Ensemble Based Ultrahigh Dimensional Variable Screening

by

Dong Yang

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Statistics

Department of Mathematical and Statistical Sciences
University of Alberta

Abstract

With the development of modern technology, researchers in various fields are equipped with powerful tools to collect ultrahigh dimensional data, where the number of features p could grow exponentially with the sample size n. It is challenging to extract useful information due to the huge number of features. To tackle this challenge, Fan and Lv [14] proposed the two-scale approach where variable screening procedure is applied first instead of traditional one-scale variable selection. The purpose of variable screening is to eliminate as many noisy features as possible while keep all the important features. There are many variable screening methods that work well with various assumptions. However, most of them are not stable in a sense that a small perturbation in the sample may result in very different selected features. On the other hand, it is difficult to verify all the assumptions in reality. Therefore, a generic guideline is desired to select appropriate screening methods that fit different applications. A natural choice is to combine multiple screening methods to adapt more general assumptions. In this thesis, we propose a group of ensemble methods to aggregate results from multiple screening methods. Our methods are capable of providing stable results and work well even if some of the candidate screening methods fail. In particular, we propose three ensemble approaches to encourage stability, namely, parallel ensemble screening, quantile ensemble screening and multi ensemble screening. We show that each of the proposed procedure has the sure screening property, which means the selected set contains the true active variables with a probability tending to one provided each of the method combined shows sure screening property. We validate our methods through both simulation studies and real data analysis.

Acknowledgements

First of all, I would address my gratitude to my supervisor Dr. Linglong Kong for his kind guidance and patience. Without his help, I could not finish this thesis alone. Dr. Kong's experience and wisdom are like navigation light that guide my research in the unknown.

Secondly, I would like to thank my colleague Wei Tu for his kind help. We had a lot of discussions about my research and some of those valuable ideas came up during those discussions.

I also wish to thank Dr. Ivan Mizera, Dr. Rohana Karunamuni and Dr. Di Niu for serving as members of my thesis examination committee.

Last but not least, I would like to thank all the instructors who have given me those meaningful lectures that help me build my statistics foundations and all the support staffs in Department of Mathematical and Statistical Sciences who have offered me a lot convenience during my research and study.

It is a great experience to have my two years master study in University of Alberta. Edmonton is a lovely city that embraces new ideas and full of energy. During my stay, I have met many good friends and colleagues, we had some unforgettable memories together.

Special thanks to my parents and my sister, thanks for their support.

Table of Contents

1 Introduction **1**

1.1 Variable Screening 1

1.2 Ensemble Methods 9

1.3 Contributions 13

2 Ensemble Based Ultrahigh Dimensional Variable Screening **15**

2.1 Preliminaries 15

2.2 Ensemble Screening Methods 19

2.3 Technical Results 22

3 Numerical Studies **29**

3.1 Simulation Studies 29

3.2 Real Data Analysis 37

4 Conclusions and Future Research **42**

Bibliography **44**

Appendices **50**

List of Tables

3.1 Variable selection for mixture screening 41

1 Example 1, Parallel Ensemble, Error= Normal 51

2 Example 1, Parallel Ensemble, Error= Cauchy 51

3 Example 1, Parallel Ensemble, Error= Mixture Normal 52

4 Example 1, Quantile Ensemble, Error= Normal 54

5 Example 1, Quantile Ensemble, Error= Cauchy 54

6 Example 1, Quantile Ensemble, Error= Mix Normal 55

7 Example 1, Multi Ensemble, Error= Normal 55

8 Example 1, Multi Ensemble, Error= Cauchy 55

9 Example 1, Multi Ensemble, Error= Mixture Normal 56

10 Example 1, Mixture Ensemble, Error= Normal 56

11 Example 1, Mixture Ensemble, Error= Cauchy 56

12 Example 1, Mixture Ensemble, Error= Mixture Normal 57

13 Example 2(a), Parallel Ensemble, Error= Normal 57

14 Example 2(a), Parallel Ensemble, Error= Cauchy 58

15 Example 2(a), Quantile Ensemble, Error= Normal 59

16 Example 2(a), Quantile Ensemble, Error= Cauchy 59

17 Example 2(a), Quantile Ensemble, Error= Mix Normal 60

18 Example 2(a), Multi Ensemble, Error= Normal 60

19 Example 2(a), Multi Ensemble, Error= Cauchy 60

20 Example 2(a), Mixture Ensemble, Error= Normal 61

21 Example 2(a), Mixture Ensemble, Error= Cauchy 61

22 Example 2(b), Parallel Ensemble, Error= Normal 61

23 Example 2(b), Parallel Ensemble, Error= Cauchy 62

24 Example 2(b), Parallel Ensemble, Error= Mixture Normal . . . 63
25 Example 2(b), Quantile Ensemble, Error= Normal 64
26 Example 2(b), Quantile Ensemble, Error= Cauchy 65
27 Example 2(b), Quantile Ensemble, Error= Mix Normal 65
28 Example 2(b), Multi Ensemble, Error= Normal 66
29 Example 2(b), Multi Ensemble, Error= Cauchy 66
30 Example 2(b), Multi Ensemble, Error= Mixture Normal . . . 66
31 Example 2(b), Mixture Ensemble, Error= Normal 67
32 Example 2(b), Mixture Ensemble, Error= Cauchy 67
33 Example 2(b), Mixture Ensemble, Error= Mixture Normal . . 67
34 Example 3, Parallel Ensemble, Error= Normal 68
35 Example 3, Parallel Ensemble, Error= Cauchy 68
36 Example 3, Parallel Ensemble, Error= Mixture Normal 69
37 Example 3, Quantile Ensemble, Error= Normal 71
38 Example 3, Quantile Ensemble, Error= Cauchy 71
39 Example 3, Quantile Ensemble, Error= Mix Normal 72
40 Example 3, Multi Ensemble, Error= Normal 72
41 Example 3, Multi Ensemble, Error= Cauchy 72
42 Example 3, Multi Ensemble, Error= Mixture Normal 73
43 Example 3, Mixture Ensemble, Error= Normal 73
44 Example 3, Mixture Ensemble, Error= Cauchy 73
45 Example 3, Mixture Ensemble, Error= Mixture Normal 74

List of Figures

1.1 Methods of model selection with ultrahigh dimensionality . . . 3
1.2 A common ensemble architecture 9
1.3 Comparison of bagging and stable selection 10
1.4 Comparison of boosting and ISIS 12

3.1 Example 1, quantile ensemble, $\boldsymbol{\varepsilon}$ =normal 32
3.2 Example 2(b), mixture ensemble, $\boldsymbol{\varepsilon}$ =Cauchy 33
3.3 Example 3, parallel ensemble(PSIS), $\boldsymbol{\varepsilon}$ =normal 34
3.4 Example 3, parallel ensemble(PSIRS), $\boldsymbol{\varepsilon}$ =normal 35
3.5 Example 3, multi ensemble, $\boldsymbol{\varepsilon}$ =mixture normal 36
3.6 Example 3, mixture ensemble, $\boldsymbol{\varepsilon}$ =mixture normal 37
3.7 Classification results for ADHD-200 40

Chapter 1

Introduction

1.1 Variable Screening

With the developing of modern data collecting techniques, researchers in various fields are equipped with powerful tools to collect high dimensional data more efficiently. For example, the gene expression microarray data [20, 44], single nucleotide polymorphism (SNP) [19, 25] data, magnetic resonance imaging (MRI) [48, 57] data are considered as high dimensional. Those data in general have very high dimensions which makes it difficult to extract useful information. For example, we may try to discover a candidate biomarker from the microarray gene expression data which is derived by measuring gene expression levels of thousands of genes simultaneously. Due to the cost of data collection or the difficulty to find enough volunteers for some specific diseases, the sample size n will not be large. To handle model selection or variable selection with high dimension in a situation of $p > n$, various methods have been proposed. To name a few, the LASSO [50], Adaptive LASSO (Adalasso) [58], SCAD [13], Danzig Selector (DS) [7] and MCP [54] are frequently used in variable selection scenarios.

The existing variable selection methods, although proved to be useful in high dimensional setting of $p > n$ deteriorate when the dimension goes to ultrahigh. The word "ultrahigh" here means a scenario that the dimension p is much larger than the sample size n or p even grows exponentially with n. In

MRI studies for example, depending on the scanner used, image quality could have a resolution as high as 1690 × 1744 [49]. Given the MRI scan results, researchers are typically interested in finding certain pixels or regions that are related to certain phenotypes. Because of the high cost of the MRI scan, we only have limited sample size which may be about 100, thus leading to a situation of $p >> n$. Another example could be found in the analysis of gene expression data in which the sample size sometimes is around one hundred and the number of candidate genes is more than $10,000$. The researchers will usually be interested in finding some specific genes relating to some diseases which are usually less than or around 10 [2, 9]. In the above situations, the aforementioned variable selection methods will perform poorly or even fail to work due to the simultaneous challenges of computational expediency, statistical accuracy and algorithm stability [15].

To overcome the issues associated with ultrahigh dimensionality, Fan and Lv [14] introduced the sure independence screening (SIS) based on correlation learning. A two-scale approach proposed by Fan and Lv [14]works as follows: Firstly, we decrease the dimension p by large scale by applying variable screening methods such as SIS to reduce the dimensionality to a moderate scale d that is usually below the sample size n. Secondly, we apply lower dimensional variable selection approaches to select the truly important features. As a result, the variable screening methods could be considered as a prior step to analyze ultrahigh dimensional data in order to make it applicable for the relatively low dimensional variable selection methods to work. The sure screening property proposed by Fan and Lv [14] is desired in the screening process, that is, with probability approaching 1, the screening algorithm keeps all of the true active variables. Motivated by SIS, a large number of variable screening methods have been proposed. See [12, 21, 56] and their references.

In the following, we introduce some of the variable screening methods which are closely related to our methods. Considering the problem of variable screening in ultrahigh dimensional feature space, we have a random sample $\{\boldsymbol{X_i}, Y_i\}_{i=1}^n$ from the population $(\boldsymbol{X}, Y)$ in which Y is the response and $\boldsymbol{X} = (X_1, \ldots, X_p)$ is the associated covariate vector. Without loss of general-

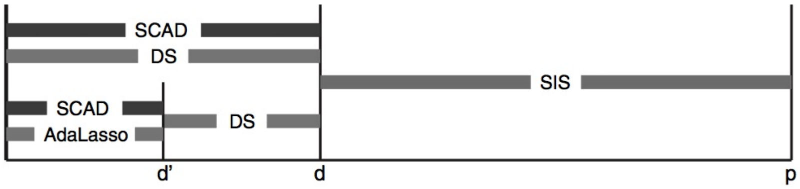


Figure 1.1 *Methods of model selection with ultrahigh dimensionality [14]: Variable screening methods such as SIS work when the dimension p is ultrahigh. After applying SIS, the dimension is decreased to a relative low dimension d which is suitable for the low dimensional variable selection methods to work in the second step.*

ity, we assume $\boldsymbol{X}$ are standardized column-wisely. We adopt the same setting in the following. Fan and Lv [14] introduced the sure independence screening (SIS) procedure by ranking all predictors using a utility measure between the response and each predictor. Mathematically, the SIS procedure could be summarized as follows: Suppose $\boldsymbol{\omega} = (\omega_1, \omega_2, \ldots, \omega_p)$ is a p-vector obtained by component-wise regression of Y on $\boldsymbol{X}$, that is

$$\boldsymbol{\omega} = \boldsymbol{X}^T Y. \tag{1.1}$$

Here $\boldsymbol{\omega}$ is used as the screening utility which is a vector of marginal correlations of predictors with the response variable, rescaled by the standard deviation of the response. For any given $\gamma \in (0,1)$, the sorted p component-wise magnitudes of the vector $\boldsymbol{\omega}$ in a decreasing order is derived and a sub-model is defined as

$$\mathcal{A}_\gamma = \{1 \leq j \leq p : |\omega_j| \text{ is among the first } [\gamma n] \text{ largest of all}\}. \tag{1.2}$$

After this step, the original model size is shrank to $\mathcal{A}_\gamma$ with size $d = [\gamma n] < n$.

The Sure Independence Screening, although works well in linear models, is limited on the assumption that the variables are assumed to be linearly correlated to the response. However, this is not always the case, since even if the true model is linear, the marginal correlation can be highly nonlinear. To overcome this problem, Fan *et al.* [12] proposed the Nonparametric

Independence Screening (NIS) in which the screening utility is built nonparametrically. Suppose we have a nonparametric model $Y = \boldsymbol{m}(\boldsymbol{X}) + \varepsilon$, in which $\boldsymbol{m}(\boldsymbol{X}) = \sum_{j=1}^{p} m_j(X_j)$ is some unknown additive structure which is not necessary linear and ε is some unknown random error. Consider the marginal nonparametric regression

$$\min_{f_j \in L_2(P)} \mathrm{E}(Y - f_j(X_j))^2, \tag{1.3}$$

where P denotes the joint distribution of $(\boldsymbol{X}, Y)$ and $L_2(P)$ denotes the class of square integrable functions under measure P. The screening utility is constructed as $\mathrm{E}f_j^2(X_j)$, where $f_j = \mathrm{E}(Y|X_j)$ is the minimizer of the above regression problem. To implement, f_j is approximated by B-splines with $f_j = \boldsymbol{\pi}(t)^T\boldsymbol{\beta}$, where $\boldsymbol{\pi}(t) = (B_1(t), \ldots, B_N(t))^T$ is B-spline basis function. The sample estimate of f_j is

$$\widehat{f}_{nj} = \boldsymbol{\pi}(X_{ij})^T \widehat{\boldsymbol{\beta}}_j, \tag{1.4}$$

where $\widehat{\boldsymbol{\beta}}_j = \underset{\boldsymbol{\beta} \in R^N}{\arg\min} \sum_{i=1}^{n} (Y_i - \boldsymbol{\pi}(X_{ij})^T \boldsymbol{\beta})$. Given a threshold ν_n, the selected variables are defined by applying the screening utilities $\|\widehat{f}_{nj}\|_n^2 = n^{-1} \sum_{i=1}^{n} \widehat{f}_{nj}^{\,2}$, $j = 1, \ldots, p$ as follows:

$$\widehat{\mathcal{A}} = \left\{ 1 \leq j \leq p : \|\widehat{f}_{nj}\|_n^2 \geq \nu_n \right\}.$$

