Post Lasso Stability Selection for High Dimensional Linear Models

Niharika Gauraha[1], Tatyana Pavlenko[2] and Swapan K. Parui[3]
[1]*Systems Science and Informatics Unit, Indian Statistical Institute, Bangalore, India*
[2]*Mathematical Statistics, KTH Royal Institute of Technology, Stockholm, Sweden*
[3]*Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata, India*
niharika@isibang.ac.in, pavlenko@math.kth.se, swapan@isical.ac.in

Keywords: Lasso, Weighted Lasso, Variable Selection, Stability Selection, High Dimensional Data.

Abstract: Lasso and sub-sampling based techniques (e.g. Stability Selection) are nowadays most commonly used methods for detecting the set of active predictors in high-dimensional linear models. The consistency of the Lasso-based variable selection requires the strong irrepresentable condition on the design matrix to be fulfilled, and repeated sampling procedures with large feature set make the Stability Selection slow in terms of computation time. Alternatively, two-stage procedures (e.g. thresholding or adaptive Lasso) are used to achieve consistent variable selection under weaker conditions (sparse eigenvalue). Such two-step procedures involve choosing several tuning parameters that seems easy in principle, but difficult in practice. To address these problems efficiently, we propose a new two-step procedure, called *Post Lasso Stability Selection* (PLSS). At the first step, the Lasso screening is applied with a small regularization parameter to generate a candidate subset of active features. At the second step, Stability Selection using weighted Lasso is applied to recover the most stable features from the candidate subset. We show that under mild (generalized irrepresentable) condition, this approach yields a consistent variable selection method that is computationally fast even for a very large number of variables. Promising performance properties of the proposed PLSS technique are also demonstrated numerically using both simulated and real data examples.

1 INTRODUCTION

Due to the presence of high dimensional data in most areas of modern applications (examples include genomics and proteomics, financial data analysis, astronomy) variable selection methods gain considerable interest in statistical modeling and inference. In this paper, we consider variable selection problems in sparse linear regression models. We start with the standard linear regression model

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon, \tag{1}$$

where $\mathbf{Y}_{n\times 1}$ is a univariate response vector, $\mathbf{X}_{n\times p}$ is the design matrix, $\beta_{p\times 1}$ is the true underlying coefficient vector and $\varepsilon_{n\times 1}$ is an error vector. In particular, we consider sparse and high dimensional linear models, where the number of variables (p) is much larger than the number of observations (n), that is $p \gg n$. Sparsity assumption implies that only a few of the predictors contribute to the response. We denote the true active set or the support of β, by $S = supp(\beta)$. The goal is to estimate the true active set S from data $(\mathbf{Y}, \mathbf{X})$.

The Lasso (Tibshirani, 1996) has been a popular choice for simultaneous estimation and variable selection in sparse high dimensional problems. The Lasso penalizes least square regression by sum of the absolute value of the regression coefficients, the Lasso estimator is defined as

$$\hat{\beta}_{lasso} = \arg\min_{\beta\in\mathbb{R}^p} \left\{ \frac{1}{2n} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda\|\beta\|_1 \right\}, \tag{2}$$

where $\lambda \geq 0$ is the regularization parameter that controls the amount of regularization and the ℓ_1-penalty encourages the sparse solution. It has been proven that, under strong conditions (i.e., Irrepresentable Condition) on the design matrix $\mathbf{X}$, the Lasso correctly recovers the true active set S with high probability, for further details we refer to (Zhao and Yu, 2006), (Meinshausen and Bühlmann, 2006) and (Bühlmann and van de Geer, 2011). Sampling based procedures (i.e., Stability Selection and bootstrap Lasso) can be used as an alternative approach for variable selection, see (Meinshausen and Bühlmann, 2010) and (Bach, 2008). Though, the Stability Selection identifies the most stable features, but repeated sampling procedures make the algorithm very slow specially with the

