gStab: An R Package for Measuring Stability of Feature Selection Methods

Abdul Wahid[1,*] and Dost Muhammad Khan[2]

[1]Department of Mathematics and Statistics, Institute of Southern Punjab Multan, Pakistan.

[2]Department of Statistics, Abdul Wali Khan University Mardan, Pakistan.

*Corresspond Author: ab.wahid1996@gmail.com

Abstract: *In this article, a new R package gStab is developed for the evaluation of stability in feature selection learning's. The main characteristic of this approach is measuring stability within subset and among selected subsets by feature selection methods in different subsampling experiments. Hence, gStab package is applicable in more general scenarios when the feature selection methods return a constant number of features or the number of features chosen is not pre-determined by the user. Firstly, the Absolute Shrinkage and Selector Operator (LASSO) is applied for the purpose of feature selection using real-world and simulated datasets. Secondly, an average stability is computed of the feature selection of LASSO by using gStab package. We can optimize the value of hyper parameter of LASSO that results higher stability. An important conclusion is that optimizing stability can be potentially achieved without significant loss of accuracy, and can help recognizing the true underlying set of features using R package gStab.*

Keywords: *Stability; Feature Selection; High-dimensional Data; Subsampling; gStab*

I. INTRODUCTION

Stability of the feature or variable selection algorithms is recent and an important issue, especially when dealing with high-dimensional data. Stability of a feature selection technique or algorithm is the stochastic nature of its selected subsets due to small changes in training data [1]. For example, a technique is comparatively more stable when a little variation in the training dataset leads to less variability in the selected subsets of features. Applications lie in many scientific fields, such as biomedicine, bioinformatics, artificial intelligence and pattern recognition, where the available datasets are in high-dimensions.

Stability of the feature or variable selection algorithms is recent and an important issue, especially when dealing with high-dimensional data. Stability of a feature selection technique or algorithm is the stochastic nature of its selected subsets due to small changes in training data [1]. For example, a technique is comparatively more stable when a little variation in the training dataset leads to less variability in the selected subsets of features. Applications lie in many scientific fields, such as biomedicine, bioinformatics, artificial intelligence and pattern recognition, where the available datasets are in high-dimensions.

The stability of learning methods with respect to small fluctuations in training data is commonly considered a desired property of methods, as it guarantees that the feature selection models are robust and are significantly influenced by noisy features or data perturbations [1]. Moreover, it is examined that stability is affected by data nature, such as noise [2], sample size and number of variables in data [3] and feature redundancy [4]. Recently, several stability approaches for evaluating learning algorithms have been proposed [5, 6, 1], etc). Some of these stability measures are included in the R package **stabm** [7].

In this article, an *R* package *gStab* is introduced, which implements two indices that find the stability within selected subset and between selected subsets in different random sub-sampling experiments. This package computes stability of learning algorithm that select subsets either containing fixed number of features or non-constant number of features. The development of *gStab R* package has been motivated by our recent contribution [8], in which we proposed a generalized stability estimator that measures similarity within the selected subsets and variability among the selected subsets in terms of number of features. Therefore, this new *R* package named *gStab*.

The rest of this paper is organized as follows: A short description to the methodology is given in the following section 2. Then, a detailed description of the *R* functions for package *gStab* is presented in next section. In section 4, we present numerical experiments by employing *gStab* to both simulated and real-world datasets. Finally, last section describes some concluding remarks and future work.

II. METHODOLOGY

In this section, the methodology of the proposed *R* package is presented. Here, we start with the following notations:

y^n: denotes the dependent variable, either continuous or dichotomous.

$X^{n\times p}$: represents the features matrix with $i = 1,2,...,n$ and $j = 1,2,...,p$. We use the subscript i to index sample observations and subscript j for feature number.

m: denote the number of random sub-sampling experiments.

$\mathcal{F}^{m\times p}$: it is the feature selection matrix of features, for $k = 1, 2, ..., m$.

e_{kp}: a binary value in $\mathcal{F}$, where $e_{kp} = 1$ when the *pth* feature in the *kth* sub-sampling experiment is selected, and $e_{kp} = 0$ otherwise.

