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

June 11, 2021

Abstract

We propose Combined Selection and Uncertainty Visualizer (CSUV), which visualises selection uncertainties for covariates in high-dimensional linear regression by exploiting the (dis)agreement among different base selectors. Our proposed method highlights covariates that get selected the most frequently by the different base variable selection methods on subsampled data. The method is generic and can be used with different existing variable selection methods. We demonstrate its performance using real and simulated data. The corresponding R package `CSUV` is at `https://github.com/christineyuen/CSUV`, and 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, with techniques such as best subset selection (Beale et al., 1967), Lasso (Tibshirani, 1996), Elastic Net (Zou and Hastie, 2005), Group Lasso (Yuan and Lin, 2006), SCAD (Fan and Li, 2001), MCP (Zhang, 2010) and a handful of others having gained widespread popularity. Fan and Lv (2010) provide a detailed review of different variable selection methods in high-dimensional settings. On the uncertainty quantification front, there has been a growing focus on post-selection inference. Van de Geer et al. (2014), Zhang and Zhang (2014) and Javanmard and Montanari (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 and Yu (2013) suggest using bootstrapping on some existing variable selectors.

In this paper, we propose a simple approach to assessing and visualizing 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 $S = \{j \in \{1, ..., p\} : \beta_j \neq 0\}$, $s = |S| \ll p$.

Given a dataset, how to select the best variable selection *method* remains an open and yet very important question to ask. Various theoretical performance guarantees are available for a range of methods (e.g. the irrepresentable condition (Zhao and Yu, 2006) is sufficient and almost necessary for the Lasso to be sign consistent) 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 practical use in method selection. To illustrate the uncertainty associated with method selection, 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 for more details). The selection results are shown in Figures 1 and 2. Figure 1 shows that for the riboflavin dataset the sets of

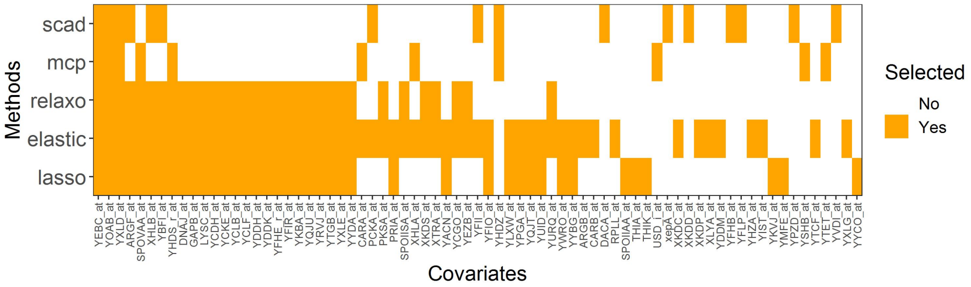

Figure 1: Graphical illustration of selections by different variable selection methods (Lasso, Elastic Net, relaxed Lasso, MCP and SCAD) with default tuning using the riboflavin dataset from Example 1. Covariates that are not selected by any methods are not shown in the graph for readability.

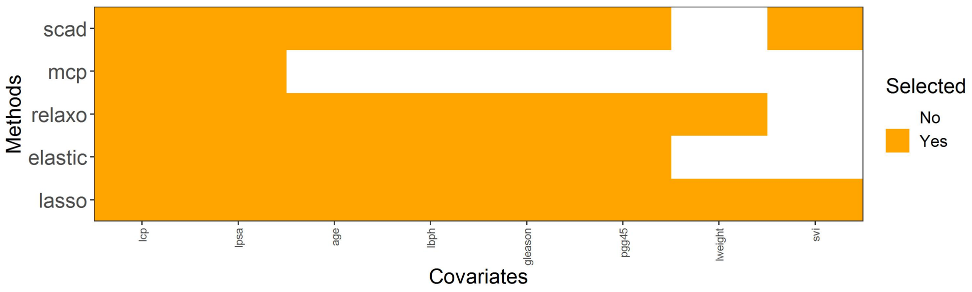

Figure 2: Graphical illustration of selections by different variable selection methods (Lasso, Elastic Net, relaxed Lasso, MCP and SCAD) with default tuning using the prostate dataset from Example 2.

