Stability Selection

Peerapat Phatpanichot

1005780558

STA315 Final Paper

Table of Content

Introduction 3

Problem setting 4

Method 5

Theory 7

Conditions 9

Simulation 10

Discussion 11

References 12

Introduction

In the contemporary field of data science, building predictive models using high-dimensional datasets is increasingly common. These datasets often contain many more features than samples, which makes it challenging to create predictive models that do not encounter overfitting. This is where the stability of feature selection becomes crucial, as the term "stability" refers to the robustness of the feature selection process with respect to data sampling. Consequently, the measurement of stability is significant to our study since it addresses the reliability of our algorithm.

Generally, feature selection can yield three types of outputs: weighting, ranking, and subsets of features to indicate the relevance of features to the target variable. In the context of this paper, we focus on stability selection with weighting, which assigns significance to each feature concerning the target variable. Several factors can contribute to instability in feature selection algorithms, including dimensionality, the number of selected features, sample size, and feature redundancy. In modern data science, numerous techniques can address this instability, such as ensemble feature selection, data variance reduction, and heuristic-based approaches. For the purpose of this paper, the stability selection method can be considered a branch of ensemble feature selection.

Ensemble feature selection comprises two main components: a randomization part that introduces diversity in the feature selection outputs through data perturbation, and an aggregation part that combines the outputs. A prominent technique exemplifying ensemble feature selection is the Stability Selection process introduced by Nicolai Meinshausen and Peter Bühlmann. Their work presents an aggregation technique based on the feature sets obtained using a variety of regularizing parameters in LASSO.

Problem setting

In the context of the stability selection algorithm, the structure estimation problems primarily arise because the paper mainly utilizes LASSO as a base feature selection algorithm.

Consider a linear model, a standard linear model represented as:

$$Y_i = B_0 + \sum_{\{j=1\}}^{p} B_j X_i^j + \epsilon_i, \quad i = 1, .., n \ll p$$

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

$$\widehat{\beta}^{\,\lambda} = argmin_{\{\beta\}} \left(n^{-1} \parallel Y - X\beta \parallel^2 + \lambda \parallel \beta \parallel_1 \right)$$

$$\hat{S}^{\lambda} = \left\{ j:, \hat{\beta}_j^{\lambda} \neq 0 \right\}$$

Here, S represents the set of active variables – the covariates with corresponding coefficients different from 0. $\hat{S}^{\lambda}$ is the feature selection procedure, as it selects only the subset among our p covariates, with subset indices ranging from 1 to p, and λ is our tuning parameter.

In the paper, the primary $\hat{S}^{\lambda}$ used by the authors is LASSO, where $\hat{\beta}^{\lambda}$ is a LASSO estimator. Minimizing the residual sum of squares, penalized by L1, reduces some of the coefficients to 0. This process exhibits a selection property, and $\hat{S}^{\lambda}$ comprises only the covariates with estimated regression coefficients different from 0. This problem setting is a standard optimization problem.

Method:

As previously mentioned in the problem setting step, we have an estimation algorithm that takes a dataset $X = X_1, \ldots., X_n$ and a regularization parameter λ, which outputs a selection set of $\hat{S}^{\lambda}$. The stability selection process can be modeled through the following steps:

1. Frist, we define a candidate set of regularization parameter Λ and a subsample of number n, where Λ is a parameter that controls the strength of the penalty applied to the model during the variable/feature selection process.

2. For each $\lambda \in \Lambda$:
    - Draw a sub-sample of size $\frac{n}{2}$ without replacement denoted by $I^* \subseteq \{1, \ldots n\}$
    - This $I^* = \frac{n}{2}$
    - Run the selection algorithm $\hat{S}^{\lambda}$ on I^*

3. Repeat step 2 a number of times to compute the relative selection frequencies - the percentage of certain variables that have been selected during this subsample run. Mathematically speaking:

$$\hat{\Pi}_j^{\lambda} = P^*\left[j \in \hat{S}^{\lambda}\right] = \frac{1}{N} \sum_{\{i=1\}}^{N} \mathbb{I}_{\left\{j \in \hat{S}_i^{\lambda}\right\}}$$