The Sure Independence Ranking and Screening approach (SIRS) introduced by Zhu *et al.* [56] is another popular variable screening method. Unlike SIS and NIS, SIRS is a model free method that does not impose any specific model structure. Formally, define the support of Y as Ψ_y and the conditional distribution of Y given $\boldsymbol{x}$ as $F(y|\boldsymbol{x}) = P(Y < y|\boldsymbol{x})$. Define the active and inactive set as

$$\mathcal{A} = \{ j : F(y|\boldsymbol{x}) \text{ functionally depends on } X_j \text{ for some } y \in \Psi_y \}$$

$$\mathcal{I} = \{j : F(y|\boldsymbol{x}) \text{ does not functionally depends on } X_j \text{ for any } y \in \Psi_y\}.$$

Under the assumption that $F(y|\boldsymbol{x})$ depends on $\boldsymbol{x}$ only through $\boldsymbol{\beta}^T\boldsymbol{x}_{\mathcal{A}}$ for some $p_1 \times K$ constant matrix $\boldsymbol{\beta}$, where $p_1 = |\mathcal{A}|$ is the number of active predictors and K is some positive integer which is usually less than or equal to 3. That is, $F(y|\boldsymbol{x}) = F_0(y|\beta^T\boldsymbol{x}_{\mathcal{A}})$, where $F_0(y|\beta^T\boldsymbol{x}_{\mathcal{A}})$ is an unknown distribution function for a given $\beta^T\boldsymbol{x}_{\mathcal{A}}$. Define $\Omega(y) = \mathrm{E}\{\boldsymbol{x}F(y|\boldsymbol{x})\}$, then followed by the law of iterated expectations we have

$$\Omega(y) = \mathrm{E}[\boldsymbol{x}\mathrm{E}\{\mathbf{1}(Y < y)|\boldsymbol{x}\}] = \mathrm{Cov}\{\boldsymbol{x}, \mathbf{1}(Y < y)\}. \tag{1.5}$$

Let $\Omega_j(y)$ be the j-th element of $\Omega(y)$ and define the population marginal screening utilities by

$$\omega_j = \mathrm{E}\left\{\Omega_j^2(y)\right\}, j = 1, \dots, p. \tag{1.6}$$

Estimators of $\omega_j, j = 1, \dots, p$ are:

$$\widetilde{\omega}_j = \frac{1}{n}\sum_{k=1}^{n}\left\{\frac{1}{n}\sum_{i=1}^{n} X_{ij}\mathbf{1}(Y_i < Y_k)\right\}^2, j = 1, \dots, p. \tag{1.7}$$

The ranking utility is derived by measuring a scaled version $\widehat{\omega}_j = n^3\widetilde{\omega}_j/n(n-1)(n-2)$. By ranking the magnitude of $\{\widehat{\omega}_j, j = 1, \dots, p\}$, the selected variables are defined as

$$\widehat{\mathcal{A}} = \{1 \le j \le p : \mathrm{Rank}(\widehat{\omega}_j) \le T_n\}, \tag{1.8}$$

where T_n is some threshold value we will discuss later on.

The screening methods such as SIS and NIS work well in homoscedasticity setting, whereas deteriorate when heteroscedasticity is present. He *et al.* [21] proposed the Quantile Adaptive Sure Independence Screening (QaSIS) framework to deal with heteroscedasticity. Benefited from quantile regression and by the fact that QaSIS allows the set of active variables to vary across different quantiles, QaSIS is capable of dealing with heteroscedasticity. Empirically they also demonstrated that QaSIS works well against heavy tailed error distribution. Under the assumption that

$$Y \text{ and } X_j \text{ are independent } \Longleftrightarrow Q_\tau(Y|X_j) - Q_\tau(Y) = 0,$$

they define the set of active variables as

$$\mathcal{A}_\tau = \{1 \leq j \leq p : Q_\tau(Y|\boldsymbol{X}) \text{ functionally depends on } X_j\}. \tag{1.9}$$

Consequently, a screening utility is built on the sample estimate of $Q_\tau(Y|X_j) - Q_\tau(Y)$, where $Q_\tau(Y|X_j) = \inf\{y : P(Y \leq y|X_j) \geq \tau\}$ is the τ-th conditional quantile of Y given X_j and $Q_\tau(Y) = \inf\{y : P(Y \leq y) \geq \tau\}$ is the τ-th unconditional quantile of Y. Consider the marginal quantile regression of Y on X_j and denote

$$f_j(X_j) = \arg\min_f \mathrm{E}[\rho_\tau(Y - f(X_j)) - \rho_\tau(Y)], \tag{1.10}$$

where $\rho_\tau(u) = u\{\tau - I(u < 0)\}$ is the quantile loss function. Similar with the NIS approach, they use B-spline approximation to approximate f_j by $\boldsymbol{\pi}(t)^T\boldsymbol{\beta}$ for some $\boldsymbol{\beta} \in R^N$, where $\boldsymbol{\pi}(t) = (B_1(t), \ldots, B_N(t))^T$ is B-spline basis function. Let $\widehat{\boldsymbol{\beta}}_j = \arg\min_{\boldsymbol{\beta} \in R^N} \sum_{i=1}^n \rho_\tau(Y_i - \boldsymbol{\pi}(X_{ij})^T\boldsymbol{\beta})$ and define the nonparametric estimator of $Q_\tau(Y|X_j) - Q_\tau(Y)$ as

$$\widehat{f}_{nj} = \boldsymbol{\pi}(X_{ij})^T\widehat{\boldsymbol{\beta}}_j - F_{Y,n}^{-1}(\tau), \tag{1.11}$$

where $F_{Y,n}^{-1}(\tau)$ is the τ-th sample quantile function based on $Y_1, \ldots, Y_n$. Consequently, by ranking on the magnitude of screening utilities $\|\widehat{f}_{nj}\|_n^2 = \frac{1}{n}\sum_{i=1}^n \widehat{f}_{nj}^{\,2}$, $j = 1, \ldots, p$, we can implement the independence screening by setting up an appropriate threshold value T_n. The selected variables are defined as

$$\widehat{\mathcal{A}} = \left\{1 \leq j \leq p : \mathrm{Rank}(\|\widehat{f}_{nj}\|_n^2) \leq T_n\right\}. \tag{1.12}$$

Shi [47] proposed to implement variable screening by combing quantile levels. By aggregating the information of multiple quantile levels, Shi's methods show more efficiency compared with QaSIS. Considering quantile levels $\{\tau_k \in (0,1), k = 1, \ldots, K\}$, Shi defines the Average Quantile Utility(AQU) as $f_j^{AQR} = K^{-1}\sum_{k=1}^{K}[Q_{\tau_k}(Y|X_j) - Q_{\tau_k}(Y)]$, which is expected to be close to zero if X_j is independent with Y with $Q_{\tau_k}(Y|X_j)$ and $Q_{\tau_k}(Y)$ being estimated the same way as in QaSIS. If X_j contributes to the quantiles of Y only at several quantile levels, it can also be captured by f_j^{AQR}. Similar with QaSIS, B-spline approximation is applied to estimate f_j^{AQR} by $\boldsymbol{\pi}(t)^T\boldsymbol{\beta}$ for some $\boldsymbol{\beta} \in R^N$, where $\widehat{\boldsymbol{\beta}}_j = \underset{\boldsymbol{\beta}\in R^N}{\arg\min} \sum_{i=1}^{n} \rho_\tau(Y_i - \boldsymbol{\pi}((X_{ij})^T\boldsymbol{\beta}))$. Define the nonparametric estimator of f_{nj}^{AQR} as

$$\widehat{f}_{nj}^{AQR} = \frac{1}{K}\sum_{k=1}^{K}\Big[\boldsymbol{\pi}(X_{ij})^T\widehat{\boldsymbol{\beta}}_j(\tau_k) + \widehat{b}_j(\tau_k) - F_{Y,n}^{-1}(\tau_k)\Big]. \tag{1.13}$$

Consequently, the screening is implemented by ranking the magnitude of screening utility

$$\|\widehat{f}_{nj}^{AQR}\|_n^2 = \frac{1}{n}\sum_{i=1}^{n}\Big[\frac{1}{K}\sum_{k=1}^{K}\Big(\boldsymbol{\pi}(X_{ij})^T\widehat{\boldsymbol{\beta}}_j(\tau_k) + \widehat{b}_j(\tau_k) - F_{Y,n}^{-1}(\tau_k)\Big)\Big]^2. \tag{1.14}$$

The selected variables are defined as

$$\widehat{\mathcal{A}} = \Big\{1 \leq j \leq p : \text{Rank}(\|\widehat{f}_{nj}^{AQR}\|_n^2) \leq T_n\Big\}. \tag{1.15}$$

To combine the information of multiple quantile levels, the composite quantile regression (CQR) [59] is also an option. It uses the assumption that some of the coefficients are constant across quantile levels in the model. Given the intercept term $b_j^{CQR}(\tau_k)$ and coefficient β_j^{CQR}, the composite quantile regression estimates them as follows

$$\left(\Big\{\widehat{b}_j^{CQR}(\tau_k)\Big\}, \widehat{\beta}_j^{CQR}\right) = \underset{\boldsymbol{b},\beta}{\arg\min}\sum_{k=1}^{K}\sum_{i=1}^{n}\rho_{\tau_k}(Y_i - \boldsymbol{\pi}(X_{ij})^T\boldsymbol{\beta} - b(\tau_k)). \tag{1.16}$$

Consequently, the Composite Quantile Utility (CQU) which is defined as

$f_j^{CQR} = \sum_{k=1}^K [Q_{\tau_k}(Y|X_j) - Q_{\tau_k}(Y)]$ is estimated by

$$\widehat{f}_{nj}^{CQR} = \sum_{k=1}^{K} \Big[\boldsymbol{\pi}(X_{ij})\widehat{\boldsymbol{\beta}}_j^{CQR} + \widehat{b}_j^{CQR}(\tau_k) - \widehat{F}_{Y,n}^{-1}(\tau_k)\Big]. \tag{1.17}$$

The screening is implemented by ranking the magnitude of screening utility

$$\|\widehat{f}_{nj}^{CQR}\|_n^2 = \frac{1}{n}\sum_{i=1}^{n}\Big[\sum_{k=1}^{K}\Big(\boldsymbol{\pi}(X_{ij})\widehat{\boldsymbol{\beta}}_j^{CQR} + \widehat{b}_j^{CQR}(\tau_k) - \widehat{F}_{Y,n}^{-1}(\tau_k)\Big)\Big]^2. \tag{1.18}$$

As a result, the selected variables are defined as

$$\widehat{\mathcal{A}} = \Big\{1 \leq j \leq p : \mathrm{Rank}(\|\widehat{f}_{nj}^{CQR}\|_n^2) \leq T_n\Big\}. \tag{1.19}$$

In addition to the AQU and CQU, Shi also proposed the corresponding weighted version of the above two methods which are WAQU and WCQU respectively.

Besides the variable screening approaches introduced, there are many screening methods available in the literature. For nonparametric models, Fan *et al.* [15] extended ISIS, without explicit definition of residuals, to a general pseudo-likelihood framework. Li *et al.* [31] developed a sure independence screening procedure based on the distance correlation under more general settings to carry out marginal screening (DC-SIS). Shao *et al.* [46] proposed a martingale difference correlation variable screening method (MDC-SIS). Kong *et al.* [27] proposed a screening procedure by ranking the canonical correlation between the response and all possible sets of k variables which makes this method more efficient in keeping variables who have joint correlations with the response. Kong *et al.* [28] developed a variable screening approach based on distance correlation. Fan *et al.* [16] proposed a sure independent screening procedure based on Pearson correlation (P-IT). Pan *et al.* [41] proposed a generic nonparametric sure independence screening procedure on Ball correlation (BCor-SIS). Ma and Zhang [36] proposed a sure independence screening procedure based on quantile correlation [30] (QC-SIS). Ma *et al.* [35] used a quantile partial correlation criterion to measure the association of each predictor with the response at a specific quantile level and introduced a screening procedure by using the sample quantile partial correlation (QPCOR).

1.2 Ensemble Methods

In the machine learning field, ensemble methods are family of learning algorithms that construct a set of learners then make predictions by taking a combination. Instead of using ordinary learning approaches that generate only one learner from the training data, ensemble methods manage to construct a set of learners and combine them [10, 55]. By training multiple learners and aggregate them together, ensemble methods could be significantly more accurate than a single base learning approach. By combining several different models, the combined model may have better performance than any of the individual model alone, provided each model has a better performance than random guessing and with some level of diversity. By considering the types of base learners in an ensemble model, ensemble methods could be classified as homogeneous ensemble when the base learners are generated by the same learning algorithm and heterogeneous ensemble when the base learners are generated by different base learning algorithms. In the following, we introduce some ensemble methods with the application in statistics.

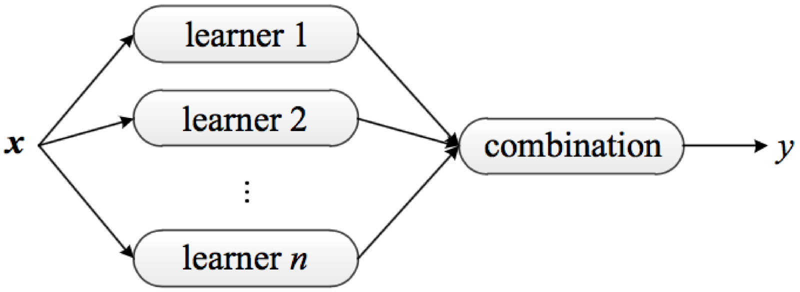
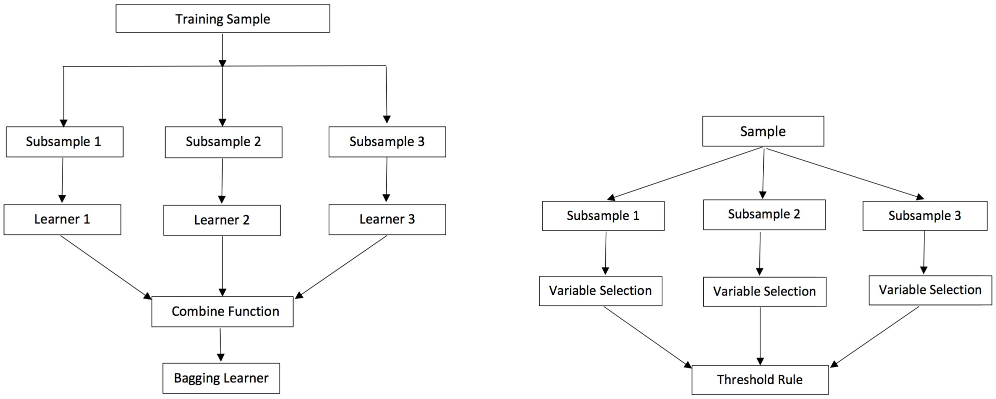


Figure 1.2 *A common ensemble architecture [55]: By using number of n learners on the data $\boldsymbol{x}$, ensemble methods aggregate the results generated from those learners through some combination functions.*

Bagging [3] is a widely adopted ensemble method with the base learners generated in parallel. In classification for example, instead of training one single classifier on the whole training set, we could draw subsamples from the training set and train multiple classifiers on each of the subsamples. Similar with subsampling, an alternative approach is bootstrap sampling [11] when the subsamples which have the same size of the original data are derived by random drawing from the original data with replacement.



