 95%. With this in mind, we set the whiskers in the box plots to describe the [5%, 95%] range (of the non-zero estimated coefficients) by default. This default range for the whiskers can be changed by users in the R package `CSUV`.

Moreover, the box plots are based on the individual empirical estimated coefficients, and do not take into account the effect of the selection uncertainty in other covariates. For example, if covariates X_1 and X_2 are highly correlated, whether X_2 is selected affects the estimated coefficients of X_1. While the conditional approach considered by Loftus and Taylor (2014) and Tibshirani et al. (2016), and the debiased approached considered by Zhang and Zhang (2014) in principle can be used here, the generalization to CSUV is not straightforward and the conditional approach is computationally intensive.

The intertwining of the selection uncertainty and the estimation uncertainty makes it difficult to propose one simple interval that covers the true covariate with a given confidence level without a complicated adjustment e.g. as in Loftus and Taylor (2014) or Tibshirani et al. (2016). We instead restrict ourselves to investigating if the whiskers are useful in deciding if a covariate selected by CSUV-m should be chosen, without providing a confidence level guarantee.

Our investigation is as follows: using the simulated data from model settings 2-5 in Section 5.1.4, for covariates selected by CSUV-m, we want to find out if the covariates for which the whiskers cover zero are more likely to be the false covariates. For each realization of the simulated data, we separate the CSUV-m selected covariates into two sets: (1) whiskers covering zero, and (2) whiskers not covering zero. We then find out the frequency with which the covariates in the two sets are the true covariates.

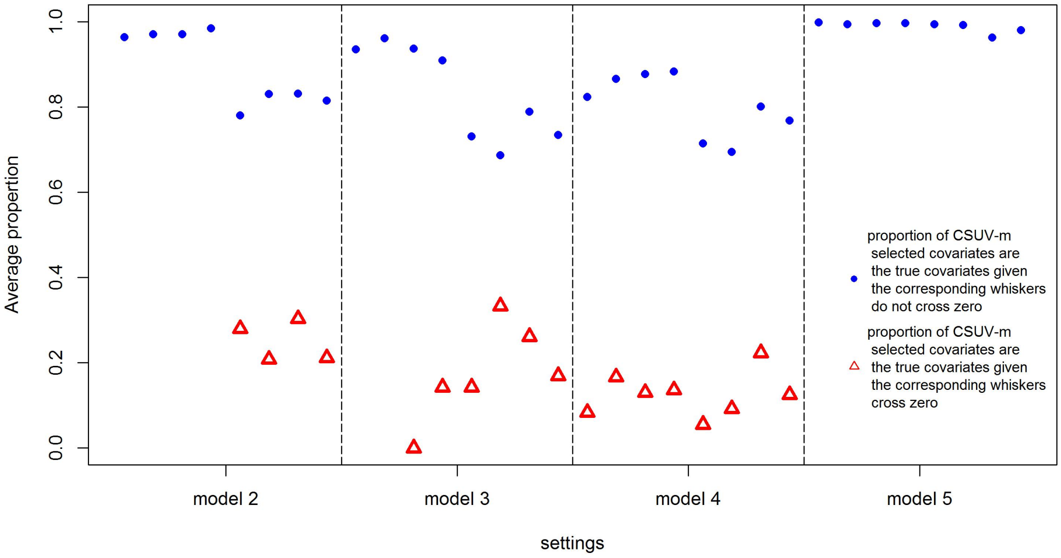

Figure 8: Average proportion of the CSUV-m selected covariates are the true covariates, using simulated data from simulation model 2-5 with eight different parameter settings under each model setting (see Section 5.1.4 for more details on the simulation settings). Circles (blue in the color version) show the average proportions of the CSUV-m selected covariates are the true covariates given the corresponding whiskers do not cross zero whereas the triangles (red in the color version) show the average proportions of the CSUV-m selected covariates are the true covariates given the corresponding whiskers cross zero. If there is no triangle for a particular setting, it means that none of the CSUV-m selected covariates have whiskers crossing zero.

The simulation results show that a covariate with whiskers crossing zero is much more likely to be a false covariate than a covariate with whiskers not crossing zero (Figure 8). This indicates that observing if the whiskers of a covariate cross zero does provide useful information in deciding if the covariate is a true one.