This P^* that the covariates j has been selected by $\hat{S}^{\lambda}(I^*)$ is with respect to the subsampling mechanism when we run a randomized selection algorithm

4. Typically, in LASSO, we can see that there are various λ values, so we look at the whole path because we don't know the optimal lambda. Thus, we examine numerous regularization parameters over the entire set of Λ. Given the selection probabilities for each component and for each value of λ, we can construct a stable set:

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

Where π_{thr} is our predefined threshold

The stability selection definition $\hat{S}^{stable}$ these are the covariates for which we take the maximum over all $\lambda \in \Lambda$. Such that the selection frequency the maximum over all λ is greater than predefined threshold.

Theory

How should we choose this π_{thr}?

Notations:

Let S be the true variables and N be the noise variable such that $S \cup N = \{1,2,\ldots,p\}$

Let $\hat{S}^{\Lambda} = \cup_{\{\lambda \in \Lambda\}} \hat{S}^{\lambda}$ be the set of selected variables if varying the regularization λ in the set Λ. It refers to the set of selected structures or variables when varying the regularization parameter λ within a specified set of regularization parameterΛ.

By varying λ within a certain range we can observe how the selection of variables for the model changes

By varying this lambda within a certain range, we can observe how the selection of variables for the model changes.

This Λ represents a predefined range of regularization parameter values λ It is used to explore the effects of different regularization strength on variable selection and algorithm performance.

Typically, if we make our λ larger we select few and few variables so this $\hat{S}^{\lambda} \approx \hat{S}^{\lambda_{\min}}$

Let q_{Λ} be the average number of selected variables, $q_{\Lambda} = E(\mid \hat{S}^{\Lambda}(I^{*}) \mid$.

This can be computed by taking the average of subsampling sizes.

V be the number of falsely selected variables with stability selection,$V = \mid N \cap \hat{S}^{stable} \mid$

Under the exchangeability assumption and the assumption that our variable selection procedure is better than random guessing represents by.

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

We can prove that:

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

The expected number of false positives is bounded by this quantity on the right hand side, this quantity depends on 3 variables π_{thr}, p, $\hat{q}_{\Lambda}$

Notice that p in the denominator could be very large but it's suggesting that there's a blessing of dimensionality where the bound on the false selection rate improves as p increases, however this bound is only controlling 2 type of errors, the number of signal variables that don't get selected, and the number of noise variables that get selected. As p increases its unlikely that the noise variable will keep cropping up repeatedly enough time to be chosen by stability selection. We can also choose the threshold and invert the formula so that we can control the expected number of false positive to be less than 1 and control the family wise error rate

e.g. Cut-off value π_{thr} = 0.9, choosing the regularization parameters Λ such that $\mathrm{q}_{\Lambda} = \sqrt{0.8\mathrm{p}}$ will control $\mathrm{E(V)} \leq 1$.

Or choosing Λ such that $\mathrm{q}_{\Lambda} = \sqrt{0.8\alpha\mathrm{p}}$ controls the familywise error rate at level α

$$\mathrm{P(V > 0) = P(V \geq 1) \leq E(V) \leq \alpha}$$

Conditions:

As mentioned previously, there are two conditions: the exchangeability condition and the requirement that our base method performs better than random guessing.

The exchangeability condition necessitates that the distribution $\left\{I_{\{j\in\hat{S}^{\lambda}]},\ j \in S^{c}\right\}$ takes an indicator and examines whether the covariate j is chosen by the selector $\hat{S}^{\lambda}$. We only care about j being noise variables; the distribution among these covariates must be exchangeable, meaning that j is invariant to permutation. In simpler terms, noise variables must have the same selection probability under the base procedure. Often, we might encounter situations where some noise variables are highly correlated with signal variables, while others are not as strongly correlated.