Gauraha, N., Pavlenko, T. and Parui, S.
Post Lasso Stability Selection for High Dimensional Linear Models.
DOI: 10.5220/0006244306380646
In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2017)*, pages 638-646
ISBN: 978-989-758-222-6

large number of predictors. In practice, it is difficult to satisfy the Irrepresentability Condition, hence the Lasso does not provide any guarantees on the number of false discoveries. However, the Stability Selection has not been widely accepted due to its computational complexity. When the irrepresentable condition (IC) is violated, two-stage procedures (e.g. thresholding or adaptive Lasso) are used to achieve consistent variable selection. Such two-step procedures involve choosing several tuning parameters that further complicates the problem. We propose to combine the strength of the (adaptively weighted) Lasso and the Stability Selection for efficient and stable feature selection. In the first step, we apply the Lasso with small regularization parameter that selects a subset consisting of small number of features. In the second step, stability feature selection using weighted Lasso is applied to the restricted Lasso active set to select the most stable features. For the weighted ℓ_1 penalization, the weights are computed from the Lasso estimator at the first stage such that large effects covariates in the Lasso fit will be given smaller weights and small effects covariates will be given larger weights. We call the combination of the two, the Post-Lasso Stability Selection (PLSS).

Several authors have previously considered two stage Lasso-type procedures that have better potential and properties for variable selection than single stage Lasso, such as adaptive Lasso (Zhao and Yu, 2006), thresholded Lasso (Zou, 2006), relaxed Lasso (Meinshausen, 2007) and Gauss-Lasso (Javanmard and Montanari, 2013) to name a few. The Post-Lasso Stability Selection, is a special case of the two stage variable selection procedure: (1) Pre-selection stage: selection of predictors using the Lasso with small tuning parameter; and (2) Selection stage: selection of the most stable features from preselected predictors using Stability Selection with weighted Lasso. We prove that under assumption of Generalized Irrepresentability Condition (GIC) (Javanmard and Montanari, 2013), the initial Lasso active set with small tuning parameter contains the true active set S with high probability. Then stability feature selection where base selection procedure is the weighted Lasso, correctly identifies the stable predictors when applied on the restricted Lasso active set. The contribution of this paper is summarized as follows.

1. We briefly review two stage procedures for stable feature selection and estimation.
2. We propose a new combined approach, namely the Post Lasso Stability Selection (PLSS): The Lasso selecting initial active set and the Stability Selection using weighted Lasso selecting stable features from the initial active set.
3. We also utilize the estimation result obtained by the initial stage Lasso for computing weights of the selected predictors considered for the next stage.
4. We prove that under assumption of GIC, the PLSS correctly identifies the true active set with high probability.
5. We empirically show that PLSS outperforms the standard Lasso and the adaptive Lasso in terms of false positives.
6. We evaluate computational complexity of PLSS, and show that it is superior than the standard stability feature selection using the Lasso.

The rest of this paper is organized as follows. In Section 2, we provide background, notations, assumptions and a brief review of the relevant work. In section 3, we define and illustrate the Post Lasso Stability Selection. In section 4, we carry out simulation studies and we shall provide conclusion in section 5.

2 BACKGROUND AND NOTATIONS

In this section, we state notations, assumptions and definitions that will be used in later sections. We also provide a brief review of relevant work and our contribution.

2.1 Notations and Assumptions

We consider the sparse high dimensional linear regression set up as in (1), where $p \gg n$. We assume that the components of the noise vector ε are i.i.d. $\mathbb{N}(0,\sigma^2)$. The true active set or support of β is denoted as S and defined as $S = \{j \in \{1,...,p\} : \beta_j \neq 0\}$. We assume sparsity in β such that $s \ll n$, where $s = |S|$ is the sparsity index. The ℓ_1-norm and ℓ_2-norm (square) are defined as $\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$ and $\|\beta\|_2^2 = \sum_{j=1}^{p} \beta_j^2$ respectively. For a matrix $\mathbf{X} \in \mathcal{R}^{n\times p}$, we use superscripts for the columns of $\mathbf{X}$, i.e., $\mathbf{X}^j$ denotes the j^{th} column, and subscripts for the rows, i.e., $\mathbf{X}_i$ denotes the i^{th} row. For any $S \subseteq \{1,...,p\}$, we denote $\mathbf{X}^S$ as the restriction of $\mathbf{X}$ to columns in S, and β_S is the vector β restricted to the support S, with 0 outside the support S. Without loss of generality we can assume that the first $s = |S|$ variables are the active variables, and we partition the empirical covariance matrix, $C = \frac{1}{n}\mathbf{X}^T\mathbf{X}$, for the active and the redundant variables as follows.