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.4 show that selection disagreement manifests itself across different 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. However, notwithstanding the usefulness of using a single, well-performing variable selection method, it is tempting to ask whether more can 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. In this paper, we propose a new tool for uncertainty visualization associated with variable selection, termed Combined Selection and Uncertainty Visualizer (CSUV). CSUV combines, in a particular way, a number of different base variable selection methods into the solution path of 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 best-performing model. 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 goal of variable selection in our context is purely interpretative: we are interested in visualising the uncertainty associated with including each individual covariate into any model. The user aware of the data-analytic context can then make their own decision as to whether or not to include each covariate, based on the task at hand (e.g. prediction). This task is facilitated by our approach, as it returns what can be interpreted as a measure of importance of each covariate, and a natural ordering of the covariates according to the certainty of their inclusion. Based on the output of the procedure, the user can select a number of sets of covariates, if this is preferred – for example, by trying different combinations of the 'most certain' variables.

An important component of CSUV is a graphical tool designed to visualize the selection uncertainty by using disagreement among the different model fits. See Figure 3 for an example of a graphical output of CSUV. The plot shows the frequency with which each covariate is selected and the variability of the estimated coefficients.

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 solution path, 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. In Section 5, we present the simulation results.

2 Related work

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.

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. 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.

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.

One high-level difference between CSUV and stability selection is the output from both procedures: CSUV is designed to visualise the uncertainty associated with the inclusion of each covariate into (any) linear model, whereas stability selection does not have a similar function. To this end, CSUV uses information from the disagreement between different model fits, a point of view that is absent from stability selection. In addition, CSUV has a mechanism for only considering the disagreement between 'good' model fits; this is achieved by filtering out (predictively) 'bad' model fits. (Clearly, there would be little point in considering disagreement between a number of unsatisfactory model fits.) This point of view is not present in stability selection, which does not automatically have a mechanism for filtering out unsatisfactory base methods or model fits.

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. 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 show that in general our method outperforms the VSD-minimizing model.

Hahn and Carvalho (2015) provide an informative review of Bayesian approaches to variable selection in linear models, a formalism in which the data-analytic output consists of probabilities assigned to different models given data. Therefore, these approaches provide natural measures of model uncertainty and should be seen as an alternative to the approach taken in CSUV and other frequentist methods. As reviewed in Hahn and Carvalho (2015), two potential weaknesses of the Bayesian approach in this context are: subjectivity in the choices of the various priors, and the fact that posterior sampling is usually computationally intensive. CSUV attempts to mitigate both these issues by minimising user involvement in the selection of its parameters, and by using computationally efficient constituent methods.

3 CSUV solution path methodology

3.1 The CSUV algorithm

The first goal of this paper is to use the similarity of fits from different methods to obtain the ranking of covariates, alternatively referred to as a solution path, which can then (if desired) be used for the purpose of variable selection. CSUV achieves this by following the simple aggregation principles below.

- 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: rank the covariates according to decreasing θ_j's. If variable selection is required, select those that correspond to the highest θ_j's, e.g. $\{X_j : \theta_j \geq 0.5\}$.

At the same time, CSUV uses the following additional mechanisms.

- Only include the fitted models that exhibit good performance (see Section 3.2.2).

- Repeat the fitting on subsampled data.

This is done, respectively, to discourage CSUV from focusing on poorly-performing methods, and to incorporate the variability in selection caused by the variability in data.

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.

The CSUV solution path procedure, detailed in Algorithm 1, 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 (see Section 3.2.4 for more details). Then, use the test set to calculate the performance of the fitted models and retain the best model (see Section 3.2.2 for more details on performance measures). Repeat the process many times. Order the covariates according to the relative same sign frequency τ_j, which we now define.