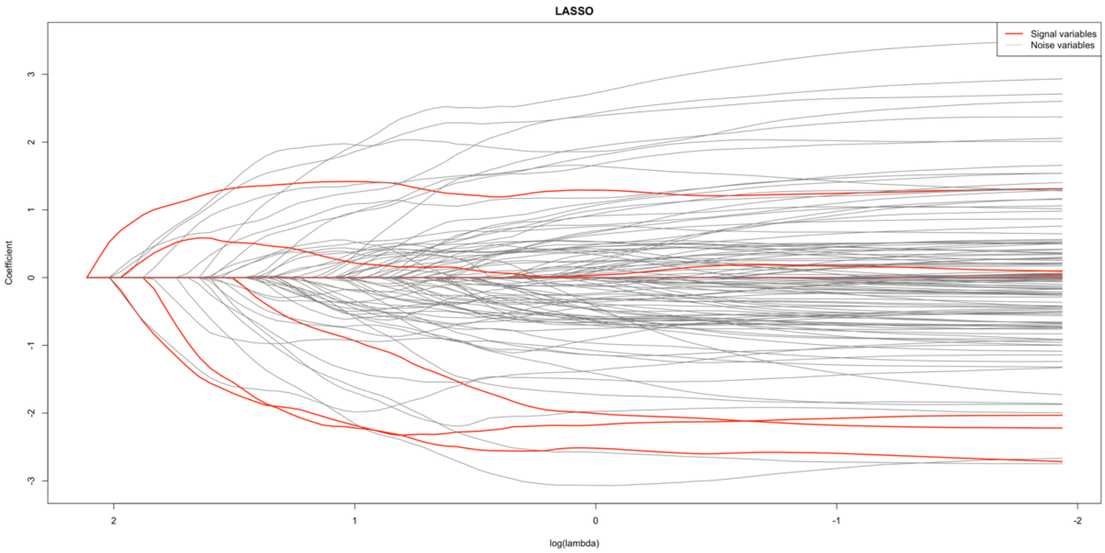
Pros:

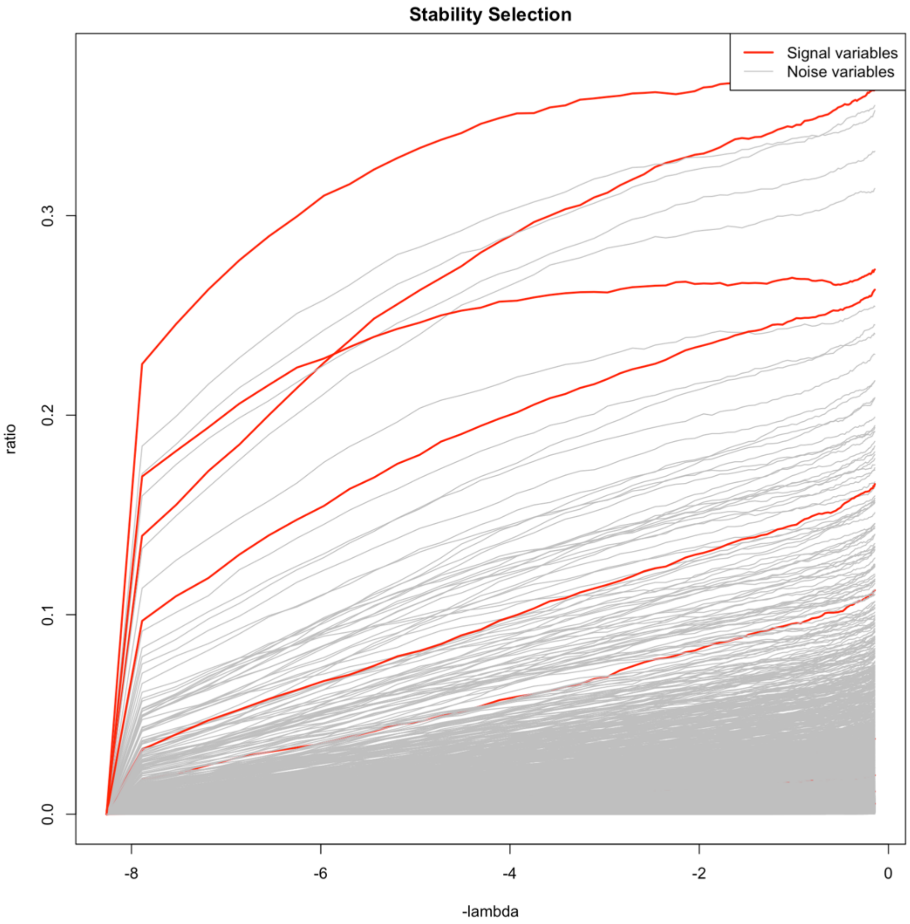
The good thing about this method is that we can apply stability selection to any existing variable selection algorithm. In addition, stability selection provides error control in the form of a finite sample upper bound on the expected number of falsely selected variable.