Let us assume that we choose m random sub-sampling datasets from the original data, and each sub-sampling set contains all p features. A vector denoted by $\alpha = (\alpha_1,\ldots,\alpha_k,\ldots,\alpha_m)^T$ is introduced, where α_k is the number of selected features in the *kth* experiment, i.e.,

$$\alpha_k = \sum_{j=1}^{p} I(e_{jk} = 1) \quad (1)$$

To find the stability among the chosen subsets by a learning technique, is denoted by $\phi_k(\alpha)$ and defined as follows:

$$\phi_k(\alpha) = \begin{cases} \frac{\alpha_1}{p} \; for \; k = 1 \\ \frac{\alpha_{k-1}}{\alpha_k} \; if \; \alpha_k > \alpha_{k-1} \; for \; k > 1 \\ \frac{\alpha_k}{\alpha_{k-1}} \; if \; \alpha_k < \alpha_{k-1} \; for \; k > 1 \\ 1 \; if \; \alpha_k = \alpha_{k-1} \; for \; k > 1 \end{cases}$$

The range of $\phi_k(\alpha)$ is [0, 1]. The larger value of $\phi_k(\alpha)$ indicates the better performance. Let

$$\gamma_j = \sum_{k=1}^{m} I(e_{jk} = 1) \quad (2)$$

be the selection frequency of the *jth* feature or variable in all m experiments and $\gamma = (\gamma_1,\ldots,\gamma_j,\ldots,\gamma_p)^T$. Furthermore, the γ vector is transformed into an m-dimensional vector as $\delta = (\delta_1,\ldots,\delta_k,\ldots,\delta_m)^T$, which denotes the empirical density of vector γ. The value of δ_k denotes the number of variables which are chosen in exactly k of the m random experiments, and defined as follows:

$$\delta_k = \sum_{k=1}^{m} I(\gamma_j = k) \quad (3)$$

Moreover, unified the quantities $\phi_k(\alpha)$ and δ in the following equation and define final stability estimator as

$$\hat{\pi}_F = \frac{1}{m^2} \sum_{k=1}^{m} \frac{k^2.\delta_k.\phi_k(\alpha)}{\alpha_k} \quad (4)$$

On the one hand, the δ measures within selected subset stability and on the other hand, the quantity $\phi_k(\alpha)$ measure the stability among the selected subsets. The k^2 determines that variables that are chosen in a high frequency of m random experiments were weighted stronger than variables with low frequencies. For more detail (see [9]). The whole procedure is illustrated by an empirical example in Figure 1.

		Feature or variable					
	m	x_1	x_2	x_3	x_4	α_k	$\phi_k(\alpha)$
Sub-sampling Experiment	1	0	1	1	0	2	0.5
	2	1	0	1	1	3	0.67
	3	1	1	1	0	3	1
	4	1	1	1	0	3	1
	5	0	0	1	0	1	0.33
	γ_j	3	3	5	1		
	δ_k	1	0	2	0	1	

Figure 1: An Empirical Illustration of the Stability Estimator Define in Eq.(4). In this example, we set m= 5 and p= 4. The stability score of $\hat{\pi}_F$ is calculated as: $\hat{\pi}_F = \frac{1}{5^2}\left[\frac{1^2.1.0.5}{2} + 0 + \frac{3^2.2.1}{3} + 0 + \frac{5^2.1.0.33}{1}\right] = 0.58$

III. THE NEW R PACKAGE

In this section, usage of the main function *gStab* and its arguments is discussed. The *gStab* package. depends on the other *R* package ***MASS***. First, a feature selection technique can be implemented on original data (y_i, X), by installing and loading concerned or relevant *R* packages. A feature selection technique should be applied separately on each training subsample, and generates a matrix of chosen features for all m subsampling experiments. Then determine that how stable these feature sets in m experiments are, by using *gStab* package.