(a) A general bagging procedure

(b) Stability selection procedure

Figure 1.3: *Comparison of bagging and stability selection: (a) First, multiple subsamples or bootstrap samples are generated from the sample. Second, multiple learners are trained on those subsamples separately. Last, a combination function is implemented to produce the bagging learner. (b) Similar with bagging, stability selection apply a variable selection procedure on multiple subsamples. Then by setting up a threshold π_{thr}, the selected variables are defined as those variables that have selection probability greater than π_{thr}.*

One example of bagging in variable selection problem could be found in stability selection [37]. When facing high dimensional data, variable selection stability is always a problem. In reality we are more concerned about the finite sample performance. In order to get a finite sample familywise error control and an improved structure estimation, Meinshausen and Buhlmann [37] proposed the stability selection method which is based on subsampling in combination with selection algorithms such as LASSO and SCAD. In detail, they look at the probability of each variable selected

$$\widehat{\Pi}_K^\lambda = P\left\{K \subseteq \widehat{S}^\lambda(A)\right\}, \tag{1.20}$$

where A is a random subsample of $\{1,\ldots,n\}$, $K \subseteq \{1,\ldots,p\}$, λ is the regularization parameter and the subsample size of $\lfloor n/2 \rfloor$ is selected as it resembles most closely to the bootstrap [17]. Furthermore, we can set a cut-off threshold π_{thr} with $0 < \pi_{thr} < 1$ and the set of stable variables is defined as

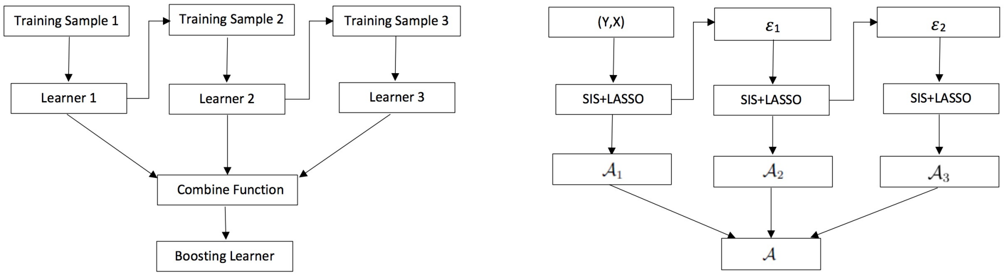
$$\widehat{S}^{stable} = \left\{ K : \max_{\lambda \in \Lambda} (\widehat{\Pi}_K^\lambda) \geq \pi_{thr} \right\}, \tag{1.21}$$

where Λ is a set of regularization parameters. Given exchangeability assumptions on the underlying model and the original selection procedure is not worse than random guessing, we have an upper bound for the number of falsely selected variables

$$\mathrm{E}(V) \leq \frac{1}{2\pi_{thr} - 1} \frac{q_\Lambda^2}{p}, \tag{1.22}$$

where V is the number of falsely selected variables, q_Λ is the average number of selected variables and p is the number of variables.

Instead of combining base learners in parallel to construct a better learner such as bagging, boosting [4, 5, 43] is a family of algorithms that uses iterative approach to convert weak learners to strong learners. In a binary classification approach where we are trying to classify an instance as positive or negative, a general boosting approach could be illustrated by the following example [55]: Given the training sample drawn from a distribution D and the ground truth function f. Suppose the space X is composed of three parts X_1, X_2 and X_3 each taking 1/3 amount of the distribution. Unluckily, we only have a weak learner that can only make correct classifications in spaces X_1 and X_2 but has wrong classification in X_3. Consequently, our learner (learner 1) at hand only has a classification rate of 1/3. The idea of boosting is to make the algorithm concentrates more on the mistakes made by learner 1 and thus improve the learning algorithm. In the next step, we derive a new distribution D' from D which makes the mistakes made by learner 1 more evident thus making the algorithm focuses more on the instances in X_3. Then we can train a new learner 2 from D'. Consequently, learner 2 will probably have better results on X_1 and X_3 but unsatisfying result on X_2. This procedure can be repeated multiple times till certain stopping criterion is reached. By appropriately combing all the learners via some combination function, the combined learner may have correct classifications in X_1 and maybe some errors in X_2 and X_3 thus leading to a lower classification error in total.



(a) A general boosting procedure

(b) ISIS procedure

Figure 1.4: *Comparison of boosting and ISIS: (a) In step one, learner 1 is trained on sample 1 which is the original sample. In step two, sample 2 is adjusted based on the deficiencies made by learner 1 to highlight the deficiencies. Learner 2 is then trained on sample 2. The procedure can be repeated multiple times till stopping criterion is reached. Last, by combining the learners, a boosting learner is generated. (b) First, a two-scale procedure is implemented on the sample to selecte $\mathcal{A}_1$. Second, regress Y over $\mathcal{A}_1$ to get residual $\boldsymbol{\varepsilon_1}$ and treat $\boldsymbol{\varepsilon_1}$ as the new response. Another two-scale procedure is applied and $\mathcal{A}_2$ is derived. The procedure can be repeated multiple times till the size or the union $\mathcal{A} = \cup\mathcal{A}_i$ reach a predefined threshold.*

The approach of Iterative Sure Independence Screeing (ISIS) [14] is close to boosting. The SIS works well when all the model assumptions are satisfied. However, some potential issues will deteriorate the performance of SIS. First, when some unimportant predictors are highly correlated with the important predictors, those unimportant ones will usually have high priority to be survived after applying SIS, since SIS only looks at the marginal correlations of the predictors and response. Secondly, when some important predictors are marginally uncorrelated but are jointly correlated with the response, SIS will tend to neglect these variables. Thirdly, collinearity of predictors is still an issue and adds difficulty to the variable screening problem. In order to overcome these issues, especially the first two, Fan and Lv [14] proposed the Iterative Sure Independence Screening (ISIS), which is a stepwise screening procedure that guarantees us getting a size $d < n$ subset in the screening process. More specifically, ISIS works as follows: In the first step, we apply SIS on the data set to get the selected variables. Then followed by a variable selection method, a subset of variables $\mathcal{A}_1 = \{X_{11}, \ldots, X_{1k_1}\}$ is selected. In the second step, we regress Y over $X_{11}, \ldots, X_{1k_1}$ to get the residuals $\boldsymbol{\varepsilon_1}$. In step three, we treat $\boldsymbol{\varepsilon_1}$ as the new response, then the same approach could be applied to the remaining $p - k_1$ variables resulting in a subset of k_2 variables $\mathcal{A}_2 = \{X_{21}, \ldots, X_{2k_2}\}$. We could repeat the procedure until we get l disjoint subsets $\mathcal{A}_1, \ldots, \mathcal{A}_l$ whose union $\mathcal{A} = \cup_{i=1}^{l} \mathcal{A}_i$ reaches a predefined threshold value. By fitting the residuals from the previous step, we can significantly weaken the priority of the unimportant variables that are highly correlated with the response through their associations with $X_{i1}, \ldots, X_{ik_i}$. Also, we can make some important variables that are missed in the previous step have higher probability to enter the model. Similar approach could be found in Fan *et al.* [12] in which they use the same idea to add an iterative procedure to the NIS approach.

1.3 Contributions

Variable screening plays an important role in the two-scale approach [14] to deal with ultrahigh dimensional data. It is crucial that all the important variables are kept by the screening method applied in the first step. There are many variable screening methods that work well with various assumptions. However, two issues are associated with the existing variable screening methods. First, given finite sample, the stabilities of variable screening methods will be deteriorated when some perturbations are injected to the sample. Second, it is difficult to verify all the assumptions in reality which means we lack of generic guideline to select an appropriate screening method. A natural choice is to combine multiple screening methods to adapt more general assumptions. Motivated by the widely application of ensemble methods in machine learning, it is interesting to see if the ensemble methods are capable of dealing with the issues associated with variable screening approaches. Instead of choosing one specific variable screening algorithm, we suggest to combine the information generated from multiple screening algorithms to get a more stable and precise result.

In particular, the contributions of this thesis are multifold: First, we introduce five ensemble functions to combine the screening methods, namely, the mean, median, rank mean, rank median and mean voting ensemble. Second, we develop three ensemble approaches which are parallel ensemble screening, quantile ensemble screening and multi ensemble screening. In addition, we combine the three approaches to produce a mixture ensemble screening. Third, we provide the sure screening property of the ensemble screening approaches. Last, simulation studies are conducted for our proposed methods. We also apply the proposed methods to the ADHD-200 data set. The rest of this thesis is organized as follows. In Chapter 2, we introduce our proposed methods including parallel, quantile, multi and mixture ensemble screening methods. In Chapter 3, the simulation studies are conducted as well as the real data analysis. The last chapter is the summary and future work of my research.

Chapter 2

Ensemble Based Ultrahigh Dimensional Variable Screening

In this chapter, we introduce our ensemble based ultrahigh dimensional variable screening methods. First we introduce some preliminaries to proceed our ensemble screening methods. We adopt five different ensemble functions to aggregate the results of different screening algorithms. Second, we propose our three main approaches to generate parallel or heterogeneous screening algorithms, we then propose to combine all of the three approaches to improve the performance. Last, we show some technical results for the sure screening property and error control.

2.1 Preliminaries

In the following, we consider a random sample $\{\boldsymbol{X_i}, Y_i\}_{i=1}^{n}$ from the population $(\boldsymbol{X}, Y)$ in which Y is the response and $\boldsymbol{X} = (X_1, \ldots, X_p)$ is the associated covariate vector. Without loss of generality, we assume $\boldsymbol{X}$ are standardized column-wisely. We first define active variables in our ensemble screening approach. The definitions of active variables are slightly inconsistent between current available screening approaches. Among those definitions, the most commonly adopted one is

$$\mathcal{A} = \{j : F(y|\boldsymbol{x}) = P(Y < y|\boldsymbol{x}) \text{ functionally depends on } X_j\}$$

$$\mathcal{I} = \{j : F(y|\boldsymbol{x}) = P(Y < y|\boldsymbol{x}) \text{ does not functionally depends on } X_j\},$$

where $F(y|\boldsymbol{x}) = P(Y < y|\boldsymbol{x})$ is the conditional distribution function of Y given $\boldsymbol{x}$. If $j \in \mathcal{A}$, X_j is considered as an active predictor. Otherwise, X_j will be considered as an inactive predictor. In quantile based screening approaches, QaSIS for example, the j-th variable is considered as active: $j \in \mathcal{A} \iff Q_\tau(Y|X_j) - Q_\tau(Y) \neq 0$. Otherwise, when j-th variable is an inactive predictor: $j \in \mathcal{I} \iff Q_\tau(Y|X_j) - Q_\tau(Y) = 0$. In our ensemble approach, if X_j is considered active in any of the candidate screening methods in the ensemble, it is an active variable in our ensemble screening method. Otherwise, X_j is considered as inactive. Intuitively speaking, when X_j is considered as an active predictor, any change in the information of X_j should change the information of the response Y.

In the following we introduce the ensemble functions we employ to aggregate the results. By applying number of K screening algorithms on a sample of $n \times p$, a $K \times p$ matrix W with entries w_{ij} where $i = 1, \dots, K$ and $j = 1, \dots, p$ representing the corresponding screening utilities are derived. In the next step, we find appropriate ensemble functions to aggregate the results. A general ensemble function is a multivariate function that projects the screening utilities in the j-th column of W to a real number, that is,

$$f(w_{1j}, \dots, w_{Kj}) \to \mathbb{R}.$$

Mean Ensemble: Taking means of prediction results is a commonly adopted approach in machine learning literature [40, 55]. In our approach, mean ensemble combines the output of the screening algorithms by taking the mean of each column in W. Provided the screening utilities associated with the j-th column as $w_{1j}, \ldots, w_{Kj}$, the mean ensemble function is defined as

$$f(w_{1j}, \ldots, w_{Kj}) = \frac{1}{K} \sum_{i=1}^{K} w_{ij}, j = 1, \ldots, p. \tag{2.1}$$

Median Ensemble: Instead of choosing the mean of $w_{1j}, \ldots, w_{Kj}$, we use the median [40, 53], which is more robust when those screening utilities are skewed or with outliers. Formally we have

$$f(w_{1j}, \ldots, w_{Kj}) = \text{Median}\left\{w_{1j}, \ldots, w_{Kj}\right\}, j = 1, \ldots, p. \tag{2.2}$$

Rank Mean Ensemble: Besides the magnitude of the original screening statistics, the ranks of the screening utilities are useful benchmarks for variable screening [14, 21, 56]. More specifically, the ranks of the variables are derived by sorting the screening utilities $\{w_j, j = 1, \ldots, p\}$, denoted as $\{r_j, j = 1, \ldots, p\}$. Therefore, the mean of ranks for the j-th variable is defined as

$$f(w_{1j}, \ldots, w_{Kj}) = \frac{1}{K} \sum_{i=1}^{K} r_{ij}, j = 1, \ldots, p. \tag{2.3}$$

We remark that when the "rank" operation is implemented on the screening utilities obtained by the screening algorithms directly, "rank" operation means sorting the magnitude of the screening utilities from largest to smallest and the largest screening statistics will be assigned a rank of 1. However, when the "rank" operation is implemented on a sequence of aggregated ranks, the "rank" operation will have an opposite result. For example, a rank series of $1.2, 5.4, 3.5, 2, 4.6$ after "rank" operation is $1, 5, 3, 2, 4$.