5 Simulation study

In this section, we evaluate the performance of CSUV with numerical examples which consists of five simulated data settings and two real datasets. The main focus of our simulation is to compare the performance of CSUV with some model selection procedures including cross-validation and information criteria as they are popular approaches when there are different variable selection methods available. We also compare the performance of CSUV under different specifications (e.g. percentile parameter $q = 0$ vs $q = 5$, different constituent methods) to verify some claims we made in Section 3.3.

5.1 Simulation settings

5.1.1 R implementations

In the simulation, we consider CSUV with different sets of constituent methods:

1. Lasso, MCP and SCAD (default)
2. Lasso, Elastic Net, relaxed Lasso, MCP and SCAD
3. MCP

The first set is our primary interest. When we mention CSUV without specifying the corresponding constituent methods, we implicitly assume that this set of methods is used. The second combination is used to verify the claim that adding some similar methods does not affect the performance too much. The third set is used to verify the claim that using more constituent methods in general provides better results. We use MCP here because in the majority of the simulation settings it has the best variable selection performance among the individual variable selection methods in terms of the F-measure and the number of false classifications.

We use publicly available R packages for the implementation of the constituent methods (Lasso, Elastic Net, relaxed Lasso, MCP, SCAD) used in CSUV. See Table 1 for the list of the corresponding R packages, functions and parameter settings used in the `CSUV` package and also in this simulation. The concavity values of SCAD and MCP are set to the value recommended by the original papers from Fan and Li (2001) and Zhang et al. (2010) respectively, which are also the default values in the `ncvreg` R package. For Elastic Net, we use $\alpha = 0.5$.

Method	R package	R function	Parameters	λ tuning
Lasso (Tibshirani, 1996)	glmnet	cv.glmnet		default 10-fold cross-validation
Elastic Net (Zou and Hastie, 2005)	glmnet	cv.glmnet	α: 0.5	default 10-fold cross-validation
Relaxed Lasso (Meinshausen, 2007)	relaxo	cvrelaxo		default 5-fold cross-validation
SCAD (Fan and Li, 2001)	ncvreg	cv.ncvreg	concavity: 3.7	default 10-fold cross-validation
MCP (Zhang et al., 2010)	ncvreg	cv.ncvreg	concavity: 3	default 10-fold cross-validation

Table 1: Variable selection methods and the corresponding R packages and functions used in CSUV

5.1.2 Methods to compare

We use eBIC and delete-$n/2$ cross-validation as the major comparing methods to CSUV. We use eBIC instead of BIC as eBIC is designed for high-dimensional data. The details of the two methods are described in Section 2.1. We also include the simulation results of each constituent method (Lasso, Elastic Net, relaxed Lasso, MCP and SCAD), VSD-minimizing method (Yang and Yang, 2017) and BIC for readers' reference.

Both eBIC and delete-$n/2$ cross-validation uses the Lasso, MCP and SCAD (i.e. the methods used in the default case of CSUV) as the base methods. eBIC selects the fitted model that minimizes the corresponding information criterion value while delete-$n/2$ cross-validation selects the method that has the lowest estimated prediction error. The R packages and the parameter values used for the base methods are the same as what we use in CSUV for fair comparison. All the variable selection methods require tuning the regularization parameter λ. Default tuning in the R packages are used to simplify the analysis and the details of the tuning are shown in Table 1. eBIC and cross-validation have their own parameters and we set them as follow: For eBIC, we set $\gamma = 0.5$, which is one of the values considered in the simulations of the original paper (Chen and Chen, 2008) and the value used in Lim and Yu (2016). For the delete-$n/2$ cross-validation, we set the number of resampling $B = 100$, which is the same as the number of iterations we use in CSUV.

For the VSD-minimizing method, we use the `glmvsd` R package to calculate the weight on each candidate model and then select the covariates that have aggregate weight greater than or equal to 0.5. Coefficients of the selected set from VSD is estimated using OLS. We use the default parameters in `glmvsd` (e.g. use the Lasso, MCP and SCAD to get the candidate models) except the weight which we use ARM instead. This is because using the default BIC to calculate the weight provides very poor results in some simulation settings.