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \tag{3}$$

Similarly, the true β is partitioned as $\begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}$.
The weighted Lasso estimator is defined as

$$\beta_{WL} = \arg\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{2n} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda \sum_{j=1}^{p} w_j |\beta_j| \right\}, \tag{4}$$

where $w \in \mathbb{R}^p$ is a known weights vector. We denote Λ as the set of considered regularization parameters defined as $\Lambda = \{\lambda : \lambda \in (0, \lambda_{max})\}$, where λ_{max} corresponds to the minimal value of λ for which the null model is selected. The following two conditions are assumed throughout the paper. (i)Beta-min condition, the non-zero entries of the true β must satisfy the condition $\beta_{min} \geq \frac{c\sigma}{\sqrt{n}}$, for some $c > 0$. (ii) Condition on the minimum number of observations, that is $n \geq s\,log(p)$.

2.2 The Lasso Variable Selection

The Least Absolute Shrinkage and Selection Operator (Lasso), is a penalized least squares method that imposes an ℓ_1-penalty on the regression coefficients. The Lasso does both shrinkage and automatic variable selection simultaneously due to nature of the ℓ_1-penalty. The Lasso estimated parameter vector denoted as $\hat{\beta}$ is defined in (2), and the Lasso estimated active set denoted by $\hat{S}_{lasso}$ can be given as

$$\hat{S}_{lasso} = \{j \in \{1, ..., p\} : \hat{\beta}_j \neq 0\}. \tag{5}$$

It is known that Irrepresentable condition is necessary and sufficient condition for the Lasso to select true model (see (Zhao and Yu, 2006)), the Irrepresentable condition is defined as follows.

Definition 1 (Irrepresentable Condition(IC)). *The Irrepresentable Condition is said to be met for the set S with a constant* $\eta > 0$*, if the following holds:*

$$\|C_{12}C_{11}^{-1} sign(\beta_1)\|_\infty \leq 1 - \eta. \tag{6}$$

In practice, IC on the design matrix $\mathbf{X}$, is quite difficult to meet. When IC fails to hold, the Lasso selected active set tends to have many false positive variables. A substantially weaker assumption than irrepresentability, called Generalized Irrepresentability Condition was introduced in (Javanmard and Montanari, 2013). They proved that, such a relaxation from irrepresentability condition to generalized irrepresentability condition allows to cover a significantly broader set of design matrices. In simple words, under generalized irrepresentability condition a little noise is allowed to get selected or the generalized irrepresentability condition can be viewed as irrepresentability condition satisfying for some superset of active set $T \supseteq S$. In (Javanmard and Montanari, 2013), authors also derived a suitable choice of λ_0, such that for the range $(0, \lambda_0)$ the Lasso selects the superset $T \supseteq S$ with high probability.

$$\lambda_0 = c\sigma\sqrt{\frac{2log(p)}{n}}, \text{ for some constant } c > 1. \tag{7}$$

2.3 Stability Variable Selection

In this section, we briefly study the stability feature selection method, which is mainly based on the concept that a feature is called stable if the probability of its getting selected is insensitive to variations in the training set. The Stability Selection, introduced by (Meinshausen and Bühlmann, 2010), is an effective method for performing variable selection in the high-dimensional setting while controlling the false positive rates. It is a combination of subsampling and high-dimensional feature selection algorithms (i.e., the Lasso). The Stability Selection can be expressed as a framework for the baseline feature selection method, to identify a set of stable predictors that are selected with high probability. The baseline feature selection method is repeatedly applied to random data sub-samples of half-size, and then the predictors which have selection frequency larger than a fixed threshold value (usually in the range $(0.6, 0.9)$) are selected as stable features.

Though the Lasso does not satisfy the oracle property and model selection consistency in high-dimensional data, but it has been proven that the Lasso selects the true active variables with high probability for more details we refer to (Meinshausen and Bühlmann, 2010). Hence, the Lasso method is commonly used as a base feature selection method for Stability Selection, we call it Stability Lasso. The active set of variables selected by Stability Lasso is given by

$$\hat{S}_{stab} = \{j \in \{1, ..., p\} : \hat{\Pi}_j \geq \pi_{thr}\}, \tag{8}$$

where $0 < \pi_{thr} < 1$ is a cut off probability. The variables with a high selection probability are selected as stable features. Here the parameter to be tuned is the exact cut off π_{thr}, the influence of π_{thr} is very small usually in the range $(0.6, 0.9)$). Tuning regularization parameter for the standard Lasso variable selection can be more challenging than for prediction, since the prediction optimal (i.e., cross-validated choice) often includes false positive selections. whereas, the stable active set does not depend much on the choice of the Lasso regularization λ, see (Meinshausen and Bühlmann, 2010) for more detailed discussion.