Rank Median Ensemble: We could also choose the median instead of the mean of ranks. The median of ranks for the j-th variable is defined as

$$f(w_{1j}, \ldots, w_{Kj}) = \text{Median}\left\{r_{1j}, \ldots, r_{Kj}\right\}, j = 1, \ldots, p. \tag{2.4}$$

Mean Voting Ensemble: Voting [6, 10, 40, 55] is also a commonly adopted ensemble function. Specifically, voting is implemented by specifying an binary indicator

$$b_j = \begin{cases} 1, & \text{if } j\text{-th variable is selected} \\ 0, & \text{if } j\text{-th variable is not selected} \end{cases}. \tag{2.5}$$

Given a threshold T_n, after we have applied K screening algorithms to the data, we can construct a matrix containing all the $K \times p$ binary votes. Denoting the votes made by the i-th screening algorithm and the j-th variable as b_{ij}, the mean of the votes for the j-th variable is

$$f(w_{1j}, \ldots, w_{Kj}) = \frac{1}{K} \sum_{i=1}^{K} b_{ij}, j = 1, \ldots, p. \tag{2.6}$$

We remark that to apply the natural extension of the mean voting ensemble to median may not be a good choice. For example, in high dimensional variable screening setting especially when the marginal signals are not strong enough, the votes in each column will end up with zero frequently which makes the median to be zero with high probability. Consequently, median ensemble fails to work in the voting scenario and we will not use the median voting ensemble in our framework. In some machine learning approaches, using min or max function [40] to aggregate the results also plays an important role. Another example could be found in pooling [45] technique in convolutional neural network (CNN) [29] where max pooling is applied to implement dimension reduction and to evade overfitting. In this thesis, we will not discuss the application of min or max ensemble in variable screening as they only use tail information hence not working well in variable screening scenario.

The choice of threshold T_n is important since it not only affects the result of our screening algorithm but also determines the generation of the votes. In the variable screening studies, there are two different threshold rules. The most widely adopted one is the hard threshold rule proposed by Fan and Lv [14]. The hard threshold rule sets a fixed number $\lfloor n/\log(n) \rfloor$ for the number of selected variables which only related to the sample size. Considering a sample of size 200, the hard threshold will specify a subset of size 37 which is large enough to guarantee all the active variables to be selected. Consequently, as proposed by Fan and Lv [14], the variable selection step of the two-scale procedure could be applied on this subset to select the truly important variables. On the other hand, the soft threshold introduced by Luo *et al.* [34] is a data driven approach. By adding random noise variables to the original data set, the soft threshold is derived by finding the largest screening utility or its corresponding rank among all the random noise variables. Specifically, we generate number of d noise variables $\boldsymbol{Z}$ following $N_d(0, \boldsymbol{I}_d)$ which is independent with both $\boldsymbol{X}$ and Y. In the screening process, we treat $(\boldsymbol{X}, \boldsymbol{Z})$ as the predictors and Y as the response. The reasoning is, since $\boldsymbol{Z}$ is generated randomly, the active variables in $\boldsymbol{X}$ should always rank in front of those noise variables. In this thesis, without special notification, we adopt the hard threshold rule. We choose the hard threshold not only because it requires less computational time but also because when applying all the combinations, hard threshold gives us a consistent benchmark. Further more, in the following sections we denote $T_n = \lfloor n/\log(n) \rfloor$ as our hard threshold.

2.2 Ensemble Screening Methods

Parallel Ensemble Screening: We adopt the technique in bagging [3] to implement our ensemble screening. More specifically, we draw random subsamples or bootstrap samples number of K times from the original data set A, leading to a series of data sets $A_1, \ldots, A_K$. The subsampling or bootstrap sampling approach could be considered as manipulating the training examples in machine learning [10]. We remark here when the subsample size is chosen by $\lfloor n/2 \rfloor$ in the subsampling approach, it resembles most closely to the bootstrap [17, 37]. By applying the screening algorithm such as SIS on each of the subsample or bootstrap sample, we can derive a matrix W of $K \times p$ where the i-th row contains the screening utilities corresponding to the variables of i-th subsample or bootstrap sample A_i. Consequently, all the five ensemble functions could be applied to this matrix to get the combined variable screening result. Denote the result by applying the ensemble functions column-wise to W as $\{R_1, \ldots, R_p\}$. By setting up the threshold value T_n and obtaining the corresponding ranks of the combined utilities $\{R_1, \ldots, R_p\}$, the selected variables are defined as

$$\widehat{\mathcal{A}} = \{1 \le j \le p : \mathrm{Rank}(R_j) \le T_n\}\,. \tag{2.7}$$

Quantile Ensemble Screening: The Quantile Adaptive Sure Independence Screening (QaSIS) [21], although provides us a powerful tool against heavy-tailed distribution or heteroscedasticity, only considered one specific quantile level and may miss some variables when the active variables differ in changing the quantile level. More specifically speaking, at a certain quantile level τ, we consider the following set of active variables

$$\mathcal{A}_\tau = \{1 \le j \le p : Q_\tau(Y|X) \text{ functionally depends on } X_j\}\,. \tag{2.8}$$

However, the active variables derived by considering only one quantile level τ are only a subset of $\mathcal{A}$, which is not desirable since when the sample size is limited, variability will arise in the set of selected variables as τ changes. Therefore, we may miss some variables simply because we did not consider a specific quantile level. For the pursue of interpretation, such variability is not preferable either, because we need a consistent variable set to build our model. Instead of using one specific quantile level, we adopt heterogeneous ensemble by combining different quantile levels. In this way, we can not only employ the information of multiple quantile levels, but also make each screening algorithm more diverse which usually promotes the performance of the combined learners in machine learning. To formalize, considering a sequence of quantile levels $\tau_1, \ldots, \tau_T$, for each quantile level we could apply QaSIS or any other quantile based variable screening approaches to the whole sample and derive a vector of screening utilities $\{q_{i1}, \ldots, q_{ip}, i = 1, \ldots, T\}$. Combining all the vectors horizontally, a matrix Q is derived with entries q_{ij} representing the screening utilities of i-th quantile level and j-th variable. Similarly, denote the result by applying the ensemble functions to Q as $\{R_1, \ldots, R_p\}$. By setting up the threshold value T_n and obtaining the corresponding ranks of the combined utilities $\{R_1, \ldots, R_p\}$, the selected variables are defined as

$$\widehat{\mathcal{A}} = \{1 \le j \le p : \mathrm{Rank}(R_j) \le T_n\}\,. \tag{2.9}$$

Multi Ensemble Screening: In ultrahigh dimensional variable screening, we often do not have enough prior knowledge about the data structures and the model specification. To our best knowledge, if the data structure is linear, SIS or NIS will have a better performance. However, when the data structure is with heteroscedasticity or heavy tail, QaSIS or SIRS tends to have a stronger performance. In general, current variable screening methods behave quite differently based on different simulation settings or real data sets. In gene expression microarray data analysis for example, the result of applying different variable screening algorithms will probably be different. Given limited prior knowledge, it is difficult to tell which result is the most reliable. If we misuse the variable screening method which eliminates some of the underlying biomarkers, then it is impossible for us to make further discoveries. A natural idea would be combining the advantage of multiple variable screening algorithms which will lead to a safe situation. To implement, suppose we have K different screening methods and by applying them to the sample, we get a matrix M containing the screening utilities m_{ij}, where i denotes the i-th method, and j denotes the j-th variable. As the screening utilities of different methods may have very different magnitudes, we can not directly combine them by simply using the mean or median ensemble. Otherwise some screening methods that produce relative large screening utilities will tend to dominate the combined result. To tackle this issue, only the rank mean, rank median and mean voting ensemble are applied. Denote the result by applying the ensemble functions column-wise to matrix M as $\{R_1, \ldots, R_p\}$. Then by setting up the threshold T_n and obtaining the corresponding ranks of the combined utilities $\{R_1, \ldots, R_p\}$, the selected variables are

$$\widehat{\mathcal{A}} = \{1 \le j \le p : \mathrm{Rank}(R_j) \le T_n\}\,. \tag{2.10}$$

Mixture Ensemble Screening: We have shown the parallel ensemble screening, quantile ensemble screening and multi ensemble screening methods. Then it would be promising to combine them to produce a mixture screening method. We propose to apply quantile ensemble to quantile based methods and parallel ensemble to non-quantile based methods. In detail, suppose we have K screening algorithms and K_1 of them are quantile based, for example QaSIS, and the other $K_2 = K - K_1$ are not. For the i-th of the K_1 quantile based methods we apply our quantile based screening method with quantile levels $\{\tau_1, \ldots, \tau_T\}$, which results in a matrix Q_i as in quantile ensemble screening. By applying the ensemble functions, vectors $\left\{R_{i_1}, \ldots, R_{i_p}, i = 1, \ldots, K_1\right\}$ containing the ensembled information of Q_i are derived. For the k-th of the K_2 non-quantile based methods, for example SIRS, we draw B random subsamples or bootstrap samples $A_1, \ldots, A_B$ from the sample and apply the corresponding screening algorithm separately on each of the subsample or bootstrap sample, which results in a matrix W_k as in parallel ensemble screening. By applying the ensemble functions, we then end up with vectors $\left\{R_{k_1}, \ldots, R_{k_p}, k = K_1 + 1, \ldots, K\right\}$ containing the ensembled information of W_k. Consequently, by combining $\left\{R_{i_1}, \ldots, R_{i_p}, i = 1, \ldots, K_1\right\}$ and $\left\{R_{k_1}, \ldots, R_{k_p}, k = K_1 + 1, \ldots, K\right\}$ horizontally, a matrix M containing the aggregated results is constructed. Denote $\{R_1, \ldots, R_p\}$ as the combined information by applying the ensemble functions column-wise to matrix M. By setting up a threshold T_n and obtaining the corresponding ranks of the combined utilities $\{R_1, \ldots, R_p\}$, the selected variables are

$$\widehat{\mathcal{A}} = \{1 \le j \le p : \text{Rank}(R_j) \le T_n\}. \tag{2.11}$$

2.3 Technical Results

As variable screening only serves as the first step in high dimensional data analysis, the most important property as far as practical application concerned is the sure screening property. That is, with probability approaching 1, the screening algorithm keeps all of the true active variables. Regarding the screening utilities, we require all the screening algorithms in the ensemble enjoy consistency of the screening utilities, that is, $P(\max_{1\leq j\leq p}|\widehat{w}_{ij} - w_{ij}| > \delta_n) \to 0$ for all $1 \leq i \leq K$, where w_{ij} is the corresponding screening utility for screening algorithm i and variable j, δ_n is some threshold number that is usually related to n. We argue that this requirement is reasonable as it is shown in most variable screening literatures [12, 21, 56]. In addition we require that the sure screening property holds for each screening algorithms. For the quantile based variable screening algorithms, the QaSIS for example, we require the sure screening property holds at each quantile level τ_k which means the selected variables contain the true active set $\mathcal{A}_k$ with a probability tending to one.

Lemma 2.3.1. *(Consistency of aggregated screening utilities) Given number of K screening algorithms which are based on screening utilities $w_{ij}, i = 1, \ldots, K$ and $j = 1, \ldots, p$. Denote f as the ensemble function. Assume the following:*

$$P(\max_{1\leq j\leq p}|\widehat{w}_{ij} - w_{ij}| > \delta_n) \to 0$$

$$|f(\widehat{w}_{1j}, \ldots, \widehat{w}_{Kj}) - f(w_{1j}, \ldots, w_{Kj})| \leq \max_{1\leq i\leq K}|\widehat{w}_{ij} - w_{ij}|,$$

in which δ_n is some threshold constant related to n. We have the consistency of the aggregated screening utility which is:

$$P(\max_{1\leq j\leq p}|f(\widehat{w}_{1j}, \ldots, \widehat{w}_{Kj}) - f(w_{1j}, \ldots, w_{Kj})| > \delta_n) \to 0.$$

Proof. Given $|f(\widehat{w}_{1j}, \ldots, \widehat{w}_{Kj}) - f(w_{1j}, \ldots, w_{Kj})| \leq \max_{1\leq i\leq K}|\widehat{w}_{ij} - w_{ij}|$, for a fixed δ_n, we have

$$P(\max_{1\leq j\leq p}|f(\widehat{w}_{1j}, \ldots, \widehat{w}_{Kj}) - f(w_{1j}, \ldots, w_{Kj})| > \delta_n)$$

$$< P(\max_{1\leq j\leq p}|\max_{1\leq i\leq K}|\widehat{w}_{ij} - w_{ij}|| > \delta_n).$$

As for all $i, 1 \leq i \leq K$ we have:

$$P(\max_{1\leq j\leq p}|\widehat{w}_{ij} - w_{ij}| > \delta_n) \to 0.$$

Hence,

$$P(\max_{1\le j\le p} |f(\widehat{w}_{1j},\ldots,\widehat{w}_{Kj}) - f(w_{1j},\ldots,w_{Kj})| > \delta_n) \to 0.$$

□

Theorem 2.3.2. *(Sure screening property) Denote w_j as the j-th aggregated screening utility, and w_j^* as the sample estimate. Denote $\mathcal{A}$ as the active variable set. Assume $\min\limits_{j\in\mathcal{A}} w_j > 2\delta_n$ and our screening method select the variables with $w_j^* > \delta_n$. Given the assumptions of the previous lemma, we have the sure screening property for our ensemble screening approach:*

$$P(\mathcal{A} \subseteq \widehat{\mathcal{A}}) \to 1,$$

in which δ_n is some threshold number related to n.

