

Summary and discussion of: “Stability Selection”

Statistics Journal Club, 36-825

Shiqiong Huang and Micol Marchetti-Bowick

1 Introduction

The estimation of model structure from data is an important statistical problem known as *structure learning*. The paper that we discussed, written by Meinshausen and Bühlmann, introduces a new method called *stability selection* whose goal is to provide an algorithm for performing model selection in a structure learning problem while controlling the number of false discoveries. This algorithm can be applied to a number of different structure learning problems, and has two main advantages over competing approaches:

1. It works in the high-dimensional data setting ($p \gg n$), which is currently a very active area of research in statistics and machine learning.
2. It provides control on the family-wise error rate in the finite sample setting, which is more practical than an asymptotic guarantee.

1.1 Problem Setting

Before presenting details of the stability selection algorithm, it’s useful to discuss a few types of problems that can roughly be called “structure estimation” problems, whose goal is to learn discrete structures from a dataset.

Variable selection. In a penalized regression problem such as the lasso, the goal is to identify a set of variables that will have nonzero weight in the model. In this case, we estimate the model parameters $\hat{\beta}$ and then define the selection set \hat{S} as follows:

$$\begin{aligned}\hat{\beta}^\lambda &= \arg \min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \\ \hat{S}^\lambda &= \{k : \hat{\beta}_k^\lambda \neq 0\}\end{aligned}$$

where $X \in \mathbb{R}^{n \times p}$ is the design matrix, $y \in \mathbb{R}^n$ is a vector of outputs, and λ is a regularization parameter that controls the size of the selection set.

Network estimation. Network estimation is often formalized as an inverse covariance estimation problem in a Gaussian graphical model. Assuming we have data drawn from a multivariate Gaussian distribution, i.e. $(X_1, \dots, X_p) \sim \mathcal{N}(0, \Sigma)$, we can estimate the pairwise conditional dependencies (which correspond to edges in the network) among the

variables $1, \dots, p$ (which correspond to nodes in the network) by estimating the inverse covariance matrix $\Theta = \Sigma^{-1}$ and then identifying the entries with nonzero value. In order to identify some conditional independencies, we must obtain a sparse estimate of the covariance matrix. In this case, we estimate the model parameters $\hat{\Theta}$ and then define the selection set \hat{S} as follows:

$$\begin{aligned}\hat{\Theta}^\lambda &= \arg \min_{\Theta \in \mathbb{R}^{p \times p}, \Theta \succeq 0} -\log \det(\Theta) + \text{tr}(S\Theta) + \lambda \|\Theta\|_1 \\ \hat{S}^\lambda &= \{(j, k) : \hat{\theta}_{j,k}^\lambda \neq 0\}\end{aligned}$$

where $S = \frac{1}{n}X^TX \in \mathbb{R}^{p \times p}$ is the empirical covariance matrix and λ is a regularization parameter that controls the size of the selection set.

Clustering. In a clustering problem such as k-means, the goal is to group a set of data points into k distinct clusters. Here k is a parameter that dictates the number of clusters. We don't provide a formal definition of this problem because the authors do not provide a way to apply stability selection to clustering, but it serves as an additional example of a structure estimation problem.

We note that in the first two problems described above, and in many other related structure estimation problems, performing "model selection" is equivalent to choosing the appropriate amount of regularization (i.e. the value of λ) such that \hat{S} is as close as possible to the true selection set S .

1.2 Existing Methods

In order to put stability selection in context, we first describe several existing methods for performing model selection.

Cross-validation. A method for choosing the value of λ that minimizes the generalization error on a held-out test validation set. This approach only works for supervised problems such as regression. It also doesn't provide any guarantees on the number of false discoveries and doesn't work well in the high-dimensional setting.

AIC/BIC scores. A method for model selection that trades off goodness of fit with model complexity. This approach tends to perform poorly in the high-dimensional setting.

Hypothesis testing methods. A newer class of methods based on hypothesis testing that can be used to perform variable selection in certain settings while controlling the false discovery rate.