2.4 Review of Relevant work

In this section, we provide a brief review of relevant work in order to show that how our proposal differs from other two stage penalized least square methods for variable selection.

The Lasso variable selection could be inconsistent when IC fails to hold, to overcome this problem various two stage procedures have been introduced. The adaptive Lasso proposed in (Zou, 2006), uses adaptive weights for penalizing different coefficients in the ℓ_1-penalty as in weighted Lasso (4). The weights are chosen by an initial model fit, such that large effects covariates in the initial fit will be given smaller weights and small effects covariates will be given larger weights. Mostly, the Lasso is applied at the initial stage for high dimensional case, to derive the weights for the weighted Lasso at the second stage, together they are called the Adaptive Lasso. The adaptive Lasso estimator is defined as follows.

$$\hat{\beta}_{ada} = \arg\min_{\beta\in\mathbb{R}^p} \left\{ \frac{1}{2n}\|\mathbf{Y}-\mathbf{X}\beta\|_2^2 + \lambda * \lambda_{ada}\sum_j \frac{|\beta_j|}{|(\hat{\beta}_{lasso})_j|} \right\},$$

where $\hat{\beta}_{lasso}$ is computed using the standard Lasso method (2) as initial fit with initial regularization parameter $\lambda > 0$, and $\lambda_{ada} \geq 0$ is the regularization parameter for the second stage. Then the adaptive Lasso active set can be computed as

$$\hat{S}_{ada} = \{j \in \{1,...,p\} : (\hat{\beta}_{ada})_j \neq 0\}.$$

The thresholded Lasso was introduced in (Zhou, 2009), which further reduces the Lasso active set by eliminating the features having estimated coefficients below some pre-defined threshold value. More precisely, in the first stage the initial estimator is obtained using the Lasso with suitable regularization parameter λ, and then predictors were selected if their estimated coefficients are large enough (larger than the threshold value, say $\hat{\beta}_{thr}$).

$$\hat{S}_{thr} = \{j \in \{1,...,p\} : |(\hat{\beta}_{lasso})_j| \geq \hat{\beta}_{thr}\},$$

where $\hat{\beta}_{thr} > 0$, is the second tuning parameter for the thresholded Lasso.

The relaxed Lasso (Meinshausen, 2007) is another two step procedure, similar to adaptive or thresholded Lasso. The relaxed Lasso consists of two Lasso steps, in the first stage the Lasso variable selection is performed for a suitable grid of regularization parameters, say $(0, \lambda_{max})$ then at the second stage every submodel $\hat{S}_\lambda$ is considered and the Lasso with smaller regularization parameter is used on those sub models. The relaxed Lasso estimator is given as follows.

$$\beta_{\hat{S}}(\lambda,\phi) := \arg\min_{\beta_{\hat{S}}} \left\{ \frac{1}{2n}\|\mathbf{Y}-\mathbf{X}^{\hat{S}}\beta_{\hat{S}}\|_2^2 + \phi * \lambda\|\beta_{\hat{S}}\|_1 \right\},$$

where $\hat{S}(\lambda)$ is the estimated sub-model from the first stage.

The above two-stage procedures are proven to be variable selection consistent under some form of restricted and sparse eigenvalue conditions, see (van de Geer et al., 2011). But the tuning of regularization parameters are the main issue in practice. They require the two dimensional cross-validation (Hastie et al., 2001) to find the optimal pair of regularization parameter used at two different stages.

Next, we discuss about the Gauss-Lasso selector (see (Javanmard and Montanari, 2013)), which is also a two-stage method that first applies the Lasso, and then in the second stage it performs ordinary least squares restricted to the Lasso active set $\hat{S}_{lasso}$.

$$\hat{\beta}_{GL} = \arg\min_{\beta} \left\{ \frac{1}{2n}\|\mathbf{Y}-\mathbf{X}^{\hat{S}_{lasso}}\beta_{\hat{S}_{lasso}}\|_2^2 \right\}$$

Given the sparsity index or number of non-zero coefficients s_0, the Gauss-Lasso selector then finds the s_0^{th} largest entry (in absolute) of $\hat{\beta}_{GL}$, denoted by $\hat{\beta}_{s_0}$. Then finally GL active set is given by

$$\hat{S}_{GL} = \{j \in \{1,...,p\} : |(\hat{\beta}_{GL})_j| \geq \hat{\beta}_{s_0}\}$$

Though, the Gauss-Lasso model selects the correct model with high probability under GIC which is weaker than the IC. But, it demands the true sparsity index $s_0 = |S|$ for selecting s_0 relevant features, but in practice, the size of active set is not known.