Cons:

The computational complexity is quite expensive, this is due to fitting multiple models on different subsamples and aggregating the results which is a technique highlighted in ensemble feature selection. Another drawback would be the tuning parameters where one must be cautious when validating these parameters to achieve the optimal results.

Simulation





In this simulation setting, we set our p to be very large number and n to be significantly smaller than p. Among these p, we know that 10 of them are the true covariates, while the other p are permuted noise variables.

The graph below illustrates the efficiency of Lasso and Stability Selection with Lasso, where we can see that with Lasso, our true covariates in red are difficult to identify. In contrast, with Stability Selection, the true covariates lines stand out much more clearly from the noise variables. It is essential to note that Stability Selection cannot be represented merely by choosing a suitable lambda; the process provides a fundamentally different solution from the base procedure.

Discussion

In conclusion, stability selection is a technique introduced by Nicolai Meinshausen and Peter Bühlmann in 2010, designed to improve the stability of feature selection. In the modern field of data science, we frequently deal with high-dimensional data settings where our goal is to identify signal variables while avoiding falsely selected variables when building models. Stability selection opens new doors to the concept of stability as it can be used with any existing variable selection method and improve its performance. Additionally, it provides error control for the expected number of falsely selected variables. However, this technique has its drawbacks, such as its computational expense and the requirement for users to be attentive when validating parameters to control falsely selected variables. Furthermore, the assumption of exchangeability must be met before using the technique. On the computational side, our modern world is well-adapted to handling high-dimensional and complex datasets. In this project, the difficulty encountered while attempting to replicate one of the simulations in the paper lies in the coding and computational aspects. Although the methodology is straightforward, implementing it in R presents challenges. Most well-known methods have built-in functions in programming languages, but in the case of stability selection, built-in libraries like c060 and stabs, which are supposed to help with plotting the stability score against lambdas, are rendered useless and require manual implementation of the stability selection process. Nevertheless, stability selection is a powerful tool for identifying signal variables and avoiding the selection of false variables when building predictive models.

References:

Thuijskens, B. (2018, July 25). Stability selection. Retrieved from https://thuijskens.github.io/2018/07/25/stability-selection/#fn:1

Meinshausen, N., & Bühlmann, P. (2010). Stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(4), 417-473. https://doi.org/10.1111/j.1467-9868.2010.00740.x

Bühlmann, P. (2008). Stability selection for high-dimensional data. VideoLectures.net. Retrieved from http://videolectures.net/sip08_buhlmann_ssfhd/

Seonghyun23. (n.d.). Sukhyun23/stability_selection: A Python package to perform stability selection. GitHub. Retrieved from https://github.com/sukhyun23/stability_selection

Hofner, B., & Hothorn, T. (n.d.). plot.stabpath: Plot Stability Paths. R Documentation. Retrieved from https://rdrr.io/rforge/c060/man/plot.stabpath.html

Tibshirani, R. (n.d.). Stability selection. Carnegie Mellon University. Retrieved from https://www.stat.cmu.edu/~ryantibs/journalclub/stability.pdf

Sivaranjani, S. (2017). Stability selection for high-dimensional data [Doctoral dissertation, University of Manchester]. University of Manchester Library. Retrieved from https://pure.manchester.ac.uk/ws/portalfiles/portal/66045529/FULL_TEXT.PDF