Knockoffs. A method for variable selection that provides false discovery rate control in the finite sample setting. This approach only works for $p < n$.

Stability methods. A class of methods for identifying the most “stable” model structure by using the idea that the same algorithm should yield similar results on similar datasets if the results are “stable”.

2 Method

Let’s assume that we have a generic structure estimation algorithm that takes a dataset $\mathbf{Z} = Z_1, \dots, Z_n$ and a regularization parameter λ and returns a selection set \hat{S}^λ . We can think of this algorithm as a black box. The stability selection algorithm then runs as follows:

1. Define a candidate set of regularization parameters Λ and a subsample number N .
2. For each value of $\lambda \in \Lambda$, do:
 - (a) Start with the full dataset $\mathbf{Z}_{(\text{full})} = Z_1, \dots, Z_n$
 - (b) For each i in $1, \dots, N$, do:
 - i. Subsample from $\mathbf{Z}_{(\text{full})}$ without replacement to generate a smaller dataset of size $\lfloor n/2 \rfloor$, given by $\mathbf{Z}_{(i)}$.
 - ii. Run the selection algorithm on dataset $\mathbf{Z}_{(i)}$ with parameter λ to obtain a selection set $\hat{S}_{(i)}^\lambda$.
 - (c) Given the selection sets from each subsample, calculate the empirical selection probability for each model component:

$$\hat{\Pi}_k^\lambda = \mathbb{P}\{k \in \hat{S}^\lambda\} = \frac{1}{N} \sum_{i=1}^N \mathbb{I}\{k \in \hat{S}_i^\lambda\}.$$

The selection probability for component k is its probability of being selected by the algorithm.

3. Given the selection probabilities for each component and for each value of λ , construct the stable set according to the following definition:

$$\hat{S}^{\text{stable}} = \{k : \max_{\lambda \in \Lambda} \hat{\Pi}_k^\lambda \geq \pi_{\text{thr}}\}$$

where π_{thr} is a predefined threshold.

Note that this procedure doesn’t simply find the best value of $\lambda \in \Lambda$ and then use it in the algorithm, but actually identifies a set of “stable” variables that are selected with high probability. The authors state that the empirical results vary little for threshold values in the range $(0.6, 0.9)$, and are also not sensitive to choices of Λ .

3 Theory

Before we present the main theorem of this paper, we first introduce some additional notations. Let S be the true variables and N be the noise variables such that $S \cup N = \{1, 2, \dots, p\}$. Furthermore, let $\hat{S}^\Lambda = \cup_{\lambda \in \Lambda} \hat{S}^\lambda$ be the set of selected structures or variables if varying the regularization λ in the set Λ . Let q_Λ be the average number of selected variables, $q_\Lambda = E(|\hat{S}^\Lambda(I)|)$. Define V to be the number of falsely selected variables with stability selection, that is $V = |N \cap \hat{S}^{\text{stable}}|$. Then we have the following theorem to control the expected number of falsely selected variables,

Theorem 1. Assume that the distribution of $\{1_{\{k \in \hat{S}^\lambda\}}, k \in N\}$ is exchangeable for all $\lambda \in \Lambda$. Also, assume that the original procedure is not worse than random guessing, i.e.

$$\frac{E(|S \cap \hat{S}^\Lambda|)}{E(|N \cap \hat{S}^\Lambda|)} \geq \frac{|S|}{|N|}.$$

The expected number V of falsely selected variables is then bounded for $\pi_{\text{thr}} \in (\frac{1}{2}, 1)$ by

$$E(V) \leq \frac{1}{2\pi_{\text{thr}} - 1} \frac{q_\Lambda^2}{p}.$$

The key idea in proving this theorem is that the variables unlikely to be selected using all the samples would also have small chance of being selected by half sampling. To put it strictly, let I_1 and I_2 be two random subsets of $\{1, 2, \dots, n\}$ with $|I_i| = \lfloor n/2 \rfloor$ for $i = 1, 2$ and $I_1 \cap I_2 = \emptyset$. Define the simultaneously selected set as

$$\hat{S}^{\text{simult}, \lambda} = \hat{S}^\lambda(I_1) \cap \hat{S}^\lambda(I_2)$$

and define

$$\hat{\Pi}_K^{\text{simult}, \lambda} = P(K \subseteq \hat{S}^{\text{simult}, \lambda}).$$