Finally, we mention about the bootstrap Lasso (Bolasso) which is more close to the Stability Selection. In Bolasso, the Lasso is applied for several bootstrapped replications of a given sample, then final active set is given by intersection of the supports of the Lasso bootstrap estimates. Bolasso is a consistent variable selection method, that does not assume any condition on the design matrix $\mathbf{X}$. The Bolasso is not a preferable choice since it is computationally expensive, as repeatedly applying Lasso on bootstrap samples specially with large number of predictors makes it slow.

We propose the Post Lasso Stability Selection as a computationally fast alternative, which does not need to be tuned across the two-dimensional grid of tuning parameters. It is a simple and consistent method for variable selection even when the irrepresentable condition is violated. We define and discuss PLSS in the next section.

3 POST LASSO STABILITY SELECTION

In this section, we introduce a new combined two stage approach, called Post Lasso Stability Selection.

The first stage involves selecting a super set of the true active set, using the Lasso with a small regularization parameter λ_0 (7). Then in the second stage, Stability Selection using the weighted Lasso is applied on the Lasso restricted set obtained at the first stage. We also compute weights from the Lasso estimator from the first stage to assign different weights to different coefficients. On the one hand, the Lasso at the first stage makes sure the true active set gets selected (along with some noise) under assumption of the generalized irrepresentability condition on the design matrix **X**. On the other hand, the Stability Selection using weighted Lasso at second stage makes sure that the most stable predictors finally get selected and the noise features are eliminated from the final model. aa

Algorithm 1: PLSS Algorithm.

```
Input: dataset $(\mathbf{Y}, \mathbf{X}, \pi_{thr})$
Output: $\hat{S}$:= set of selected variables
Steps: 1. Perform Lasso with
$\lambda_0 = \sqrt{2log(p)/n}$. Denote the Lasso estimator
as $\hat{\beta}_{lasso}$ and the Lasso active set as $\hat{S}_{lasso}$
2. Compute weights as $w_j = |(\hat{\beta}_{lasso})_j|$.
3. Compute the weighted reduced design
matrix, $\mathbf{X}_{red} = \{w_j * \mathbf{X}^j : j \in \hat{S}_{lasso}\}$.
4. Perform stability feature selection based on
data $(\mathbf{Y}, \mathbf{X}_{red})$ and obtain the estimated
probabilities $\pi_j$ for all $j \in \hat{S}_{lasso}$.
Determine the selected active set as

    $\hat{S} = \{j \in \hat{S}_{lasso} : \hat{\pi}_j \geq \pi_{thr}\}$

return $\hat{S}$
```

3.1 Consistency of PLSS

We assume that the GIC holds on the design matrix **X** for some $T \subseteq \{1, ..., p\}$, and T contains the true active set S, i.e. $T \supseteq S$. Without loss of generality we can assume that the GIC corresponds to the first $t = |T|$ predictors. In the first stage, then the Lasso active set contains the true active set with high probability under GIC assumption. In the second stage, for Stability Selection with adaptively weighted Lasso, the bounds on maximal and minimal eigenvalues are required. As GIC holds for the set $T \supseteq S$, therefore the covariance matrix $C(T)$ is invertible and uncorrelated with the noise features that implies the minimum and maximum eigenvalue of sub-matrices of C of size $t \times t$ are bounded away from 0 and ∞ respectively. Hence, under GIC assumption the PLSS method is variable selection consistent.

3.2 Computation Complexity for PLSS

In this section, we discuss the computational complexity of the PLSS. The computation steps are given in Algorithm (1). Since, the PLSS performs the Lasso and Stability Selection in two different stages, therefore we use the results from those studies. The LARS (Efron et al., 2004) algorithm is used to compute the Lasso, the computational cost of LARS is of order $O(np^2)$. Computation cost of Stability Selection using Lasso as a base feature (with 100 sub samples), is approximately $O(25np^2)$, where the constant 25 is due to running 100 simulations on the on sub samples of size $\frac{n}{2}$, we refer to (Meinshausen and Bühlmann, 2010) for more details on the derivation of the result. So, the computational cost of the PLSS for its different stages can be given as:
stage one: $O(np^2)$
stage two: $O(25ns_1^2)$, where we assume s_1 is the size of the Lasso active set and in practice $s_1 \ll p$. Hence, the computation cost of the PLSS is of order $O(np^2 + 25ns_1^2)$.