Definition 1 (Relative same sign frequency τ_j)**.** *Assume 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 CSUV solution path (and variable selection)

```
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 $B$; percentage of data used in training set $w\%$; performance
    measure; (for variable selection only) frequency threshold $t$.
Output: ranking of covariates; (for variable selection only) 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) according to the given performance measure. Retain the best fitted model and
       denote it as $\hat{M}^b_{(1)}$.
6: end for
7: Denote the set of retained fitted models by $\mathcal{M} = \{\hat{M}^1_{(1)}, \ldots, \hat{M}^B_{(1)}\}$.
8: Calculate the relative same sign frequency $\tau_j$ for each variable $j$, which defines the
   ranking of covariates.
9: If variable selection is required, select the covariates $\hat{S}$ defined by

       $\hat{S} = \{j : \tau_j \geq t\}.$
```

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, which makes it feasible for high-dimensional data analysis.

3.2 Specifications for CSUV variable selection

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

3.2.1 Coefficient estimation

If Algorithm 1 is tasked with variable selection, it 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, the default option in our code is to use ridge regression to estimate the coefficients by cross-validation (with the default cross-validation setting from the `glmnet` R package); however, this should be treated as an exception mode as the number of covariates selected is almost always much smaller than the number of observations.

3.2.2 Performance measure

Step (5) of Algorithm 1 aims to select the fitted models based on their variable selection performance. 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)).

3.2.3 Percentage of data used in training

Following Yang (2001), Yuan and Yang (2005) and Zhang and Yang (2015), we use $w\% = 50\%$ of the data for fitting and the remaining 50% for testing by default; 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, we found that departures from 50:50 were occasionally beneficial; many procedures display at least some sensitivity to the training/test split, so this phenomenon in our case is neither unique nor unexpected. For example, we applied CSUV in a higher-dimensional case involving $p = 1000$ (with $n = 100$) and the 75:25 split was preferred there – possibly because it was particularly important for the algorithm to maximise the size of the training sample when there were so many variables. On the other hand, when n was large in relation to p ($n = 1000$ and $p = 100$), the performance of CSUV with three different splits ratios (25:75, 50:50 and 75:25) was similar, with 25:75 and 50:50 performing slightly better than 75:25, quite possibly because with such a large n, even 25:75 was seen as sufficient for training.

3.2.4 Constituent variable selection methods

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 base methods should also be computationally efficient they are fitted 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 used by default as they are relatively similar to Lasso. Our default base methods 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. We do not consider two-stage methods such as the adaptive Lasso (Zou, 2006) as they are relatively slow, or methods without default parameter tuning in R (e.g. the Dantzig selector, Candes and Tao, 2007) as they make comparison with other methods like delete-$n/2$ cross-validation more complicated.

CSUV can tolerate duplicated or very similar methods, although it is not recommended due to the computational time. In our experience, including duplicated or very similar methods tends not to affect the variable selection performance. In our simulations, when methods that usually select similar sets to the Lasso (such as the Elastic Net or relaxed Lasso) are included, the performance of CSUV does not change much.

A suggestion made by a referee is to incorporate a randomised method (or a selection thereof) as a constituent variable selector in CSUV, to increase the diversity of the constituent methods. In a randomised method, each model uses a random subset of variables, chosen according to a particular recipe (akin to a Bayesian stochastic search). However, it is not obvious to us how to select reasonable random models in a computationally efficient way, and we therefore leave this interesting idea for future work.

3.2.5 Number of repetitions

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. We use $B = 100$ in our experiments.

3.2.6 Frequency threshold

The frequency threshold t features in Step (9) of Algorithm 1 and is used when variable selection is required. The CSUV method described by Algorithm 1 and using $t = 1/2$ is denoted by CSUV-m, where the "m" in CSUV-m stands for median, because selecting covariates with $\tau_j \geq 1/2$ is equivalent to selecting covariates with a non-zero median in

$\mathcal{M}$. The choice $t = 1/2$ is optimal in the sense of minimising certain distance functions; the details are in Propositions 1 and 2 in the Appendix. Empirically, even though CSUV-m strikes a good balance between false inclusion and false omission, compared to other variable selection methods it 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 which we introduce next.

3.3 Solution path and alternative variable selection

Algorithm 1 generates a solution path (formally defined below) 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.

Definition 2 (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 (7) of Algorithm 1.