In the first place, the function *selection()* is implemented before using the *gStab* package as:

```
selection <- function(x){
  j <- 1:length(x) {
  ifelse(x[j]!= 0, 1, 0) }
```

The package g*Stab* has been loaded as usual with library("*gStab*"), the main function a user requires to call up is as follows:

```
gStab(x,K,Verbose=FALSE)
```

The available arguments are *x*, *K* and Verbose. The argument *x* is the feature selection matrix over m subsampling experiments by a learning technique, while *K* represents the number of training subsamples. Note that the argument *Verbose* is logical and if TRUE, more information of the package *gStab* is returned. The default is FALSE which give only the stability score of Eq.(4).

IV. NUMERICAL EXAMPLES

This section illustrates the capabilities of new *R* package *gStab* on a benchmark gene expression data, as well as on a simulated dataset. Here, the LASSO [7] is considered as a feature selection method, which is included in *R* package ***glmnet***. The cross-validation technique is employed to find the optimal tuning parameter lam for the LASSO. All computations are carried out on a 64-bit Intel machine with 1.00 GHz CPU and 4.00 GB of RAM.

A. Simulated Data

In this example, we present a synthetic data example and evaluate the stability performance of the LASSO. In this simulation setup, the original data matrix $X^{n\times p}$ with n= 500 and p= 50 is generated from the multivariate normal distribution $N(0,1)$. The outcome variable y_i is generated from standard linear model:

$$y_i = \sum_{j=1}^{p} \beta_j X_{ij} + \epsilon_i \; for \; i = 1, 2, \ldots, n$$

where ϵ_i's are independent and identically distributed errors from $N(0,1)$ and $\beta = (\beta_1,\ldots,\beta_p)^T$ is a pdimensional vector consisting of first 20 non-zero coefficients (relevant features) and the remaining 30 zeros (noise features). The redundancy between feature x_j and x_l was set to be $\rho^{|j-l|}$ with ρ= 0.2. Furthermore, 50 simulation runs are performed and the average of stability score is reported. The number of random

sub-sampling experiments are taken as m = 100. We have set up the following function *Generation()* for generated relevant features:

```
Generation < − function(n,R,bet){
X < − mvrnorm(n,rep(0,length(bet)),R);
Y < − rep(0,n);
beta < − matrix(bet,ncol=1);
err < − rnorm(n,0,1);
Y < − X%*%bet + err;
list(X=X,Y=Y); }
```

The noise features are generated independently from $N(0,1)$. Nevertheless, to find stability of the LASSO over m = 100 sub-sampling experiments, the following codes are executed:

```
library(glmnet)
library(MASS)
library(ISLR)
source("F:/The R Journal/gstab.R")
n < − 500;p < − 100;nzc < − 50;zc < − 50;r < − 0.2;rho < − 0.2;
beta < − rnorm(nzc);
cvec < − c(beta,rep(0,zc));
corr < − matrix(0,nzc,nzc);
for(i in seq(nzc)){
for(j in seq(nzc)){
corr[i,j] < − rho^abs(i-j);
}
}
cr < − floor(n*(1-r));
sam < − sample(1:n,cr,replace=F);
m < − 100;runs < − 50;
grid < − seq(0.1,1,length=runs);
laso FS < − matrix(0,nrow=m,ncol=p);
stability < − rep(0,runs);
for(j in 1:runs){
for(i in 1:m){
sim.data< −Generation(n, R=corr,bet=beta);
Xr < − matrix(rnorm(n*zc),n,zc);
XX < − cbind(sim.data$X,Xr);
y < − sim.data$Y;
d1 < − data.frame(cbind(y,XX));
colnames(d1)=c("y",paste0("f",seq(p)));
X5 < − model.matrix(y∼.,d1)[,-1];
y5 < − d1$y;
X5.train < − X5[sam,];
X5.test < − X5[-sam,];
y5.train < − y5[sam];
y5.test < − y5[-sam];
mod.lasso< −
glmnet(X5.train,y5.train,family="gaussian",alpha=1,
lambda= grid[j]);
laso FS[i,] < − coef(mod.lasso)[2:101,];
}
stability[j] < − gStab(laso FS,K=m);
}
meanStab < − mean(stability);
```