5.1.3 Performance measures

For the datasets for which we know the true sets of covariates (i.e. simulated data and the modified real dataset), we compare the variable selection performance among different methods by the F-measure, the number of false positives (FP), number of false negatives (FN) and the total number of variable selection error (FP+FN). The F-measure is the harmonic mean of precision and recall:

$$F = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2}{\frac{TP+FP}{TP} + \frac{TP+FN}{TP}} = \frac{2TP}{2TP + FN + FP}$$

Note that comparing the above numbers individually can be misleading. For example, using only FN favors models that select a large number of covariates and using only FP favors models that select fewer number of covariates. Although the F-measure takes both precision and recall into account, assigning same weight to precision and recall is arbitrary. Nevertheless, we use the F-measure as our major measure when we compare the variable selection performance between different methods. Powers (2011) provide a detailed comparison of different evaluation methods.

Although our main focus is variable selection performance, we also compute the prediction mean square errors (MSE) on test set data and the coefficient estimation error (l_1 and l_2) for CSUV and the comparing methods.

5.1.4 Synthetic data

Set $\boldsymbol{Y} = \tilde{\boldsymbol{X}}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, $\epsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2)$. We generate observations with 100 realizations of $\boldsymbol{X}$ using the model specifications below. We then normalize $\boldsymbol{X}$ to get $\tilde{\boldsymbol{X}}$ so that all covariates have mean 0 and variance 1. Except from Model 1, the number of observation is $n = 100$, the number of predictors $p = \{100, 300\}$, the number of true covariates $s = \{5, 10\}$ and $\sigma^2 = 1$.

- **(Model 1) modified example 1 from the original Lasso paper (Tibshirani, 1996):** $\boldsymbol{\beta} = \{3, 1.5, 0, 0, 2, 0, 0, 0\}$, $p = 8$ and $n = 50$. Predictors $\boldsymbol{X}$ follow $\mathcal{N}(0, \Sigma)$, where $\Sigma_{k,m} = 0.5^{|k-m|}$ and $\sigma = \{1, 3, 6\}$. In the Lasso paper $n = 20$ but here we use $n = 50$ so that there are enough observations for subsampled fit. We include a more challenging SNR with $\sigma = 6$ ($\sigma = 3$ in the Lasso paper).

- **(Model 2) Toeplitz structure:** predictors $\boldsymbol{X}$ follow $\mathcal{N}(0, \Sigma)$, where Σ is in Toeplitz structure with $\Sigma_{k,m} = \rho^{|k-m|}$ with $\rho = \{0, 0.9\}$.

- **(Model 3) block structure:** predictors $\boldsymbol{X}$ follow $\mathcal{N}(0, \Sigma)$, where Σ is in block structure with $\Sigma_{k,m} = 1$ for $k = m$. For $k \neq m$, $\Sigma_{k,m} = 0$ except $mod_{10}(m) = mod_{10}(k)$ which $\Sigma_{k,m} = \{0.5, 0.9\}$.

- **(Model 4) factor model:** latent covariates $\phi_j, j = 1, ..., J$ are i.i.d. and follow $\mathcal{N}(0, 1)$. Each covariate is generated by $X_k = \sum_{j=1}^{J} f_{k,j}\phi_j + \eta_k$, where $f_{k,j}$, η_k are i.i.d. and follow $\mathcal{N}(0, 1)$. The number of factor $J = \{2, 10\}$.

- **(Model 5) modified example from Zhang and Yang (2015):** $\beta_j = 6/j$ for the true covariates $j = 1, ..., s$ and $\beta_j = 0$ otherwise. Predictors $\boldsymbol{X}$ follow $\mathcal{N}(0, \Sigma)$, where $\Sigma_{k,m} = \rho^{|k-m|}, \rho = \{0.5, -0.5\}$. The difference between Zhang and Yang (2015) and the model 5 here is that we use the same n and p as model 2-4.

For models 2-4, $\lfloor \frac{s}{2} \rfloor$ of the coefficient of the true s are chosen randomly from $U(0.5, 1.5)$ and $\lceil \frac{s}{2} \rceil$ of them are chosen uniformly from $U(-1.5, -0.5)$. The true βs are chosen randomly among the predictors, and once the βs are set, the same set of βs are used for all realizations.