The standardization of the design matrix in Equation (1) ensures the comparison of the size of the estimated coefficients is meaningful. The nested character of the solution path makes it easy to entertain alternatives to CSUV-m. In particular, consider replacing line (9) of Algorithm 1 by $\hat{S} = \{j : R_j \leq s\}$, where s is a given size thresholds which specifies the model size. In the particular implementation of CSUV described in this paper, we set the size threshold s equal to the median size of the selected sets in $\mathcal{M}$ in Step (7) of Algorithm 1 and we define CSUV with this threshold as CSUV-s.

4 CSUV visualization of uncertainty

4.1 Graphical component of CSUV

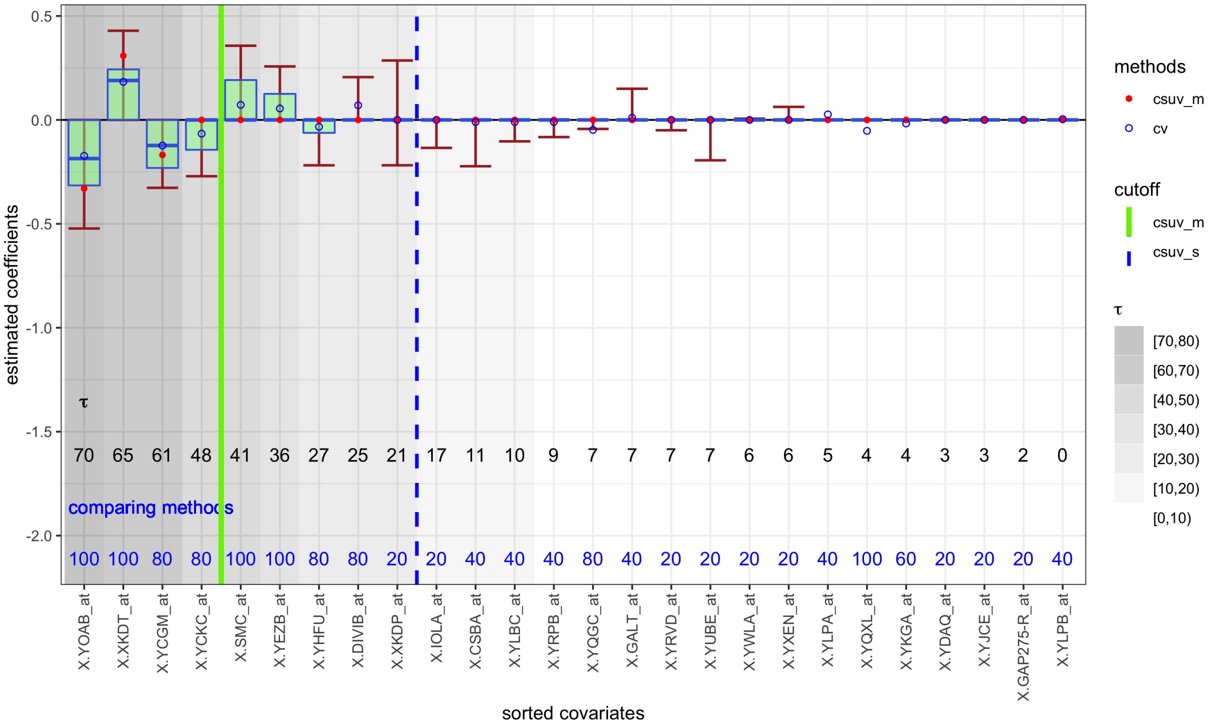

Figure 3: CSUV graphical tool, riboflavin data from Example 4 (modified Example 1). Box plots: empirical distributions of the estimated coefficients; whiskers: their 5% and 95% percentiles. Ordering of the covariates according to the CSUV solution path. Numbers at the bottom of the graph (black): τ_j times 100; shade in the background: level of τ_j with ranges as in the legend. Numbers at the bottom of the graph (blue): percentage proportion of user-provided methods (here: Lasso, Elastic Net, relaxed Lasso, MCP, SCAD) that select the corresponding covariate. Red (full) dots: coefficients estimated by CSUV-m; empty dots: coefficients estimated by a single user-provided method (here: delete-$n/2$ cross-validation). Solid vertical line (green): cut-off of CSUV-m; dotted vertical line (blue): cut-off of CSUV-s. Remaining covariates not shown for readability.