For any $K \subset \{1, 2, \dots, p\}$, if $P(K \subseteq \hat{S}^\lambda) \leq \epsilon$, then

$$P(\max_{\lambda \in \Lambda} (\hat{\Pi}_K^{\text{simult}, \lambda}) \geq \xi) \leq \epsilon^2 / \xi.$$

The author mentioned that the expected number of falsely selected variables is called the per-family error rate or, if divided by p , the per-comparison error rate in multiple testing. To control the per-comparison error rate, one can either control π_{thr} or q_Λ . They also said that the influence of π_{thr} is very small and for practice, a sensible range would be $\pi_{\text{thr}} \in (0.6, 0.9)$. Once the threshold has been chosen at some default value, the regularization region Λ is determined by the error control desired. Specifically, for a default cut-off value $\pi_{\text{thr}} = 0.9$, choosing the regularization parameters Λ such that say $q_\Lambda = \sqrt{0.8p}$ will control $E(V) \leq 1$, or choosing Λ such that $q_\Lambda = \sqrt{0.8\alpha p}$ controls the familywise error rate at level α , i.e. $P(V > 0) = P(V \geq 1) \leq E(V) \leq \alpha$.

However, our simulations indicate that it is not always the case as the authors claimed. The choice of π_{thr} is actually critical, and in some cases we even need to choose $\pi_{\text{thr}} < 0.5$ or $\pi_{\text{thr}} > 0.9$. Choosing q_Λ is also not a trivial task since there is no intuitive formula for the selection of Λ given q_Λ . We will give more detailed discussions in section 4 and section 5.

4 Simulations

4.1 Comparison of Stability Selection and Cross-Validation

In order to understand whether stability selection is an effective method for performing variable selection in the high-dimensional setting, we ran some experiments on simulated data. We first generated p predictor variables by drawing each element uniformly at random to create a design matrix $X \in \mathbb{R}^{n \times p}$. We then generated a coefficient vector $\beta \in \mathbb{R}^p$ by setting the first 5 values to either 2 or -2 . Finally, we generated the output variable according to $y \sim \mathcal{N}(X\beta, \sigma^2)$ where σ^2 is the variance of the noise.

In order to compare these methods in a variety of conditions, we fixed the sample size at $n = 50$ and varied both the noise variance and the dimensionality over $\sigma^2 \in \{1, 9\}$ and $p \in \{500, 5000\}$. The regularization paths and stability paths for the data generated in each of these settings are shown in Figure 1. We then ran both stability selection with threshold $\pi_{\text{thr}} = 0.6$ and 10-fold cross-validation. The cutoff selection probability for stability selection is shown by the horizontal green dashed line on the stability path. The optimal value of lambda for cross-validation is shown by the vertical blue dashed line on the regularization path. We also use the vertical purple dashed line to indicate the largest value of lambda that yields a generalization error within one standard deviation of the minimum for cross-validation.

These results show that stability selection generally has a lower FDR than cross-validation, but also achieves a lower power. However, we observe that the overall ranking of variables appears to be fairly similar between the two methods. As a caveat, we note that we only tested a limited set of conditions and that each result shown in Figure 1 is based only on one randomly generated dataset.