Proof. Suppose $\mathcal{A} \nsubseteq \widehat{\mathcal{A}}$, then there must be some $j \in \mathcal{A}$ such that $w_j^* \le \delta_n$. As $\min\limits_{j\in\mathcal{A}} w_j > 2\delta_n$, there must be some $j \in \mathcal{A}$ such that $|w_j^* - w_j| > \delta_n$. We denote those j as J^*. Denoting $W = \left\{j \in J^* : \mathcal{A} \nsubseteq \widehat{\mathcal{A}}\right\}$ and $M = \left\{j \in J^* : |w_j^* - w_j| > \delta_n\right\}$, hence the above statement is equivalent to

$$W \Rightarrow M.$$

Hence,

$$W^C \supseteq M^C.$$

Therefore,

$$\eta = M^C = \left\{\max_{j\in J^*} |w_j^* - w_j| \le \delta_n\right\} \subseteq \left\{j \in J^* : \mathcal{A} \subseteq \widehat{\mathcal{A}}\right\}.$$

Consequently,

$$\begin{aligned} &P\left\{j \in J^* : \mathcal{A} \subseteq \widehat{\mathcal{A}}\right\} \ge P(\eta) \\ =&1 - P(\eta^C) \\ =&1 - P(\left\{j \in J^* : |w_j^* - w_j| > \delta_n\right\}). \end{aligned}$$

For each j, we have: $P(|w_j^* - w_j| > \delta_n) \to 0$ as was shown in Lemma 2.3.1. Finally, as J^* is a finite set, we have

$$P(\mathcal{A} \subseteq \widehat{\mathcal{A}}) \to 1.$$

□

Remark. *For different variable screening methods, the threshold δ_n may differ. For example, QaSIS requires δ_n to be $c_1 n^{-\tau}/8$, where $c_1 > 0$ is some positive constant and $0 \le \tau < 2d/(2d+1)$ with $d > 0.5$. In SIRS procedure, they only require δ_n to be any small enough number ε. To guarantee the sure screening property holds for our ensemble procedure, we just need δ_n to be the smallest among all the thresholds we combined. Given n large enough, the magnitude of δ_n could be a sufficient small number.*

Lemma 2.3.3. *(Lower bound for simultaneous screening probability) Denote $\widehat{\mathcal{T}}_1$ and $\widehat{\mathcal{T}}_2$ as two selected sets by applying two different screening algorithms. Define $\widehat{\mathcal{T}}^{simu} = \widehat{\mathcal{T}}_1 \cap \widehat{\mathcal{T}}_2$ as the variable set selected by both of the screening algorithms. Define $\widehat{\Pi}_K^1$, $\widehat{\Pi}_K^2$, $\widehat{\Pi}_K^{simu}$ as the probabilities of selected set of screening algorithm 1, screening algorithm 2 and the simultaneous set containing a variable set K, $K \subseteq \{1, \ldots, p\}$. Then we have:*

$$\widehat{\Pi}_K^{simu} \ge 2 \min \left\{ \widehat{\Pi}_K^1, \widehat{\Pi}_K^2 \right\} - 1.$$

Proof. Denote $s(1,1)$ as the probability of $P\Big[\Big\{ K \subseteq \widehat{\mathcal{T}}_1 \Big\}, \Big\{ K \subseteq \widehat{\mathcal{T}}_2 \Big\} \Big]$. Note that the two events are not independent as the probability is only with respect to the screening algorithm chosen. Correspondingly, the probabilities $s(0,1)$, $s(1,0)$ and $s(0,0)$ are defined as
$P\Big[\Big\{ K \nsubseteq \widehat{\mathcal{T}}_1 \Big\}, \Big\{ K \subseteq \widehat{\mathcal{T}}_2 \Big\} \Big]$, $P\Big[\Big\{ K \subseteq \widehat{\mathcal{T}}_1 \Big\}, \Big\{ K \nsubseteq \widehat{\mathcal{T}}_2 \Big\} \Big]$ and
$P\Big[\Big\{ K \nsubseteq \widehat{\mathcal{T}}_1 \Big\}, \Big\{ K \nsubseteq \widehat{\mathcal{T}}_2 \Big\} \Big]$ respectively. Note that:

$$\widehat{\Pi}_K^{simu} = s(1,1);$$

$$\widehat{\Pi}_K^1 = s(1,0) + s(1,1), 1 - \widehat{\Pi}_K^1 = s(0,1) + s(0,0);$$

$$\widehat{\Pi}_K^2 = s(0,1) + s(1,1), 1 - \widehat{\Pi}_K^2 = s(1,0) + s(0,0).$$

As $s(0,0) \geq 0$, we have:

$$s(0,1) \leq 1 - \widehat{\Pi}_K^1, s(1,0) \leq 1 - \widehat{\Pi}_K^2.$$

Hence we have:

$$s(1,1) \geq \widehat{\Pi}_K^1 + \widehat{\Pi}_K^2 - 1.$$

Hence, $\widehat{\Pi}_K^{simu} \geq 2\min\left\{\widehat{\Pi}_K^1, \widehat{\Pi}_K^2\right\} - 1$. □

Lemma 2.3.4. *Let $K \subset \{1,\ldots,p\}$ be a set of variables and $\widehat{\mathcal{T}}_i$ be the set of selected variables by applying a variable screening algorithm i. If $\max\left\{P(K \subseteq \widehat{\mathcal{T}}_1), P(K \subseteq \widehat{\mathcal{T}}_2)\right\} \leq \varepsilon$, then*

$$P(\widehat{\Pi}_K^{simu} \geq \xi) \leq \varepsilon^2/\xi.$$

Proof. Define $H = \mathbf{1}\left\{K \subseteq \left\{\widehat{\mathcal{T}}_1 \cap \widehat{\mathcal{T}}_2\right\}\right\}$ and denote the data by X. Then the simultaneous probability is $\widehat{\Pi}_K^{simu} = E(H|X)$.
Since $\max\left\{P(K \subseteq \widehat{\mathcal{T}}_1), P(K \subseteq \widehat{\mathcal{T}}_2)\right\} \leq \varepsilon$, we have

$$P(H = 1) \leq \max\left\{P(K \subseteq \widehat{\mathcal{T}}_1), P(K \subseteq \widehat{\mathcal{T}}_2)\right\}^2 \leq \varepsilon^2.$$

Therefore, $\mathrm{E}\left\{\mathrm{E}(H|X)\right\} = \mathrm{E}(\widehat{\Pi}_K^{simu}) \leq \varepsilon^2$. By using a Markov-type inequality, $\xi P(\widehat{\Pi}_K^{simu} \geq \xi) \leq \mathrm{E}(\widehat{\Pi}_K^{simu}) \leq \varepsilon^2$. The case for more than two screening algorithms follows analogously. □

Theorem 2.3.5. *(Error control) Denote $S = |\mathcal{T}|$ and $N = |\mathcal{F}|$ as the number of underlying true important and unimportant variables. Correspondingly, denote $\widehat{S} = |\mathcal{T} \cap \widehat{\mathcal{T}}^{simu}|$ and $\widehat{N} = |\mathcal{F} \cap \widehat{\mathcal{F}}^{simu}|$ as the number of estimated important and unimportant variables. In addition denote $V = \mathrm{E}(|\mathcal{F} \cap \widehat{\mathcal{T}}^{simu}|)$ as the expected number of falsely selected variables in $\widehat{\mathcal{T}}^{simu}$. Assume exchangeablity which is $P(k \in \widehat{\mathcal{T}}) = \mathrm{E}(\widehat{N})/N$, where $k \in (1,\ldots,p)$. Also assume that the candidate variable screening process is not worse than random guessing.*

Given the screening threshold T_n and the threshold of selection probability π_{thr}, we have:

$$\mathrm{E}(V) \leq \frac{1}{2\pi_{thr}-1}\frac{T_n^2}{p}.$$

Proof. The expected number of falsely selected variables can be expressed as $\mathrm{E}(\widehat{N}) = T_n - \mathrm{E}(\widehat{S})$. By the assumption that the original process is not worse than random guessing, we have

$$\mathrm{E}(\widehat{S}) \geq \mathrm{E}(\widehat{N})S/N.$$

Putting them together, we have

$$(1 + S/N)\mathrm{E}(\widehat{N}) \leq T_n.$$

Hence, $N^{-1}E(\widehat{N}) \leq T_n/p$. By applying the exchageablility assumption, we have

$$P(k \in \widehat{\mathcal{T}}) \leq T_n/p.$$

By applying Lemma 2.3.3 and Lemma 2.3.4, we have

$$P\Big(\min_i \widehat{\Pi}_k^i \geq \pi_{thr}\Big) \leq P\Big(\big[(\widehat{\Pi}_k^{simu} + 1)/2\big] \geq \pi_{thr}\Big),$$

which implies

$$P\Big(\widehat{\Pi}_k^{simu} \geq \pi_{thr}\Big) \leq \frac{1}{2\pi_{thr}-1}(\frac{T_n}{p})^2.$$

Therefore,

$$\mathrm{E}(V) = \sum_{k\in N} P\Big(\widehat{\Pi}_k^{simu} \geq \pi_{thr}\Big) \leq \frac{1}{2\pi_{thr}-1}\frac{T_n^2}{p}.$$

□

Remark. *If a variable is important, the underlying true ranking of the variable is higher than the other unimportant variables. With a threshold T_n, the true important variables are supposed to rank within T_n. If T_n is a relative small number say 30, and the probability threshold π_{thr} is decently greater than 50%, given $p = 1000$ the expectation number of falsely discovered variables could be* *controlled within a small number. Intuitively, when each candidate screening method is good enough and the threshold T_n is small, the number of falsely selected variables could be controlled at a very low level.*

Chapter 3

Numerical Studies

3.1 Simulation Studies

In order to assess the finite sample performance of our methods, for convenience, we make some notations as follows: (1) We denote the parallel ensemble screening results of SIS, NIS, SIRS and QaSIS as PSIS, PNIS, PSIRS and PQaSIS respectively. Regarding different ensemble functions, using PQaSIS for example, we use PQaSIS_mean, PQaSIS_median, PQaSIS_Rmean, PQaSIS_Rmedian and PQaSIS_Bmean to denote the mean, median, rank mean, rank median and mean voting ensemble respectively. (2) Regarding the quantile ensemble screening, we denote our method as QES. The mean, median, rank mean, rank median and mean voting ensemble functions are denoted same as in parallel ensemble screening. (3) For the multi ensemble screening, we denote it as MultiSIS. In this simulation study, we only use the rank mean and rank median ensemble functions which are denoted the same way as in parallel ensemble screening. (4) For the mixture ensemble screening, we denote it as MixSIS. Same with multi ensemble approach, we only use rank mean and rank median ensemble functions which are denoted the same way as in parallel ensemble screening.

We consider the following distributions for the error term ε:

1. the standard normal distribution;
2. student-t distribution with degree of freedom one, which is also known

as the Cauchy distribution;

3. mixture normal distribution: $0.9N(0,1) + 0.1N(10,1)$.

In the parallel and mixture ensemble screening approaches, we use bootstrap samples and the number of bootstrap samples is set to be 30. The quantile ensemble screening approach is implemented by combining 10 equally spaced quantile levels $\{0.05, 0.15, \ldots, 0.85, 0.95\}$. The multi ensemble screening is done by combining the results of NIS, QaSIS at quantile level 0.75 and SIRS. We choose NIS over SIS since in most of the settings, these two have similar performance but NIS is nonparametric and can handle nonlinear cases. As in ensemble methods, the performance will usually be better when the algorithms we combined show diversity. To compute QaSIS, NIS as well as our methods, the number of B-spline basis functions is set to be 5. Regarding the mixture ensemble screening, we also use NIS, QaSIS and SIRS as our candidate methods. In the first step, for QaSIS, we adopt quantile ensemble screening by combining the results on ten equally spaced quantile levels $\{0.05, 0.15, \ldots, 0.85, 0.95\}$. After getting the $10 \times p$ matrix containing the corresponding screening utilities, we employ our rank mean and rank median ensemble functions to aggregate the results. For the other two methods, we employ bootstrap sampling approach 30 times each and derive two $30 \times p$ matrices containing the screening utilities. Then we apply the same rank mean and rank median ensemble functions to aggregate the results. In the next step, we further apply the rank mean and rank median ensemble functions to combine the vectors containing the ensembled results of the first step.

In this study, we consider two criteria [56] for the evaluation of simulation models. The first criterion is $\mathcal{R}$ which is the minimum model size to contain all the true active predictors. The second criterion is the proportion of the active predictors being included in the model after screening procedure which is denoted by $\mathcal{S}$. In this simulation study, each method is repeated 100 times to exclude occasional bias. The median value of both $\mathcal{R}$ and $\mathcal{S}$ are reported in our results. We use IQR($\mathcal{R}$) and IQR($\mathcal{S}$) to illustrate the spread of $\mathcal{R}$ and $\mathcal{S}$ in the 100 runs. The number of true active variables is denoted as p^*. By looking at these two criteria, we would except an effective screening procedure tend to produce a reasonably small $\mathcal{R}$ that is as close to the total number of true activate variables as possible and a relative large $\mathcal{S}$ that is close to or equal to one. In addition, we would also expect an efficient screening algorithm to have small IQR($\mathcal{R}$) and IQR($\mathcal{S}$) to show stability. In the following we use boxplots to illustrate the results. The results of $\mathcal{R}$ showing in the boxplots are preprocessed to remove extreme values. The detailed tables are shown in appendices.

EXAMPLE 1: $(n = 200, p = 2000)$. This example is adopted from Fan and Lv [14] which is a linear model of the form

$$Y = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

in which all the columns of $\boldsymbol{X}$ are generated from a standard normal distribution and $\boldsymbol{\varepsilon}$ is a standard normal random error term. There are in total eight true active predictors which are generated as follows: set $a = 4\log(n)/n^{\frac{1}{2}}$ and the coefficients corresponding to the active predictors are derived by $(-1)^u(a + |z|)$, where u follows a Bernoulli distribution with $p = 0.4$ and z is drawn from a standard normal distribution.

Considering the results of quantile ensemble for Example 1 with normal error (Figure 3.1, Table 4), we observe the following: The performance of our candidate QaSIS at quantile levels 0.25, 0.5 and 0.75 are not satisfying. The medians of $\mathcal{R}$ for the candidates are 354, 210 and 322 respectively. Compared with the candidates, we observe some improvements in our quantile ensemble methods, where we see decreases in the magnitudes of $\mathcal{R}$ with respect to all our ensemble functions. Specifically, the medians of $\mathcal{R}$ drop to 89, 97, 69, 99.5 and 163 with respect to our five ensemble functions. Considering IQR($\mathcal{S}$), all of our ensemble methods show smaller IQR except for the mean voting ensemble. Regarding $\mathcal{S}$, the medians all increase to 0.875 which is higher than our candidates which have medians of 0.75. This fact shows that our quantile ensemble method is capable of discovering more active variables than the candidate QaSIS method.