In this section, we introduce the graphical component of CSUV, 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 at `https://csuv.shinyapps.io/csuv` and in the R package `CSUV` (Yuen, 2021). It has the following ingredients.

Box plots that visualize the estimated coefficient uncertainty. Each box plot corresponds to a covariate X_j and shows the lower and the upper quartiles of the empirical distributions of the estimated coefficients. Figure 3 shows the unconditional distributions for each covariate. The `CSUV` package also has a conditional option, in which the box plots show the distributions conditional on the estimated coefficients being non-zero. In addition, users desiring more details of the empirical distribution of estimated coefficients are able to superimpose the corresponding violin plots on the box plots in the `CSUV` package. The extent of the whiskers, set by default to the 5 and 95 percentiles of each estimated coefficient, can be adjusted in the `CSUV` package. The median value of each estimated coefficient is shown as a horizontal line in each box (blue in the color version). The box plots are ordered according to the solution path (Definition 2).

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 selects all covariates to the left of the solid vertical line; CSUV-s selects all those to the left of the dotted vertical line.

Figure 3 illustrates (dis)agreement between the constituent variable selectors, in the sense of showing the range of estimates for each linear parameter obtained for each sub-sample, for each method that wins a prediction competition on that sub-sample. This ensures that only ‘good’ methods are evaluated (there is little point in including less adequate methods for the purpose of uncertainty visualisation, if a better method can be run instead). If the chosen ‘good’ methods exhibit large variation in terms of the parameter value associated with a given covariate, this can be viewed as a sign of high uncertainty as to whether the covariate should enter any linear model for the data (given that various well-performing predictive methods disagree as to its value).

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 able to produce an enhanced CSUV plot, showing all of the above, and with addition of the items below, which also appear in Figure 3.

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).

5 Simulation study

5.1 Settings

In this section, we evaluate the variable selection performance of CSUV. 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) Lasso, Elastic Net, relaxed Lasso, MCP, SCAD and trees; (4) MCP only. 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 and the third combinations are used to verify the claim that adding some similar or inappropriate methods does not affect the performance too much. The fourth 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, tree) used in CSUV. Further details are in Section A.6.1. We use eBIC and delete-$n/2$ cross-validation as the major competitors to CSUV. We use eBIC instead of BIC as eBIC is designed for high-dimensional data. 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. Further details are in Section A.6.2.

For the datasets for which we know the true sets of covariates (i.e. simulated data and the modified real datasets), 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). 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. Further details are in Section A.6.3.

Our simulations involving synthetic data use five different linear models, details of which are in Section A.6.4. Our real datasets are described below.

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 (randomised) are used as the training data and the other half are used as the test set; $m = 100$ repetitions.

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. In the simulation results, we refer the 10 unpermuted covariates as the "true" covariates. We repeat the process $m = 100$ times with random selection of the 10 unpermuted covariates.

5.2 Results

The simulation results are summarized in Tables 1–3 and also Tables 3–19 in the Appendix A.7. A qualitative summary is provided below.