To run the above codes we obtain the following output:

```
> meanStab < − mean(stability);
> meanStab
0.7759772
> stability
0.8750750, 0.8430831, 0.8248260, 0.8320092, 0.8605829,
0.8621636, 0.9074591, 0.8731464, 0.8849457, 0.9175898,
0.9071261, 0.9004661, 0.8171632 ,0.8952447 , 0.9255768,
0.8375710, 0.8723764, 0.8831744, 0.8683178, 0.8495601,
0.8974107, 0.8678137, 0.8055741 ,0.8042673, 0.7960376,
0.7376750, 0.7633126, 0.6956213, 0.8387217, 0.7384191,
0.6748816, 0.6802028 0.6971497, 0.7299358, 0.6611369,
0.6899050, 0.6915134, 0.7017685, 0.6993993 ,0.6559976
,0.6745231, 0.6604722 , 0.6487309, 0.6914595, 0.6760169,
0.6396359, 0.6528464, 0.6421226 ,0.6493926,0.5994600
```

Furthermore, we plot the stabilty score against 50 regularization parameters of LASSO in the range [0.1, 1] by using *R* package ggplot2. Figure 2 shows the plot from the following code.

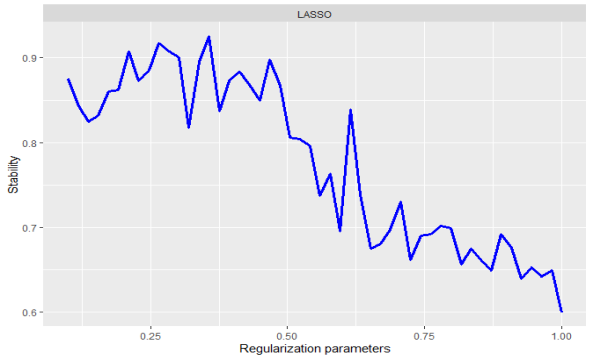

Figure 2: The Stability Scores Against Regularizing Parameters of LASSO.

```
library(ggplot2);
results < − data.frame("Method" =
rep("LASSO",50),Hyperparameters=grid ,stability)
;results$Hyperparameters < −
as.numeric(as.vector(results$Hyperparameters))
p1 < − ggplot(results, aes(x=Hyperparameters,
y=stability)) + geom line(size=0.75)+ geom
line(size=1.2,color="blue")+ xlab("Regularization
parameters")+ ylab("Stability")+ facet wrap( Method)
;p1+theme();
```

However, as can be seen from Figure 2 that for some values of regularization parameters the LASSO achieves maximum stability. The best value can be obtained by cross-validation approach in glmnet package and the required functions are easily incorporated in the above code. Besides, the *predict()* function is also available in glment to find prediction error by exploiting test data sets (see *X5.test*, *y5.test* in above code).

B. Colon Data

In this example, a high-dimensional Colon tumor data is considered which contains n = 62 samples and p = 2000 genes (or features). The outcome variable (y) is binary and the distribution of y consists of 40 tumor tissues and 22 normal tissues. This dataset is freely available as part of the *R* package

plsgenomics. Here, we also set *m* = 100 and stability of the LASSO is computed as follows in *R*:

```
laso_FS < − matrix(0, nrow=runs,ncol=p);
for(i in 1:runs){
d1 < − data.frame(cbind(Y,X));
colnames(d1) < − c("y",paste0("f",seq(p)));
X5 < − model.matrix(y˜.,d1)[,-1];
y5 < − d1$y;
 X5.train =X5[sam,];
 X5.test =X5[-sam,];
y5.train =y5[sam]; y5.test = y5[-sam];
mod.lasso < − glmnet(X5.train,y5.train, family="binomial",alpha=1,lambda= grid);
cv.out < − cv.glmnet(X5.train,y5.train, alpha=1,family="binomial",nfold=10);
bestlam < − cv.out$lambda.min;
 best.model< −glmnet(X5.train, y5.train,alpha=1,family="binomial", lambda=bestlam);
laso_FS[i,]= coef(best.model)[2:2001,];
}
stability < − gStab(laso FS,K=runs);
```