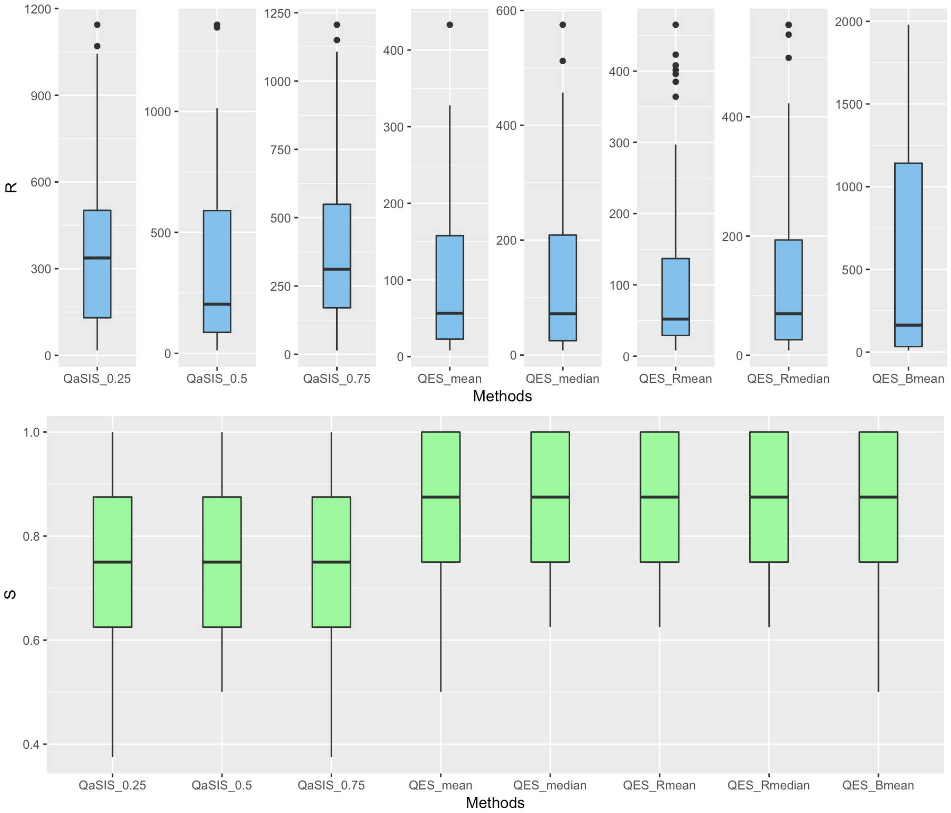


Figure 3.1: Example 1, quantile ensemble, $\boldsymbol{\varepsilon}$ =normal

EXAMPLE 2(a): $(n = 400, p = 1000)$. This example is originally from Fan *et al.* [12]. First, define the following functions

$$g_1(x) = x;$$

$$g_2(x) = (2x - 1)^2;$$

$$g_3(x) = \sin(2\pi x)^2/(2 - \sin(2\pi x));$$

$$g_4(x) = 0.1\sin(2\pi x) + 0.2\cos(2\pi x) + 0.3\sin(2\pi x)^3 + 0.5\sin(2\pi x)^3.$$

The random data are generated from:

$$Y = 5g_1(X_1) + 3g_2(X_2) + 4g_3(X_3) + 6g_4(X_4) + \sqrt{1.74}\boldsymbol{\varepsilon},$$

where $\boldsymbol{X} = (X_1, \ldots, X_{1000})$ follows a multivariate normal distribution $\boldsymbol{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

in which $\boldsymbol{\mu} = \mathbf{0}$ and $\boldsymbol{\Sigma}_{ij} = \rho^{|i-j|}$ with $\rho = 0.8$.

EXAMPLE 2(b): $(n = 400, p = 1000)$. Same as Example 2(a), except that $\rho = 0.6$.

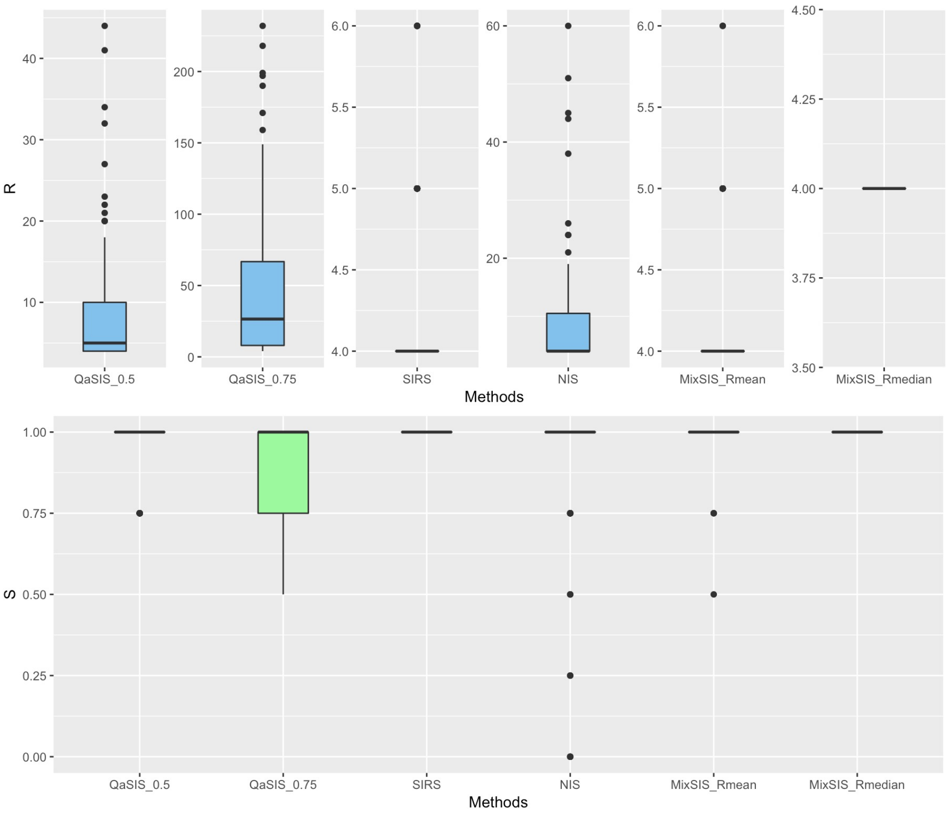


Figure 3.2: Example 2(b), mixture ensemble, $\boldsymbol{\varepsilon}$ =Cauchy

Considering the results of mixture ensemble method for Example 2(b) with Cauchy error (Figure 3.2, Table 32), we observe the following: Among three of our candidate methods, SIRS shows the best result. QaSIS and NIS are decent but fail to discover all the active variables in some occasional cases. Compared with the candidates, our MixSIS_Rmean and MixSIS_Rmedian show almost perfect performance regarding $\mathcal{R}$ with MixSIS_Rmean performing exactly the same with SIRS and MixSIS_Rmedian performing even better than SIRS. Regarding $\mathcal{S}$, both of our methods show medians of 1. The IQR($\mathcal{S}$) of MixSIS_Rmean is 1 and MixSIS_Rmean is 0. This example shows that when all the candidate work decently, our mixture ensemble method may lead to a better performance.

EXAMPLE 3: $(n = 200, p = 2000)$. This example is adopted from Zhu *et al.* [56], where the random data is generated from

$$Y = 2(X_1 + 0.8X_2 + 0.6X_3 + 0.4X_4 + 0.2X_5) + \exp(X_{20} + X_{21} + X_{22}) \cdot \boldsymbol{\varepsilon}.$$

$\boldsymbol{X} = (X_1, X_2, \ldots, X_{2000})$ follows a multivariate normal distribution with the same correlation structure described in Example 2. Being different from the first two examples, this model is heteroscedastic with the number of active variables 5 at the median but 8 elsewhere. Compared with other examples, this example is more challenging hence deserves more discussion.

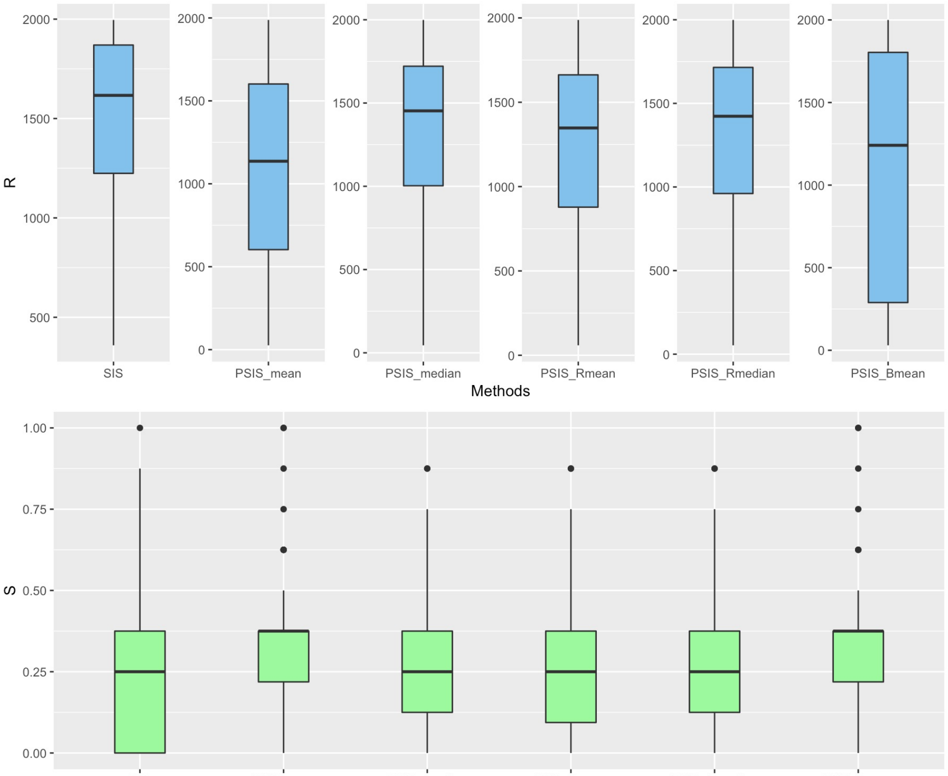


Figure 3.3: Example 3, parallel ensemble(PSIS), $\boldsymbol{\varepsilon}$ =normal

(1) For the parallel ensemble of SIS, we could observe the following (Figure

3.3, Table 34): The performance of SIS is poor in a sense that it fails to discover all the active variables. Our parallel ensemble screening method manages to promote the performance of SIS. Specifically, the median of $\mathcal{R}$ for SIS is above 1500. By applying our parallel ensemble method we observe decreases in all our ensemble functions with all of them showing medians of $\mathcal{R}$ below 1500. In addition, we observe IQR($\mathcal{R}$) showing decreases in PSIS_median and PSIS_Rmedian. Regarding $\mathcal{S}$, our methods show slightly better result for mean and mean voting ensemble functions.

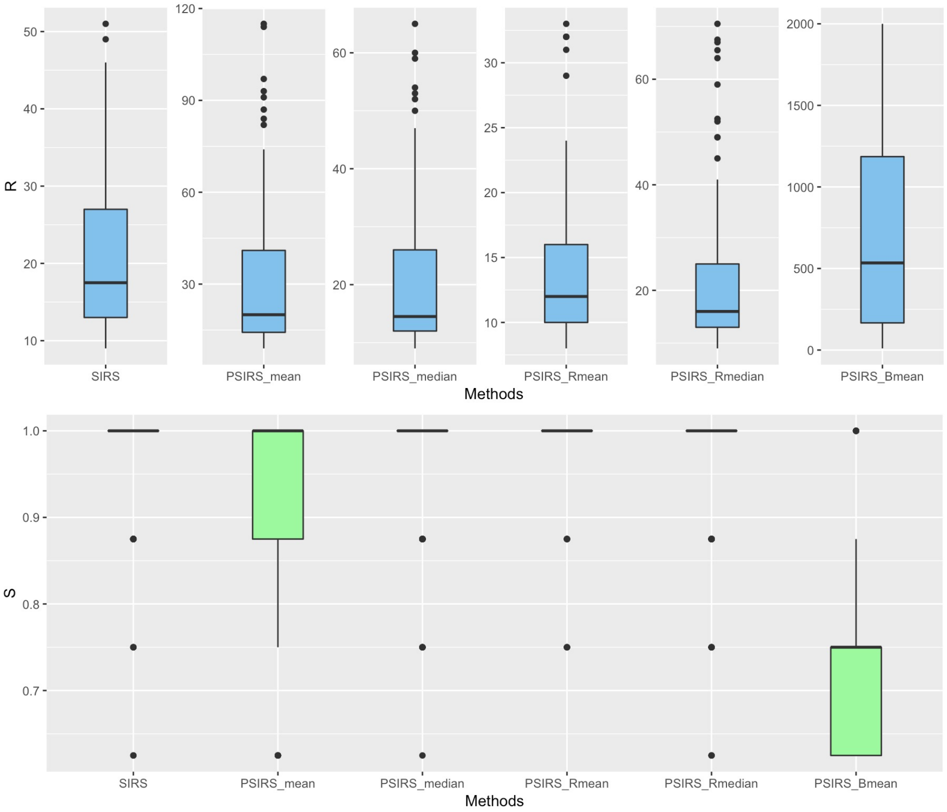


Figure 3.4: Example 3, parallel ensemble(PSIRS), ε =normal

(2) For the parallel ensemble of SIRS, we could observe the following (Figure 3.4, Table 34): In this heteroscedastic example, SIRS performs decently with a relative low $\mathcal{R}$ and a high $\mathcal{S}$. Both the candidate SIRS and our parallel ensembles show similar medians of $\mathcal{R}$. Considering the spread of $\mathcal{R}$, except for the mean voting ensemble, our methods show better results which

means the screening results are more stable. Regarding $\mathcal{S}$, our PSIRS_median, PSIRS_Rmean and PSIRS_Rmedian show similar results with PSIRS_Rmean slightly better.

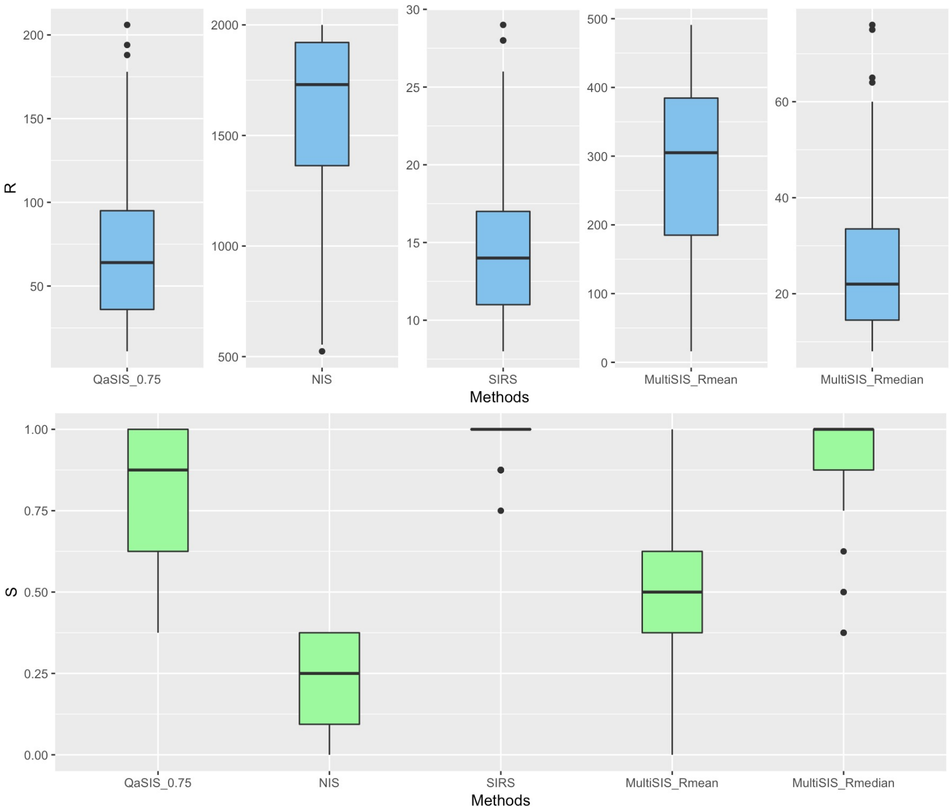


Figure 3.5: Example 3, multi ensemble, ε =mixture normal

Considering the results of multi ensemble method for Example 3 with mixture normal error (Figure 3.5, Table 42), we observe the following: Among the three candidate screening methods, only SIRS is working well. NIS fails to work with median of $\mathcal{R}$ more than 1500. QaSIS is better than NIS but still fails to discover all the active variables in most of the 100 runs. With two of the three candidate models working poorly, our MultiSIS_Rmedian manages to work well. Specifically, the median of $\mathcal{R}$ is around 25 and the IQR($\mathcal{R}$) is 25.75, which means our method performs slightly weaker than SIRS but much better than QaSIS and NIS. Regarding $\mathcal{S}$, the performance of our MultiSIS_Rmedian is also much better than QaSIS and NIS. This example shows, as long as one method in our ensemble works, the result may be satisfying.