methods	f	FP+FN	FP	FN	pred.err	l1.diff	l2.diff	size
lasso	0.39	25.02	24.45	0.57	2.83	4.48	1.12	30.95
elastic net	0.33	32.64	32.14	0.5	3.01	5.32	1.24	38.71
relaxed lasso	0.58	12.29	11.35	0.94	2.87	3.98	1.09	17.48
mcp	0.7	4.86	3.32	1.54	2.93	3.2	1.13	8.84
scad	0.63	8.04	6.79	1.25	2.91	3.05	1.09	12.6
vsd	0.72	3.2	0.48	2.72	3.9	3.38	1.32	4.82
bic	0.64	10.18	8.98	1.2	2.85	3.29	1.06	14.85
ebic	0.71	5.4	4.02	1.37	2.87	3.04	1.08	9.71
cv	0.54	17.25	16.61	**0.64**	**2.72**	3.74	**1.01**	23.03
csuv.m	**0.77**	**2.7**	**0.65**	2.05	3.23	**2.76**	1.1	5.66
csuv.s	0.7	5.92	4.69	1.23	2.82	3.2	1.04	10.52
csuv.m.25	0.56	4.35	0.09	4.26	5.25	4.36	1.74	2.9
csuv.m.75	0.79	2.98	1.77	1.22	2.73	2.55	0.95	7.61
csuv.m.all	0.77	2.75	0.78	1.97	3.16	2.75	1.09	5.88
csuv.m.mcp	0.68	3.34	0.15	3.19	4.31	3.47	1.38	4.02
csuv.m.with.tree	0.77	2.75	0.77	1.98	3.18	2.75	1.09	5.85

Table 1: Model summary: performance of CSUV and methods it compares with. Variable selection performance in terms of F-measure (f), total error (FP+FN), false positives (FP) and false negatives (FN), prediction error in terms of mse (pred.err), estimation error in terms of l1 and l2 distance (l1.diff and l2.diff) and average model size (size) are shown. The numbers are based on 100 simulations. The last 7 rows are the performance of CSUV with different parameters (e.g. csuv.m.mcp corresponds to CSUV with MCP as constituent method, and csuv.m.25 corresponds to CSUV with splitting ratio 25:75). Bold numbers: best result among delete-$n/2$ cross validation, eBIC and CSUV using Lasso, MCP and SCAD; underlined numbers: worst among those. Standard errors are shown inside the parentheses.

CSUV-m vs CSUV-s. In general CSUV-m has better variable selection performance in terms of the F-measure with our synthetic datasets. CSUV-s usually has a better prediction performance, which is also more stable (not too far off from the best method when CSUV is not performing particularly well) in terms of MSE than CSUV-m. This may because CSUV-s selects a larger set of covariates than CSUV-m. For the real dataset Riboflavin, the reverse is observed.

MCP only vs three different methods. CSUV using MCP only in general has worse performance than CSUV using three different constituent methods; sometimes by a large margin (e.g. model 3 with parameter settings 7 and 8) in terms of both prediction and variable

methods	pred.err	size
lasso	26.08 (0.39)	12.39 (0.09)
elastic net	26.12 (0.4)	12.38 (0.1)
relaxed lasso	26.53 (0.42)	11.21 (0.14)
mcp	26.23 (0.39)	11.31 (0.17)
scad	26.14 (0.39)	11.56 (0.13)
vsd	28.54 (0.43)	5.98 (0.17)
bic	26.27 (0.4)	10.92 (0.16)
ebic	26.27 (0.4)	10.92 (0.16)
cv	**26.08** (0.39)	12.36 (0.09)
csuv.m	26.59 (0.4)	10.04 (0.16)
csuv.s	26.65 (0.4)	9.98 (0.16)
csuv.m.25	28.47 (0.41)	6.52 (0.19)
csuv.m.75	26.49 (0.4)	10.17 (0.16)
csuv.m.all	26.63 (0.4)	9.99 (0.17)
csuv.m.mcp	26.38 (0.4)	10.46 (0.16)
csuv.m.with.tree	26.66 (0.4)	9.94 (0.17)

Table 2: Boston data: performance of CSUV and methods it compares with. See caption to Table 1 for explanatory details.

selection.

Including similar or inappropriate methods. The results of CSUV using three different constituent methods (Lasso, MCP and SCAD), five different methods (Lasso, Elastic Net, relaxed Lasso, MCP and SCAD, for which the Lasso, Elastic Net and relaxed Lasso are relatively similar) and with an additional method inappropriate for the linear regression setting (trees) are very similar.

Training:test split ratio. CSUV with higher training:test ratios performs better than that with the lower ratio of 25:75 on the synthetic datasets and the Riboflavin data. When $p \gg n$, CSUV with the 75:25 split has the best performance, but the opposite is true when p is relatively small in comparison with n.