3.3 Illustration of PLSS

To illustrate the PLSS, we consider a small simulation example with the following setup.

Data simulation setup

- $p = 1000$, $n = 200$ and $\sigma = 1$.
- The design matrix **X** is sampled from a multivariate normal $\mathbb{N}_p(0, \Sigma)$, where Σ is the identity matrix except the left most 5×5 sub matrix, which is defined as follows.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \rho \\ 0 & 1 & 0 & 0 & \rho \\ 0 & 0 & 1 & 0 & \rho \\ 0 & 0 & 0 & 1 & \rho \\ \rho & \rho & \rho & \rho & 1 \end{bmatrix}$$

- The active set is defined as $S = \{1, 2, 3, 4\}$, and for each $j \in S$ we set $\beta_j = 1$.

In the above setting, the fifth variable is equally correlated with all four active predictors. Using the above setup, we run the following simulation steps to perform variable selection using Lasso, stability selection and PLSS.

Simulation steps

1. Construct the design matrix $\mathbf{X}$ with $\rho = 0.25$.
2. Generate an error vector as $\varepsilon_{n\times 1} \sim \mathbb{N}_n(0, I)$ and then compute the response using Eq. (1).
3. Compute the Lasso estimator using the simulated data set $(\mathbf{Y}, \mathbf{X})$ (choose λ using cross validation) and obtain the Lasso active set.
4. Perform Stability Selection on the data set $(\mathbf{Y}, \mathbf{X})$ and obtain the stability path.
5. Perform PLSS (defined later) on the data set $(\mathbf{Y}, \mathbf{X})$, Then compute the stability active set and obtain the stability path.

In the following, the results for the above simulation are presented. We remark that, for $\rho \geq 0.25$ the IC is violated by the design matrix $\mathbf{X}$. As a result, the Lasso always selects the fifth predictor with the first four relevant predictors and with some other noise feature as reported by the Lasso active set $\hat{S}_{lasso}$ is $\{1,2,3,4,5,239,265,326,374,469,531,747,794,865,942\}$. When applying the Stability Selection using weighted Lasso, the first four important predictors are selected with their estimated probabilities close to 1, while the irrelevant variables are selected with much lower probability, see Figure (2) for probabilities of features getting selected.

We also compare the stability paths of the standard Stability Selection (using Lasso) with the PLSS for the above example. For each predictor $j = \{1, ..., p\}$, the stability path is given by the selection probabilities $\{\hat{\Pi}_j(\lambda) : j = \{1, ..., p\}, \lambda \in (0, \lambda_{max})\}$. From Figures (1) and (2) (the four important predictors are plotted as red lines, while the paths of noise features are shown as black lines) , we see that the stability path of the PLSS is much cleaner or it can be interpreted as, a lot of computational effort is saved in PLSS as most of the noise features are getting filtered at the first stage itself.

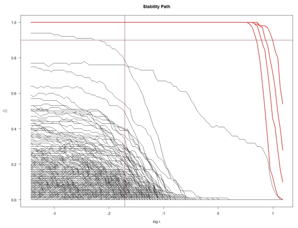

Figure 1: Stability Path of the Stability Lasso.

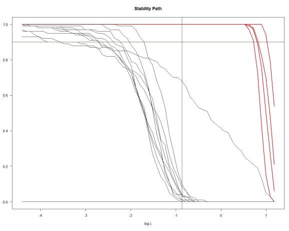

Figure 2: Stability Path of the PLSS.

4 NUMERICAL RESULTS

In this section, we consider simulation settings and pseudo-real data examples to compare the performances of Lasso, Adaptive Lasso, Stability Selection and PLSS in terms of variable selection. In particular, we consider the true positive rate and the false discovery rates as a measure of performances, which are defined as follows.

$$TPR = |\hat{S}\bigcap S|/|S|, \text{ and } FDR \quad = |\hat{S}\bigcap S^c|/|\hat{S}| \qquad (9)$$

The Statistical analysis is performed in $R3.2.5$. We used, the packages “glmnet” for penalized regression methods (Lasso, Adaptive Lasso) and the package “c060” to perform stability feature selection using Lasso. All mentioned packages are available from the Comprehensive R Archive Network (CRAN) at http://cran.r-project.org/.