5.1.5 Real datasets

Example 3 (Boston housing data, Harrison Jr and Rubinfeld, 1978)**.** The dataset consists of the median value of owner-occupied homes as response and $p = 13$ covariates (crime rate, proportion of residential land, etc). Number of observations is $n = 506$. The dataset is publicly available in R with the MASS package. For each simulation, half of the observations are used as the training data and the other half are used as the test set.

Example 4 (Modified riboflavin data)**.** Here we re-examine the riboflavin dataset introduced in Example 1. In order to assess the variable selection performance, we randomly permute all but 10 of the 4088 covariates in the riboflavin dataset across all the observations. The same permutation is used for all permuted covariates to keep the original dependence structure among them. The set of 10 unpermuted covariates is chosen randomly among the 200 covariates with the highest marginal correlation with the response.

The modification for the riboflavin dataset ensures that the permuted covariates cannot be the true covariates in this modified dataset. In the simulation results, we refer the 10 unpermuted covariates as the "true" covariates, although in reality they may not be the true covariates.

For the Boston data, we repeat the process for $m = 100$ times with random cuttings of the training and the test data. For the riboflavin data, we repeat the process for $m = 100$ with random selection of the 10 unpermuted covariates to stabilize the results.

5.2 Simulation results

The simulation results are summarized in Table 1-15 in the Appendix A.4. Below we discuss the simulation results in detail.

5.2.1 Verification of claims made in Section 3

In Section 3, we claim that:

- CSUV-m is designed for variable selection whereas CSUV-s is designed for better prediction.
- Performance of CSUV should be similar as long as q is small (e.g. $q = 0$ or $q = 5$).
- Including more (diverse) methods should improve the performance of CSUV.

- Including some similar methods should not worsen the performance of CSUV by much.

The simulation results support the claims above:

- CSUV-m vs CSUV-s: In general CSUV-m has better variable selection performance in terms of the F-measure. CSUV-s usually has a better prediction performance, and it also has a more stable (not too far off from the best method when CSUV is not performing particularly well) prediction performance in terms of MSE than CSUV-m. This may because CSUV-s selects a larger set of covariates than CSUV-m.

- $q = 0$ vs $q = 5$: the performance of CSUV-m when $q = 0$ and $q = 5$ is quite similar in terms of the number of covariates selected, and the prediction and variable selection performance, although $q = 0$ performs slightly better than $q = 5$.

- MCP only vs three different methods: here we only consider $q = 0$ as by using $q = 0$ we do not need to worry about the difference in terms of the number of fitted models selected (with $q = 5$ for example, the number of fitted models from three variable selection methods are around three times of the number of fitted models from a single method). In our simulation, CSUV using MCP only in general has worse performance than CSUV using three different constituent methods. In some other cases like the model 3 with parameter setting 7 and 8, both the prediction and variable selection performance of CSUV using MCP only is much worse than CSUV using three different constituent methods.

- Including some similar methods: here again we only consider $q = 0$. The results of CSUV using three different constituent methods (Lasso, MCP and SCAD) and five different methods (Lasso, Elastic Net, relaxed Lasso, MCP and SCAD, for which the Lasso, Elastic Net and relaxed Lasso are relatively similar) are very similar.

5.2.2 Comparing the performance between CSUV and some existing final model selection procedures

In the majority of settings, CSUV-m has a better variable selection performance than the eBIC, delete-$n/2$ cross-validation and VSD-minimizing method in terms of the total number of variable selection error and the F-measure, and a better coefficient estimation performance in terms of the l_1 loss. For example, out of the 36 simulation settings that we know the true set of covariates (i.e. the simulated data and the modified riboflavin dataset), CSUV-m has a higher F-measure on 33 of the settings when comparing with the delete-$n/2$ cross-validation and 32 of the settings when comparing with eBIC. CSUV-m also has higher F-measure than VSD-minimizing method on 23 settings. CSUV-m usually selects the smallest set of covariates when comparing with eBIC or delete-$n/2$ cross-validation and the individual variable selection methods. In some cases like model 4 parameter setting 6, it selects a much smaller set of covariates than the truth. While this worsens the prediction performance of CSUV-m and we may view it as a limitation of CSUV-m, it may well due to the limitation of variable selection as a whole: Other methods which select much larger sets of covariates usually include a few more true covariates but inevitably they also include many more false covariates. They may perform better than CSUV-m in terms of prediction, but CSUV-m in general outperforms them in terms of variable selection.