CSUV vs other final model selection procedures. In the majority of settings, CSUV-m has a better variable selection performance than eBIC, delete-$n/2$ cross-validation and VSD in terms of FP+FN and the F-measure, and a better coefficient estimation performance in terms of the l_1 loss. For example, out of the 48 simulation settings in which we know the true set of covariates, CSUV-m has a higher F-measure in, respectively, 45, 41 and 32 of the settings compared to delete-$n/2$ cv, eBIC and VSD. CSUV-m usually selects the smallest

methods	f	FP+FN	FP	FN	size
lasso	0.46 (0.01)	19.96 (1.45)	16.88	3.08	23.8 (1.48)
elastic net	0.45 (0.01)	22.59 (1.51)	20.44	2.15	28.29 (1.53)
relaxed lasso	0.56 (0.02)	10.15 (0.81)	5.85	4.3	11.55 (0.97)
mcp	0.45 (0.01)	8.83 (0.19)	2.53	6.3	6.23 (0.24)
scad	0.48 (0.02)	12.34 (0.59)	7.69	4.65	13.04 (0.49)
vsd	NaN (NA)	10 (0)	0	10	0 (0)
bic	0.42 (0.01)	16.38 (1.48)	11.17	5.21	15.96 (1.67)
ebic	0.44 (0.01)	11.21 (1.18)	5.05	6.16	8.89 (1.31)
cv	0.45 (0.01)	19.54 (1.44)	16.09	**3.45**	22.64 (1.53)
csuv.m	0.45 (0.01)	7.12 (0.13)	**0.02**	7.1	2.92 (0.13)
csuv.s	**0.64** (0.01)	**5.59** (0.19)	1	4.59	6.41 (0.33)
csuv.m.25	0.18 (0)	9.92 (0.03)	0	9.92	0.08 (0.03)
csuv.m.75	0.59 (0.01)	5.95 (0.14)	0.37	5.58	4.79 (0.16)
csuv.m.all	0.49 (0.01)	6.74 (0.14)	0.02	6.72	3.3 (0.14)
csuv.m.mcp	0.28 (0.01)	8.61 (0.09)	0	8.61	1.39 (0.09)
csuv.m.with.tree	0.48 (0.02)	6.77 (0.14)	0.02	6.75	3.27 (0.14)

Table 3: Riboflavin data with permutation: performance of CSUV and methods it compares with. See caption to Table 1 for explanatory details. VSD did not execute correctly.

set of covariates when compared with eBIC, 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 its limitation, it may be 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 also many more false ones. 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 is more difficult to draw conclusions on. Overall, CSUV-s appears better than delete-$n/2$ cross-validation in terms of variable selection, but roughly comparable to eBIC.

One encouraging result about CSUV is that in many simulation settings like model 2, CSUV-m outperforms not only the final model selection procedures, but also all individual constituent methods in terms of the F-measure and FP+FN. In some simulation settings, CSUV performs better than the best individual variable selection method in terms of both prediction and the F-measure. For example in model 2, there are several parameter settings (e.g. 2) in which 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 the F-measure and the total variable selection error.

SUPPLEMENTARY MATERIAL

Appendix: Some optimality results on CSUV-m; heat maps to illustrate selection disagreement; CSUV assessment of uncertainty; further details of simulation results.

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 30*(3), 1485–1516.

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. *Annals of statistics 35*(6), 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 95*(3), 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.

Hahn, P. and C. Carvalho (2015). Decoupling shrinkage and selection in Bayesian linear models: A posterior summary perspective. *Journal of the American Statistical Association 110*, 435–448.

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. and A. Montanari (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, and J. E. Taylor (2016). Exact post-selection inference, with application to the lasso. *The Annals of Statistics 44*(3), 907–927.

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

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.

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) 58*(1), 267–288.

Tibshirani, R. J., A. Rinaldo, R. Tibshirani, and L. Wasserman (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, and R. Dezeure (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.

Yuen, C. (2021). *CSUV: Combined Selection and Uncertainty Visualiser (CSUV).* R package version 0.1.1.

Zhang, C.-H. (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.