4.1 Example 1: Simulation

In order to compare the computational cost of the Stability Selection and the PLSS, we simulated a dataset with the following details.

Simulation setup for Example 1

- Fix $n = 500$, and $p = 10000, 20000, 30000, 40000, 50000, 100000$.
- Set $\sigma = 1$, and the design matrix $\mathbf{X}$ is sampled from a multivariate normal $\mathbb{N}_p(0, I)$.
- The active set is defined as $S = \{1, ..., 20\}$, and for each $j \in S$ we set $\beta_j = 1$.

The time complexity for both Stability Selection using the Lasso (using 100 sub samples) and PLSS on a super computer, are reported in the Table (1).

Table 1: Measure of time complexity (in seconds).

p	Stability Selection	PLSS
10000	29.070	1.989
20000	53.613	4.654
30000	59.259	5.527
40000	99.986	6.273
50000	202.103	8.762
100000	459.341	28.418

4.2 Example 2: Simulation

We use the following simulation setup for generating data $(\mathbf{Y},\mathbf{X})$.

Simulation setup for Example 2

- Set $p = 1000$, $\sigma = 3$, and $n = 100,\ 200,\ 400$.
- Generate the design matrix $\mathbf{X}$ from $\mathbb{N}_p(0,\Sigma)$, here we consider two different settings for $\Sigma = \{\Sigma_1,\ \Sigma_2\}$, where $\Sigma_1 = I_p$ and

$$\Sigma_2(i,j) = \begin{cases} (0.5)^{|i-j|} & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases},$$

- The active set is defined as $S = \{1,...,20\}$, and for each $j \in S$ we set $\beta_j = 1$.

The performance measures for simulations with Σ_1 and Σ_2 are reported in Table (2).

Table 2: performance measures for example 2.

n	Method	Σ_1		Σ_2	
		TPR	FDR	TPR	FDR
100	Lasso	0.75	0.55	.55	0.71
	Ada Lasso	0.75	0.51	.55	0.67
	Stab Lasso	0.05	0	0	0
	PLSS	0.5	0	.4	0.27
200	Lasso	1	0.65	1	0.66
	Ada Lasso	1	0.58	1	0.50
	Stab Lasso	0.6	0	0.65	0
	PLSS	1	0	1	0
400	Lasso	1	0.54	1	0.53
	Ada Lasso	1	0.35	1	0.25
	Stab Lasso	1	0	1	0
	PLSS	1	0	1	0

4.3 Example 3: Riboflavin Data

We consider Riboflavin data (see (Bühlmann et al., 2014)) for the design matrix $\mathbf{X}$ with synthetic parameters β and simulated Gaussian errors $\varepsilon \sim \mathbb{N}_n(0,I)$. To fulfil the minimum sample size condition ($n \geq s\,log(p)$) we reduce the dimension to $p = 1000$, and the pseudo data generation steps are given as follows.

Data generation for Riboflavin example

- For the design matrix $\mathbf{X}$, select first 1000 covariates which are most associated with the response.
- Fix $s = 10$ and for the true active set, sample ten numbers randomly from the set $\{1,...,50\}$, and for each $j \in S$ we set $\beta_j = 1$.
- Compute the response using the Equation (1).

The performance measures are reported in Table (3), and Figures (3) and (4).

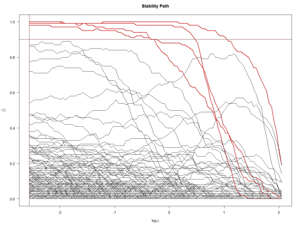

Figure 3: Stability Path of the Stability Lasso for Riboflavin.

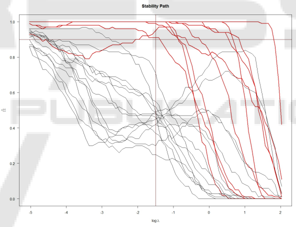

Figure 4: Stability Path of the PLSS for Riboflavin.

Table 3: Performance measures for Riboflavin example.