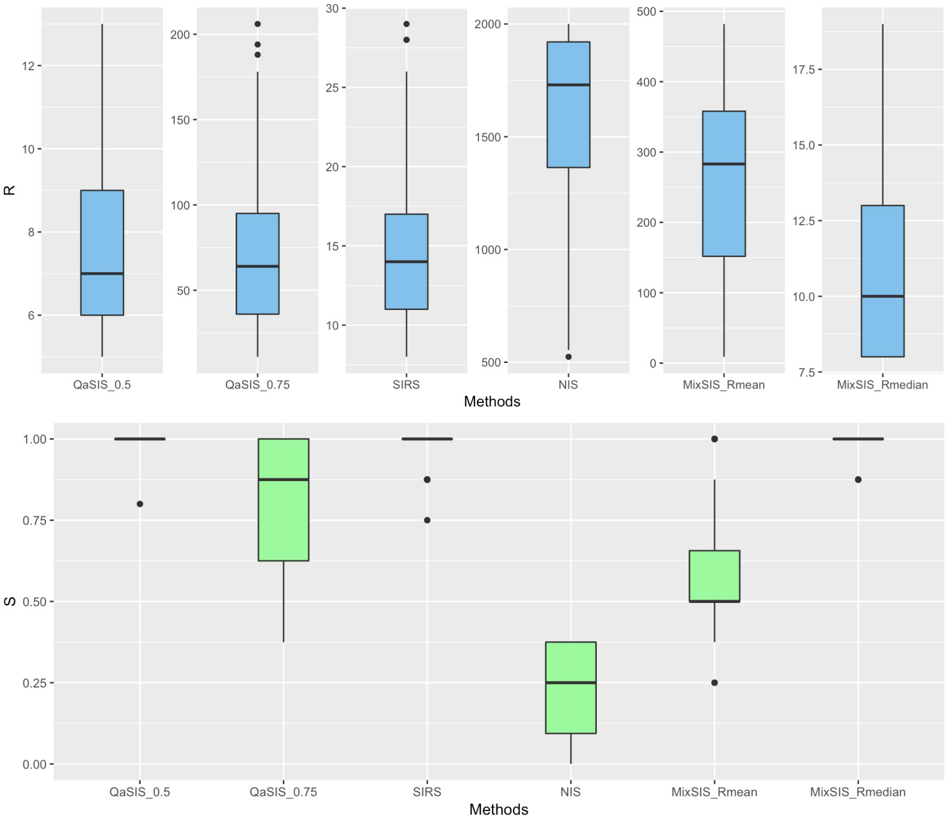


Figure 3.6: Example 3, mixture ensemble, ε =mixture normal

Considering the results of mixture ensemble method for Example 3 with mixture normal error (Figure 3.6, Table 45), we observe the following: The results are similar as in multi ensemble. However, benefited from parallel ensemble and quantile ensemble, our MixSIS_Rmedian outperforms SIRS. In detail, the median of $\mathcal{R}$ is 10 for MixSIS_Rmedian and IQR($\mathcal{R}$) is around 5. Regarding the performance with respect to $\mathcal{S}$, our MixSIS_Rmedian also shows the best performance compared with all candidate methods.

3.2 Real Data Analysis

In this section, we use the ADHD-200 data to illustrate the performance of our proposed ensemble method. Attention deficit hyperactivity disorder

(ADHD) is a brain disorder marked by an ongoing pattern of inattention and hyperactivity-impulsivity that interferes with functioning or development. The psychopathology of ADHD is marked by developmentally inappropriate and pervasive expressions of inattention, overactivity and impulsiveness. ADHD is the most commonly diagnosed mental disorder of children. Children with ADHD may be hyperactive and unable to control their impulses, or they may have trouble paying attention. ADHD is usually discovered during the early school years, when a child begins to have problems paying attention. Adults with ADHD may have trouble managing time, being organized, setting goals and holding down a job. They may also have problems with relationships, self-esteem. The understanding of the underlying pathophysiology of neuropsychiatric illnesses remains unclear [32] and few biomarkers are discovered to be related to ADHD [39]. Instead of biomarker detection approaches, recent development of medical imaging such as functional magnetic resonance imaging (fMRI) and diffusion tensor imaging (DTI) shows promising potential in predicting patients outcomes and understanding the underlying pathophysiology of diseases [8, 18, 42, 52].

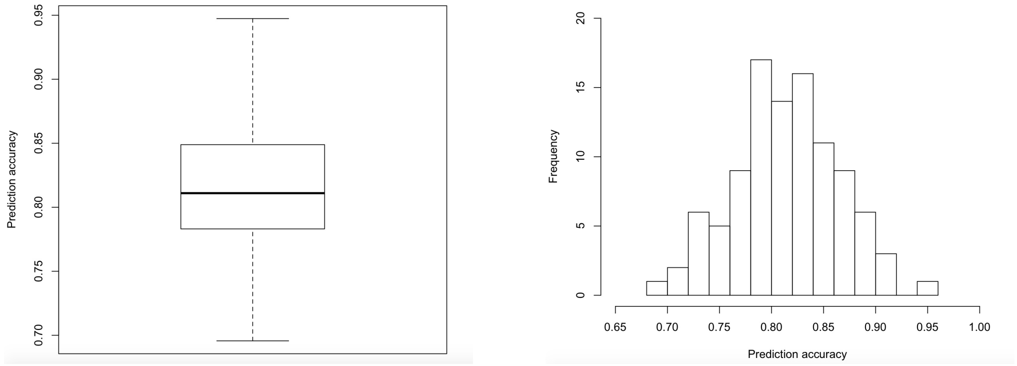
We use the ADHD-200 Consortium data which is a publicly available resting-state fMRI (rs-fMRI) data [38] in this study. fMRI measures brain activity by detecting changes associated with blood flow [24]. This technique relies on the fact that cerebral blood flow and neuronal activation are coupled. When an area of the brain is in use, blood flow to that region also increases [33]. rs-fMRI is useful for exploring the brain's functional organization and determining whether it is altered in neurological or psychiatric diseases. Resting-state functional connectivity research has revealed a number of networks which are consistently found in healthy subjects, different stages of consciousness and represent specific patterns of synchronous activity [1, 23]. This data set contains 120 subjects ($n = 120$) from the NYU site (New York University Child Study Center) of the ADHD-200 Consortium. The Anatomical Automatic Labeling (AAL) atlas [51] was used for the parcellation.

In this study, our goal is to find which partial connectivity pair of ROIs (region of interest) is contributing to the ADHD express levels hence each partial connectivity is considered to be a variable. For each subject, there are

172 time courses and the AAL has 116 ROIs. We use the suffixes .L and .R to differentiate the left and right hemispheres for some bilateral regions. The cerebra include 90 regions (45 in each hemisphere), and the cerebella include 26 regions (9 in each cerebellar hemisphere and 8 in the vermis). With respect to the 120 subjects, 42 are typically developing children showing ADHD negative and 78 are diagnosed as ADHD. For each subject, we obtain the mean time series for each of the 116 regions by averaging the fMRI time series over all voxels in the region, hence initially we have $p = (116 \times 116 - 116)/2 = 6670$ predictors. The partial correlation coefficient which measures the degree of the association between two interested regions are computed while controlling the effect of the remaining regions.

In the first step of the analysis, because of the large p small n scenario ($n = 120$, $p = 6670$), a variable screening procedure is necessary to remove some noise partial connectivities in order to apply lower dimensional variable selection approaches. For the same ADHD-200 data set, Karunamuni *et al.* [26] also applied a two-scale approach. In the screening step, they applied a Fisher's r-to-z transformation to improve the normality. Then they applied a two-tailed t test between the z values of the ADHD group and the control group to determine whether the functional connectivities are different. The selected significant functional connectivities between the ADHD subjects and the controls must satisfy two criteria: (1) Significantly different z values at the threshold of $p < 0.01$; (2) z values for the correlations that are significantly different from zero between groups at the threshold of $p < 0.01$. After this screening step, they selected $p = 34$ functional connectivities for the further variable selection approach.

In the screening step of our approach, we apply our mixture ensemble screening approach with the rank median ensemble function which is the best performed method in the simulation studies. We adopt the same settings as in the simulation studies to implement the screening. In the variable selection step, we employ both LASSO and SCAD by using R package **glmnet** and **ncvreg**. The tuning parameters in LASSO and SCAD are selected by five folds cross validation. The variable screening and selection results are showed in Table 3.1. In the next step, we choose 19 variables that are simultaneously selected by both LASSO and SCAD. For the classification, we adopt the support vector machine classifier (SVM) [22] with linear kernel by using the R package **caret**. The tunning parameter of SVM is selected by 10 folds cross validation. In order to get rid of occasional bias, firstly, the data are split by a bootstrap procedure 100 times to produce 100 train and test data pairs. Secondly, the SVM classifier are trained and tested 100 times on those data pairs. Consequently, the prediction accuracies are collected for each classification procedure.



(a) Boxplot for 100 prediction accuracies

(b) Histogram for 100 prediction accuracies

Figure 3.7: Classification results for ADHD-200

By looking at the classification results, we reach a prediction accuracy of around 81%, showing that the ROI connectivities we selected via the two-scale approach are significant in predicting ADHD. This high prediction rate therefore indicate we did not miss too many important predictors in screening step.

Table 3.1: Variable selection for mixture screening

ROI Connectivity	LASSO	SCAD
Frontal_Sup_Orb_L.Rolandic_Oper_R	-1.264142587	-1.886468375
Frontal_Inf_Tri_L.Olfactory_L	-1.196284422	-1.949297171
Olfactory_L.Insula_R	0	0
Insula_R.Fusiform_L	1.680086247	1.936251836
Amygdala_R.Postcentral_R	0.869929748	1.426413766
Fusiform_R.Postcentral_R	-0.986139105	-1.997351631
Occipital_Mid_R.Parietal_Inf_R	-1.652724358	-2.195266055
Hippocampus_L.Precuneus_L	-0.220848226	-0.40806945
ParaHippocampal_R.Precuneus_R	-1.082784621	-1.750876231
Precuneus_R.Caudate_L	-1.489974656	-2.158762855
Frontal_Inf_Tri_L.Putamen_R	0	0.025047932
Parietal_Inf_L.Temporal_Pole_Sup_R	0	0
Thalamus_L.Temporal_Pole_Mid_R	1.432851072	1.765494134
Temporal_Pole_Sup_L.Temporal_Inf_R	0	0
Temporal_Pole_Sup_R.Temporal_Inf_R	-0.606877197	-1.339274016
Temporal_Pole_Sup_L.Cerebelum_3_L	0	0
Parietal_Sup_R.Cerebelum_4_5_L	-0.372403171	-0.598690565
Frontal_Sup_Orb_L.Cerebelum_6_R	1.542967614	1.974995426
Caudate_L.Cerebelum_8_L	-0.08231351	-0.206272823
Cerebelum_7b_R.Cerebelum_8_L	1.35054048	1.563113485
Temporal_Pole_Mid_L.Cerebelum_8_R	0.857221497	1.3822996
Frontal_Sup_Medial_L.Cerebelum_10_R	1.655181645	2.163348383
Temporal_Mid_L.Vermis_8	1.444627193	1.765395888
Supp_Motor_Area_L.Vermis_10	-0.022173236	0
SupraMarginal_R.Vermis_10	0.612759358	1.470068916

Chapter 4

Conclusions and Future Research

By applying our ensemble screening approaches and employing the mean, median, rank mean, rank median and mean voting ensemble functions, we managed to implement ensemble techniques conveniently and efficiently in the variable screening scenarios. Considering the performance of our ensemble methods, we give a brief summary as follows: (1) The parallel ensemble screening will generally improve the performance of poorly performed screening algorithms and increase stability. For the methods that are already well performed, the effect may be minor. (2) The quantile ensemble screening will improve the performance of the candidate quantile based screening algorithms. In our simulations, the performance of ensembled QaSIS is boosted and sometimes outperforms other candidate methods. (3) The multi ensemble screening by combining different screening methods is benefited from all the candidate methods. (4) The mixture ensemble which is a hybrid of all our proposed ensemble screening methods is very promising in a sense that it is benefited from all of the three ensemble methods. Compared with other methods, it performs the best most of the times.

In addition, regarding the performance of the five ensemble functions, we have the following conclusions: (1) Mean voting ensemble seems the least preferred ensemble function since in general it has the weakest performance among all our ensemble functions. (2) In most of the settings, rank median ensemble function is the best performed among the five ensemble functions we adopted. Therefore, we recommend to use the rank median ensemble function in most of the screening problems.

The following issues related to this thesis deserve further discussion: (1) A natural extension of our mean, rank mean and mean voting ensemble is the weighted mean ensemble. Variable screening is different with the classification problem where we are offered training data that the true labels are given. However, in variable screening setting, we do not know which variable is the true active one. A tentative approach could be defining the weight as the reliability of each individual screening algorithms in which the reliability could be the variable reoccurrence. In our mixture ensemble screening for example, if the same set of variables are kept by a certain screening method multiple times based on different bootstrap samples, the variable reoccurrence would be considered high hence this method will be considered stable and assigned with higher weight. (2) We only adopted the hard threshold rule in this thesis for the sake of convenience and saving computation power. However, the soft threshold rule may be more applicable in weighted mean voting ensemble. The hard threshold rule is set as a number $\lfloor n/\log(n) \rfloor$ and this number will usually be much larger than the number of active variables. When considering a larger set of selected variables, it is more likely to have fluctuated sets instead of stable ones. The soft threshold rule will be likely to produce a smaller selected set which is more handy in distinguishing the stable methods from the unstable ones. (3) In our multi ensemble and mixture ensemble approaches, we only combined three screening methods. With a large amount of screening methods available, it is possible to ensemble more methods to produce a better multi ensemble or mixture ensemble screening method.