The performance of CSUV-s, on the other hands, is much more difficult to draw conclusions on. CSUV-s is better than delete-$n/2$ cross-validation in terms of variable selection. When comparing with eBIC, while it performs better than eBIC in one measure in some settings, it performs worse than eBIC in some other settings with the same measure.

One encouraging result about CSUV is that in many simulation settings like model 2, CSUV-m outperforms not only the final model selections procedures but it also outperforms *all* individual constituent methods in terms of the F-measure and the total number of variable selection error. In some simulation settings, CSUV performs better than the best individual variable selection method in terms of both prediction and variable selection measured by F-measure. For example in model 2, there are quite a few parameter settings (e.g. parameter setting 2) that the MSE of CSUV is lower and the F-measure is higher than all individual variable selection methods.

For the variable selection performance on the real data, both versions of CSUV perform very well on the riboflavin data example. CSUV-s has the best performance in terms of F-measure and the total number of variable selection error.

5.3 Analysis of the selection by CSUV

5.3.1 Reasons for selected set to be small for CSUV-m

The number of covariates selected by CSUV-m is often small when comparing with other methods and the true size. Investigation into the collection of fitted models $\mathcal{M}$ shows that for many simulation settings, the fitted models in $\mathcal{M}$ can be very different in terms of variable selection. Sometimes all fitted models in $\mathcal{M}$ select different sets of covariates. When the selection decision is so different among $\mathcal{M}$, it is very likely that only a few covariates will have $\tau_j \geq 1/2$. This causes the number of covariates chosen by CSUV-m to be small. Whether a small selected set is desirable depends on the purpose of variable selection. Selecting small(er) number of covariates by this selection rule may cause omission of some true covariates and possibly exclusion of some false covariates that are helpful for prediction. This may result in poor prediction in some situations. On the other hand, the set of covariates selected by CSUV-m often includes fewer false positives than other variable selection methods, as only covariates that are selected by the majority of the subsampled fits are included in CSUV-m.

6 Conclusion

Many variable selection methods are available. However, there is no clear guideline on how to select which method to use with the data at hand, or how we can trust the set of covariates selected by a method. In practice, cross-validation and information criteria may be used to select the final models: Zhang and Yang (2015) advocate to use the delete-$n/2$ cross-validation and Chen and Chen (2008) extend the use of BIC to high-dimensional data (eBIC).

In this paper we suggest a competitive alternative to these two procedures. We also provide a graphical illustration of the selection uncertainties. CSUV does not attempt to select the best method or to find the optimal regularization parameter. Instead we aggregate the fitted results from different variable selection methods via subsampling, and use a graphical tool to illustrate the uncertainties in selection and estimation. CSUV is very general and can be used with different variable selection methods. The simulation results show that CSUV in general outperforms the delete-$n/2$ cross-validation and eBIC in terms of variable selection. We also show that the graphical tool of CSUV has the capability to aid analysts in variable selection.

SUPPLEMENTARY MATERIAL

Appendix: Proofs for Proposition 1 and 2, heat maps to illustrate selection disagreements and detailed simulation results. (.pdf file)

References

Bach, F. R. (2008). Bolasso: model consistent lasso estimation through the bootstrap. In *Proceedings of the 25th international conference on Machine learning*, pp. 33–40. ACM.

Baranowski, R., Y. Chen, and P. Fryzlewicz (2020). Ranking-based variable selection for high-dimensional data. *Statistica Sinica*.

Beale, E., M. Kendall, and D. Mann (1967). The discarding of variables in multivariate analysis. *Biometrika 54*(3-4), 357–366.

Candes, E. and T. Tao (2007). The Dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 2313–2351.

Chatterjee, A. and S. N. Lahiri (2011). Bootstrapping Lasso estimators. *Journal of the American Statistical Association 106*(494), 608–625.

Chen, J. and Z. Chen (2008). Extended Bayesian information criteria for model selection with large model spaces. *Biometrika*, 759–771.