The output of the above code is as under:

```
> stability
0.8049302
 dim(laso FS)
100 2000
```

From the output, we can see that stability of the LASSO is 0.8049302 over 100 subsampling sets. In the above code we also determine the best value for regularizing parameter of LASSO by using 10-fold cross-validation for each subsampling experiment. The visualization of feature selection matrix that is "laso_FS" by using ***ggplot2*** has also shown in heatmap (Figure 3). The selected genes (features) over m = 100 experiments were shown by black vertical lines in the Figure 3. The plot shows that a few genes are selected by LASSO in almost all subsampling experiments while the most of the genes are not selected which is shown by the light blue color.

Figure 3: Heatmap for Visualizing Selected and Not-Selected Genes by LASSO for Colon Tumor Data

V. SUMMARY

This article has introduced the *R* package called *gStab* for measuring stability of variable selection methods. The *gStab* can be used to assess the stability in subsets of features chosen by methods that produce any sizes of subsets and applicable in different dimensions and large datasets. In numerical experiments, the new *R* package is implemented to quantify the stable selection of variables subsets of LASSO method on both synthetic and a high-dimensional real datasets. We can also implement it to other large number of state-of-the-art supervised learning methods and compare their stability performance due to small perturbations in training data. However, examples illustrating the use of *gStab* in practical applications have been presented. The *R* code of *gStab* for numerical experiments is given in the article to enable research reproducible. The implementation of *gStab* package to assess the unsupervised feature selection methods [8] and their comparison in terms of stability is part of the future work.

REFERENCES

[1] S. Nogueira, K. Sechidis and G. Brown, "On the Stability of Feature Selection Algorithms", *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6345–6398, 2017.

[2] A. A. Shanab, T. M Khoshgoftaar, R. Wald and A. Napolitano, "Impact of Noise and Data Sampling on Stability of Feature Ranking Techniques for Biological Datasets", *IEEE 13th International Conference on Information Reuse & Integration (IRI)*, 08-10 August, 2012, USA, pp. 415–422.

[3] S. Alelyani, "On Feature Selection Stability: A Data Perspective", *Arizona State University*, 2013.

[4] R. Wald, T. M. Khoshgoftaar and A. Napolitano, "Stability of Filter and Wrapper-based Feature Subset Selection", *IEEE 25th International Conference on Tools with Artificial Intelligence*, 04-06 November, 2-13, USA, pp 374–380.

[5] R. A-Rodriguez and A. C. Parnell, "An Information Theoretic Approach to Quantify the Stability of Feature Selection and Ranking Algorithms", *Knowledge-Based Systems*, vol. 195, pp. 1-13, 2020.

[6] W. W. B. Goh and L. Wong, "Evaluating Feature-Selection Stability in Next-Generation Proteomics", *Journal of Bioinformatics and Computational Biology*, vol. 14, no. 5, pp. 1-23, 2016.

[7] A. Bommert and M. Lang, "stabm: Stability Measures for Feature Selection", *Journal of Open Source Software*, vol. 6, no. 59, pp. 1-4, 2021.

[8] A. Wahid, D. M. Khan, I. Hussain, S. A. Khan and Z. Khan, "Unsupervised Feature Selection with Robust Data Reconstruction (UFS-RDR) and Outlier Detection", *Expert Systems with Applications*, vol. 201, 117008, 2022.

[9] A. Wahid, D. M. Khan, N. Iqbal, H. T. Janjuhah and S. A. Khan, "A Generalized Stability Estimator Based on Inter-Intrastability of Subsets for High-Dimensional Feature Selection", *Chemometrics and Intelligent Laboratory Systems*, vol. 220, 104457, 2022.