4.2 Stability Selection in Low-dimensional Case

Knockoffs method deals with the “ $p < n$ ” case, so in this simulation, we will be looking at the low dimensional simulated data to compare stability selection and knockoffs. The data generation method is the same as before, but the parameters we choose to use is different. We use $n = 500$ and $p = 50$. And the first 25 variables are true variables, which makes it half of the total variables. $\sigma^2 = 9$ is chosen to be the variance. Similar results of lasso and stability selection path for one randomly generated data are shown in Figure 2. We see from the figure that the behaviors of stability selection is quite different for low dimensional case. In order to achieve a reasonable FDR, π_{thr} is chosen to be 0.98. In fact, any $\pi_{\text{thr}} \leq 0.9$ would result to choosing all the variables, causing FDR to be 0.5. This contradicts the authors’ suggestion on $\pi_{\text{thr}} \in (0.6, 0.9)$.

The power of both methods is 1, but stability selection outperforms cross-validation in FDR with stability selection being 0.24 and lasso path being 0.36. We also computed the power and FDR using knockoffs. The FDR for knockoffs is 0.17 (with FDR controlled at level 0.3) and the power is 0.8. The FDR of knockoffs is lower than stability selection with the sacrifice of the power. Overall, there does not seem to be any obvious advantages of stability selection over knockoffs. The choice of which method to use depends on whether one focus more on FDR or power.