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

Fan, J. and J. Lv (2010). A selective overview of variable selection in high dimensional feature space. *Statistica Sinica 20*(1), 101.

Harrison Jr, D. and D. L. Rubinfeld (1978). Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management 5*(1), 81–102.

Javanmard, A., A. Montanari, et al. (2018). Debiasing the Lasso: Optimal sample size for gaussian designs. *The Annals of Statistics 46*(6A), 2593–2622.

Lee, J. D., D. L. Sun, Y. Sun, J. E. Taylor, et al. (2016). Exact post-selection inference, with application to the lasso. *The Annals of Statistics 44*(3), 907–927.

Lim, C. and B. Yu (2016). Estimation stability with cross-validation (escv). *Journal of Computational and Graphical Statistics 25*(2), 464–492.

Liu, H., B. Yu, et al. (2013). Asymptotic properties of lasso+ mls and lasso+ ridge in sparse high-dimensional linear regression. *Electronic Journal of Statistics 7*, 3124–3169.

Loftus, J. R. and J. E. Taylor (2014). A significance test for forward stepwise model selection. *arXiv preprint arXiv:1405.3920*.

Mares, M. A., S. Wang, and Y. Guo (2016). Combining multiple feature selection methods and deep learning for high-dimensional data. *Transactions on Machine Learning and Data Mining 9*, 22–45.

Meinshausen, N. (2007). Relaxed lasso. *Computational Statistics & Data Analysis 52*(1), 374–393.

Meinshausen, N. and P. Bühlmann (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72*(4), 417–473.

Nan, Y. and Y. Yang (2014). Variable selection diagnostics measures for high-dimensional regression. *Journal of Computational and Graphical Statistics 23*(3), 636–656.

Pohjalainen, J., O. Räsänen, and S. Kadioglu (2015). Feature selection methods and their combinations in high-dimensional classification of speaker likability, intelligibility and personality traits. *Computer Speech & Language 29*(1), 145–171.

Powers, D. (2011). Evaluation: From precision, recall and fmeasure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technologies 2*, 37–63.

Shah, R. D. and R. J. Samworth (2013). Variable selection with error control: another look at stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 75*(1), 55–80.

Stamey, T. A., J. N. Kabalin, J. E. McNeal, I. M. Johnstone, F. Freiha, E. A. Redwine, and N. Yang (1989). Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. ii. radical prostatectomy treated patients. *The Journal of urology 141*(5), 1076–1083.

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

Tibshirani, R. J., A. Rinaldo, R. Tibshirani, L. Wasserman, et al. (2018). Uniform asymptotic inference and the bootstrap after model selection. *The Annals of Statistics 46*(3), 1255–1287.

Tibshirani, R. J., J. Taylor, R. Lockhart, and R. Tibshirani (2016). Exact post-selection inference for sequential regression procedures. *Journal of the American Statistical Association 111*(514), 600–620.

Tsai, C.-F. and Y.-C. Hsiao (2010). Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches. *Decision Support Systems 50*(1), 258–269.

Van de Geer, S., P. Bühlmann, Y. Ritov, R. Dezeure, et al. (2014). On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics 42*(3), 1166–1202.

Wang, X., P. Peng, and D. B. Dunson (2014). Median selection subset aggregation for parallel inference. In *Advances in Neural Information Processing Systems*, pp. 2195–2203.

Yang, W. and Y. Yang (2017). Toward an objective and reproducible model choice via variable selection deviation. *Biometrics 73*(1), 20–30.

Yang, Y. (2001). Adaptive regression by mixing. *Journal of the American Statistical Association 96*(454), 574–588.

Yuan, M. and Y. Lin (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68*(1), 49–67.

Yuan, Z. and Y. Yang (2005). Combining linear regression models: When and how? *Journal of the American Statistical Association 100*(472), 1202–1214.

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

Zhang, C.-H. and S. S. Zhang (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.

Zhang, Y. and Y. Yang (2015). Cross-validation for selecting a model selection procedure. *Journal of Econometrics 187*(1), 95–112.

Zhao, P. and B. Yu (2006). On model selection consistency of lasso. *Journal of Machine learning research 7*(Nov), 2541–2563.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association 101*(476), 1418–1429.

Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67*(2), 301–320.