Method	TPR	FDR
Lasso	0.9	0.57
Ada Lasso	0.9	0.21
Stab Lasso	0.4	0
PLSS	0.9	0

4.4 Example 4: Myeloma Data

Here, we consider the first 1000 highest variance genes of the real dataset Myeloma (see (Tian et al., 2003)) for the design matrix $\mathbf{X}$ with synthetic parameters β and simulated Gaussian errors. The pseudo data generation steps are similar as the previous example (Riboflavine). The performance measures are reported in Table (4), and Figures (5) and (6).

Table 4: Performance measures for Myeloma example.

Method	TPR	FDR
Lasso	1	0.63
Ada Lasso	1	0.16
Stab Lasso	0.8	0
PLSS	1	0

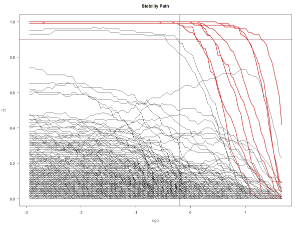

Figure 5: Stability Path of the Stability Lasso for Myeloma.

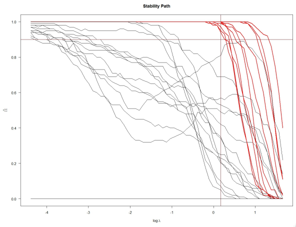

Figure 6: Stability Path of the PLSS for Myeloma.

4.5 Empirical Results

It is evident from the results of the simulation and pseudo real examples that the PLSS method outperforms others, the number of false positives selected by the Lasso and the adaptive Lasso is much larger than the PLSS (except when the requirement of the minimum number of observations is not met, for $n = 100$ case in Example 2). The PLSS performs better than Stability Selection, the Stability Selection misses the true predictors when the sample size is small, for $n = 200$ in Table (2), and for real data case see Tables (3) and (4), and Figures (3) and (4). From Example 1, it is apparent that the PLSS outperforms the Stability Selection in terms of computation complexity.

5 CONCLUSIONS

In this article, we have proposed a two stage variable selection procedure, Post-Lasso Stability Selection with controlled false positives. At the first stage, the Lasso is performed with a small regularization parameter to obtain initial estimator, where small value of regularization parameter and Generalized Irrepresentable Condition on the design matrix $\mathbf{X}$, ensures that the Lasso active set contains the true active set S. At the second stage, Stability Selection using weighted Lasso is performed on the restricted set, where the weights are computed from the initial Lasso estimator. We have shown that the PLSS combines the strength of the Lasso and the Stability Selection. We illustrated the method using simulated and real data examples and our empirical results have shown that the PLSS compares favorably with other two stage variable selection techniques. We have also proved that under GIC assumption on the design matrix, the PLSS has substantially less false positives than the Lasso and potentially faster than the Stability Selection.

ACKNOWLEDGEMENTS

The authors acknowledge the PDC Center for High Performance Computing at the KTH Royal Institute of Technology for providing computational resources.

REFERENCES

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

Bühlmann, P., Kalisch, M., and Meier, L. (2014). High-dimensional statistics with a view towards applications in biology. *Annual Review of Statistics and its Applications*, 1:255–278.

Bühlmann, P. and van de Geer, S. (2011). *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Verlag.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Ann. Statist.*, 32(2):407–499.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning; Data Mining, Inference and Prediction*. New York: Springer.

Javanmard, A. and Montanari, A. (2013). Model selection for high-dimensional regression under the generalized irrepresentability condition. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, pages 3012–3020.

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

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34:1436–1462.

Meinshausen, N. and Bühlmann, P. (2010). Stability selection (with discussion). *J. R. Statist. Soc*, 72:417–473.

Tian, E., Zhan, F., Walker, R., Rasmussen, E., Ma, Y., Barlogie, B., and Shaughnessy, J. J. (2003). The role of the wnt-signaling antagonist dkk1 in the development of osteolytic lesions in multiple myeloma. *N Engl J Med.*, 349(26):2483–2494.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Statist. Soc*, 58:267–288.

van de Geer, S., Bhlmann, P., and Zhou, S. (2011). The adaptive and the thresholded lasso for potentially misspecified models (and a lower bound for the lasso). *Electron. J. Statist.*, 5:688–749.

Zhao, P. and Yu, B. (2006). On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2541–2563.

Zhou, S. (2009). Thresholded lasso for high dimensional variable selection and statistical estimation. *NIPS*, pages 2304–2312.

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