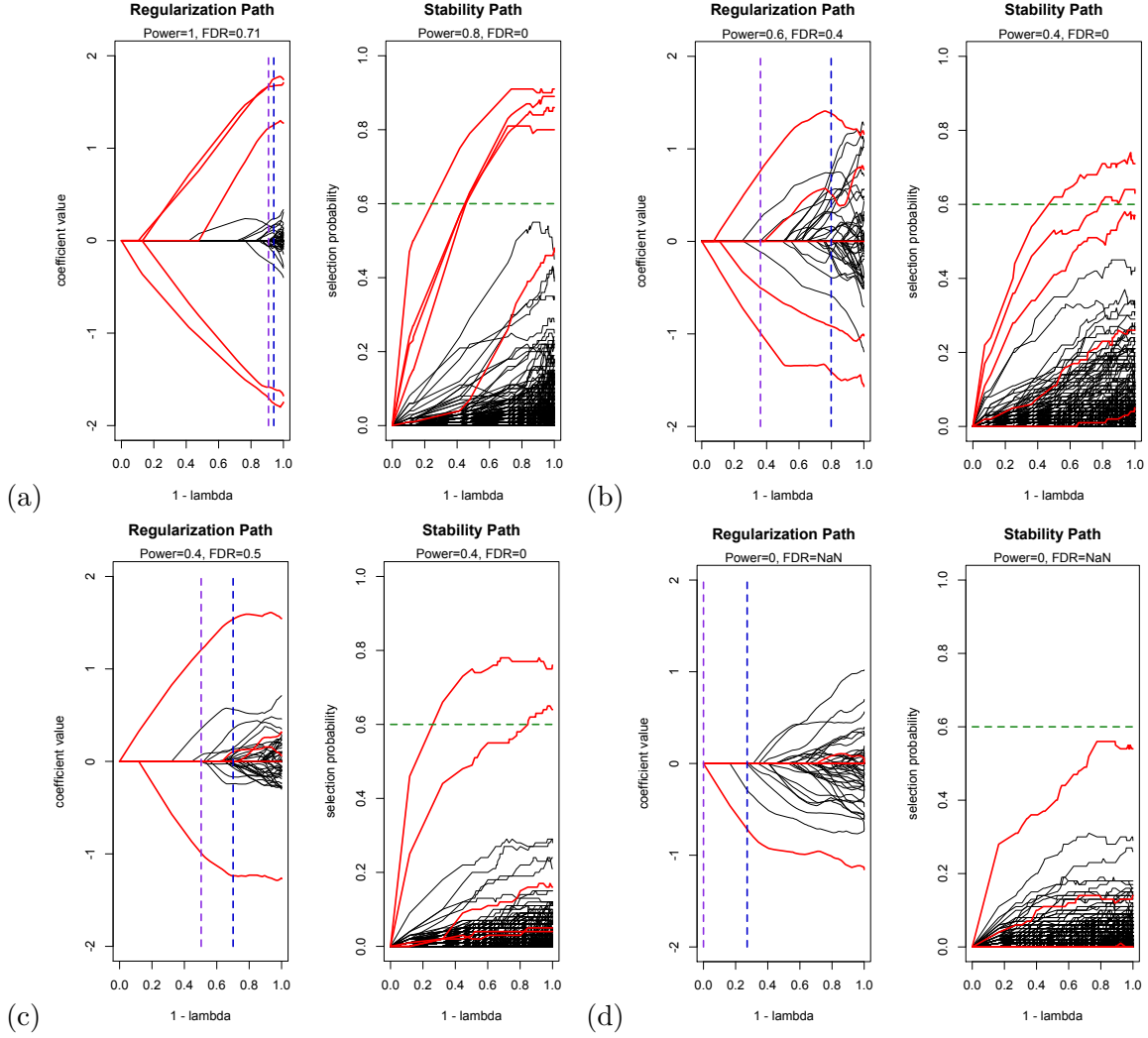


Figure 1: A comparison of stability selection and cross-validation for variable selection. Red lines show active variables, black lines show noise variables. All datasets are generated with $n = 50$. (a) Data generated with $\sigma^2 = 1, p = 500$. (b) Data generated with $\sigma^2 = 9, p = 500$. (c) Data generated with $\sigma^2 = 1, p = 5000$. (d) Data generated with $\sigma^2 = 9, p = 5000$.

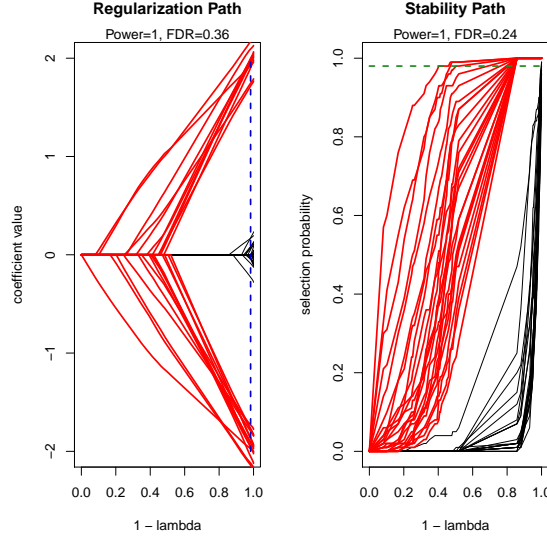


Figure 2: Lasso and stability selection path for $n = 500, p = 50, \sigma^2 = 9$. The left panel is lasso and the right panel is stability selection.

5 Discussion

The biggest contribution of this paper comes from the practical use instead of theoretical support. In fact, we found the theorem presented in this paper with several limitations in both conditions and results. Firstly, the exchangeability condition of $\{1_{\{k \in \hat{S}^\lambda\}}, k \in N\}$ is rather strong and by no means can be verified. However, this condition is required in terms of theoretical proof and the we can still perform the procedure without confirming whether this assumption is fulfilled.

One more important issue is about the result of controlling falsely selected variables. In most cases, we do not need a strict control on exact how many errors do we make in total. In practice, false discovery rate (FDR), which computes the proportion of falsely selected variables, is of more interest to statisticians. The FDR under the notations of this paper would be $E(\frac{V}{|\hat{S}^{\text{stable}}|})$. Usually, we can substitute it with $\frac{E(V)}{E(|\hat{S}^{\text{stable}}|)}$. Stability selections fails to provide any results on \hat{S}^{stable} . Therefore, we can not obtain an control on either FDR or power.

On the other hand, stability selection requires sampling the data enough times and each time using half of the samples. The repeated sampling procedures make the algorithm slow in terms of computation time. Besides, using only half of the samples could end up losing a lot of information, especially for the data that the samples are limited and every sample actually matters. These might be the reason why stability selection has not been widely adopted by statisticians and people still prefer to use cross-validation or other methods to perform model selection.