Bibliography

[1] Bharat B Biswal, Joel Van Kylen, and James S Hyde. Simultaneous assessment of flow and bold signals in resting-state functional connectivity maps. *NMR in Biomedicine*, 10(4-5):165–170, 1997.

[2] Johan Botling, Karolina Edlund, Miriam Lohr, Birte Hellwig, Lars Holmberg, Mats Lambe, Anders Berglund, Simon Ekman, Michael Bergqvist, and Fredrik Pontén. Biomarker discovery in non-small cell lung cancer: integrating gene expression profiling, meta-analysis and tissue microarray validation. *Clinical Cancer Research*, pages 194–204, 2012.

[3] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[4] Leo Breiman. Bias, variance, and arcing classifiers. Technical report, 1996.

[5] Leo Breiman. Arcing classifier. *The Annals of Statistics*, 26(3):801–849, 1998.

[6] Gavin Brown and Ludmila I. Kuncheva. "Good" and "bad" diversity in majority vote ensembles. In Neamat El Gayar, Josef Kittler, and Fabio Roli, editors, *Multiple Classifier Systems*, pages 124–133, Berlin, Heidelberg, 2010.

[7] Emmanuel Candes and Terence Tao. The Dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, 2007.

[8] F Xavier Castellanos, Jay N Giedd, Wendy L Marsh, Susan D Hamburger, A Catherine Vaituzis, Daniel P Dickstein, Stacey E Sarfatti, Yolanda C Vauss, John W Snell, and Nicholas Lange. Quantitative brain magnetic resonance imaging in attention-deficit hyperactivity disorder. *Archives of General Psychiatry*, 53(7):607–616, 1996.

[9] Wei Chu, Zoubin Ghahramani, Francesco Falciani, and David L Wild. Biomarker discovery in microarray gene expression data with gaussian processes. *Bioinformatics*, 21(16):3385–3393, 2005.

[10] Thomas G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15, Berlin, Heidelberg, 2000.

[11] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.

[12] Jianqing Fan, Yang Feng, and Rui Song. Nonparametric independence screening in sparse ultra-high dimensional additive models. *Journal of the American Statistical Association*, 106(494):544–557, 2011.

[13] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

[14] Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):849–911, 2008.

[15] Jianqing Fan, Richard Samworth, and Yichao Wu. Ultrahigh dimensional feature selection: beyond the linear model. *Journal of Machine Learning Research*, 10:2013–2038, December 2009.

[16] Yingying Fan, Yinfei Kong, Daoji Li, and Jinchi Lv. Interaction pursuit with feature screening and selection. *arXiv:1605.08933*, 2016.

[17] David Freedman. A remark on the difference between sampling with and without replacement. *Journal of the American Statistical Association*, 72(359):681–681, 1977.

[18] Michael D Greicius, Benjamin H Flores, Vinod Menon, Gary H Glover, Hugh B Solvason, Heather Kenna, Allan L Reiss, and Alan F Schatzberg. Resting-state functional connectivity in major depression: abnormally increased contributions from subgenual cingulate cortex and thalamus. *Biological Psychiatry*, 62(5):429–437, 2007.

[19] International SNP Map Working Group. A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature*, 409(6822):928, 2001.

[20] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, March 2003.

[21] Xuming He, Lan Wang, and Hyokyoung Grace Hong. Quantile-adaptive model-free variable screening for high-dimensional heterogeneous data. *The Annals of Statistics*, 41(1):342–369, 2013.

[22] Marti Hearst, Susan T. Dumais, E Osman, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems*, 13(4):18–28, July 1998.

[23] CJ Honey, O Sporns, Leila Cammoun, Xavier Gigandet, Jean-Philippe Thiran, Reto Meuli, and Patric Hagmann. Predicting human resting-state functional connectivity from structural connectivity. *Proceedings of the National Academy of Sciences*, 106(6):2035–2040, 2009.

[24] Scott A Huettel, Allen W Song, and Gregory McCarthy. *Functional magnetic resonance imaging*, volume 1. Sinauer Associates Sunderland, MA, 2004.

[25] Vernon M Ingram. A specific chemical difference between the globins of normal human and sickle-cell anaemia haemoglobin. *Nature*, 178(4537):792, 1956.

[26] Rohana J Karunamuni, Linglong Kong, and Wei Tu. Efficient robust doubly adaptive regularized regression with applications. *Statistical Methods in Medical Research*, Doi: 10.1177/0962280218757560, 2018.

[27] Xinbing Kong, Zhi Liu, Yuan Yao, and Wang Zhou. Sure screening by ranking the canonical correlations. *Test*, 26(1):46–70, 2017.

[28] Yinfei Kong, Daoji Li, Yingying Fan, and Jinchi Lv. Interaction pursuit in high-dimensional multi-response regression via distance correlation. *The Annals of Statistics*, 45(2):897–922, 2017.

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[30] Guodong Li, Yang Li, and Chih-Ling Tsai. Quantile correlations and quantile autoregressive modeling. *Journal of the American Statistical Association*, 110(509):246–261, 2015.

[31] Runze Li, Wei Zhong, and Liping Zhu. Feature screening via distance correlation learning. *Journal of the American Statistical Association*, 107(499):1129–1139, 2012.

[32] David EJ Linden. The challenges and promise of neuroimaging in psychiatry. *Neuron*, 73(1):8–22, 2012.

[33] Nikos K Logothetis, Jon Pauls, Mark Augath, Torsten Trinath, and Axel Oeltermann. Neurophysiological investigation of the basis of the fMRI signal. *Nature*, 412(6843):150, 2001.

[34] Xiaohui Luo, Leonard A Stefanski, and Dennis D Boos. Tuning variable selection procedures by adding noise. *Technometrics*, 48(2):165–175, 2006.

[35] Shujie Ma, Runze Li, and Chih-Ling Tsai. Variable screening via quantile partial correlation. *Journal of the American Statistical Association*, 112(518):650–663, 2017.

[36] Xuejun Ma and Jingxiao Zhang. Robust model-free feature screening via quantile correlation. *Journal of Multivariate Analysis*, 143:472–480, 2016.

[37] Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

[38] Michael P Milham, Damien Fair, Maarten Mennes, and Stewart Mostofsky. The ADHD-200 consortium: a model to advance the translational potential of neuroimaging in clinical neuroscience. *Frontiers in Systems Neuroscience*, 6:62, 2012.

[39] Eric J Nestler and Steven E Hyman. Animal models of neuropsychiatric disorders. *Nature Neuroscience*, 13(10):1161, 2010.

[40] Tien Thanh Nguyen, Xuan Cuong Pham, Alan Wee-Chung Liew, and Witold Pedrycz. Aggregation of classifiers: a justifiable information granularity approach. *IEEE Transactions on Cybernetics*, pages 1–10, 2018.

[41] Wenliang Pan, Xueqin Wang, Weinan Xiao, and Hongtu Zhu. A generic sure independence screening procedure. *Journal of the American Statistical Association*, Doi: 10.1080/01621459.2018.1462709, 2018.

[42] Serge ARB Rombouts, Jessica S Damoiseaux, Rutger Goekoop, Frederik Barkhof, Philip Scheltens, Stephen M Smith, and Christian F Beckmann. Model-free group analysis shows altered bold fMRI networks in dementia. *Human Brain Mapping*, 30(1):256–266, 2009.

[43] Robert E Schapire and Yoav Freund. *Boosting: Foundations and Algorithms.* MIT press, 2012.

[44] Mark Schena. *Microarray Analysis.* Wiley-Liss Hoboken, NJ, 2003.

[45] Dominik Scherer, Andreas Müller, and Sven Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In *Artificial Neural Networks–ICANN 2010*, pages 92–101. Springer, 2010.

[46] Xiaofeng Shao and Jingsi Zhang. Martingale difference correlation and its use in high-dimensional variable screening. *Journal of the American Statistical Association*, 109(507):1302–1318, 2014.

[47] Qian Shi. Variable screening based on combining quantile regression. Master's thesis, University of Alberta, 2014.

[48] John G Sled, Alex P Zijdenbos, and Alan C Evans. A nonparametric method for automatic correction of intensity nonuniformity in MRI data. *IEEE Transactions on Medical Imaging*, 17(1):87–97, 1998.

[49] Daniel Stucht, K Appu Danishad, Peter Schulze, Frank Godenschweger, Maxim Zaitsev, and Oliver Speck. Highest resolution in vivo human brain MRI using prospective motion correction. *PLOS ONE*, 10(7), 2015.

[50] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, (1):267–288, 1996.

[51] Nathalie Tzourio-Mazoyer, Brigitte Landeau, Dimitri Papathanassiou, Fabrice Crivello, Olivier Etard, Nicolas Delcroix, Bernard Mazoyer, and Marc Joliot. Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. *Neuroimage*, 15(1):273–289, 2002.

[52] Liang Wang, Yufeng Zang, Yong He, Meng Liang, Xinqing Zhang, Lixia Tian, Tao Wu, Tianzi Jiang, and Kuncheng Li. Changes in hippocampal connectivity in the early stages of Alzheimer's disease: evidence from resting state fMRI. *Neuroimage*, 31(2):496–504, 2006.

[53] Faisal Zaman and Hideo Hirose. A robust bagging method using median as a combination rule. *2008 IEEE 8th International Conference on Computer and Information Technology Workshops*, pages 55–60, 2008.

[54] Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942, 2010.

[55] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms.* Chapman and Hall/CRC, 2012.

[56] Liping Zhu, Lexin Li, Runze Li, and Lixing Zhu. Model-free feature screening for ultrahigh-dimensional data. *Journal of the American Statistical Association*, 106(496):1464–1475, 2011.

[57] Alex P Zijdenbos, Reza Forghani, and Alan C Evans. Automatic pipeline analysis of 3-D MRI data for clinical trials: application to multiple sclerosis. *IEEE transactions on medical imaging*, 21(10):1280–1291, 2002.

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

[59] Hui Zou and Ming Yuan. Composite quantile regression and the oracle model selection theory. *The Annals of Statistics*, 36(3):1108–1126, 2008.

Appendices

Table 1: Example 1, Parallel Ensemble, Error= Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.5)	210	514.5	0.75	0.25	8
PQaSIS_mean(0.5)	170.5	497.25	0.75	0.125	8
PQaSIS_median(0.5)	214	592.75	0.75	0.125	8
PQaSIS_Rmean(0.5)	179	530.125	0.875	0.125	8
PQaSIS_Rmedian(0.5)	204.75	535.875	0.75	0.125	8
PQaSIS_Rmean(0.5)	381.5	858	0.75	0.125	8
QaSIS(0.75)	322	427.75	0.75	0.25	8
PQaSIS_mean(0.75)	218	421.25	0.75	0.125	8
PQaSIS_median(0.75)	275.5	568.5	0.75	0.25	8
PQaSIS_Rmean(0.75)	211	445.5	0.75	0.125	8
PQaSIS_Rmedian(0.75)	260	499.375	0.75	0.25	8
PQaSIS_Bmean(0.75)	495.5	1065.5	0.75	0.25	8
NIS	43	115	0.875	0.125	8
PNIS_mean	42.5	154.75	0.875	0.125	8
PNIS_median	46.5	177	0.875	0.125	8
PNIS_Rmean	47.5	159	0.875	0.125	8
PNIS_Rmedian	50.25	162.375	0.875	0.125	8
PNIS_Bmean	56	203.25	0.875	0.15625	8
SIRS	22	67.5	1	0.125	8
PSIRS_mean	51	174	0.875	0.125	8
PSIRS_median	45	220	0.875	0.125	8
PSIRS_Rmean	42	178	0.875	0.125	8
PSIRS_Rmedian	54	236.125	0.875	0.125	8
PSIRS_Bmean	184.5	523.75	0.875	0.125	8
SIS	13.5	45.25	1	0.125	8
PSIS_mean	16	36	1	0.125	8
PSIS_median	15.5	29	1	0.03125	8
PSIS_Rmean	18.5	36	1	0.125	8
PSIS_Rmedian	15	31.125	1	0.125	8
PSIS_Bmean	17	42.25	1	0.125	8

Table 2: Example 1, Parallel Ensemble, Error= Cauchy

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.5)	327.5	423.75	0.75	0.15625	8
PQaSIS_mean(0.5)	278.5	355.5	0.75	0.125	8

PQaSIS_median(0.5)	349	475.75	0.75	0.125	8
PQaSIS_Rmean(0.5)	270	388.5	0.75	0.25	8
PQaSIS_Rmedian(0.5)	323	412.25	0.75	0.125	8
PQaSIS_Bmean(0.5)	840.5	1146	0.625	0.125	8
QaSIS(0.75)	581.5	675.25	0.625	0.125	8
PQaSIS_mean(0.75)	522.5	509.5	0.5	0.25	8
PQaSIS_median(0.75)	587	730.75	0.625	0.15625	8
PQaSIS_Rmean(0.75)	451.5	613.5	0.625	0.15625	8
PQaSIS_Rmedian(0.75)	492	663.5	0.625	0.15625	8
PQaSIS_Bmean(0.75)	1230	1181.75	0.5	0.25	8
NIS	1291.5	1107.25	0.125	0.5	8
PNIS_mean	1305.5	1096.25	0	0.375	8
PNIS_median	1124	1170.75	0.125	0.5	8
PNIS_Rmean	1130.75	1004.5	0.25	0.375	8
PNIS_Rmedian	1172	1180.25	0.125	0.5	8
PNIS_Bmean	1413.5	1361	0.25	0.5	8
SIRS	66	132.5	0.875	0.125	8
PSIRS_mean	117.5	221.75	0.875	0.125	8
PSIRS_median	128	263	0.875	0.125	8
PSIRS_Rmean	121.5	215	0.875	0.125	8
PSIRS_Rmedian	123.5	274.25	0.875	0.125	8
PSIRS_Bmean	416.5	787.5	0.75	0.25	8
SIS	1198.5	1328.75	0.375	0.625	8
PSIS_mean	682.5	848.5	0.5	0.5	8
PSIS_median	962.5	1097.5	0.375	0.5	8
PSIS_Rmean	842	880.75	0.375	0.5	8
PSIS_Rmedian	967.5	1133.375	0.375	0.5	8
PSIS-_Bmean	940.5	1408.5	0.5	0.625	8

Table 3: Example 1, Parallel Ensemble, Error= Mixture Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.5)	203.5	297.5	0.75	0.25	8
PQaSIS_mean(0.5)	145.5	333.25	0.875	0.125	8
PQaSIS_median(0.5)	177.5	353.25	0.75	0.125	8
PQaSIS_Rmean(0.5)	168	375.25	0.875	0.125	8
PQaSIS_Rmedian(0.5)	178.75	370.5	0.8125	0.125	8
PQaSIS_Bmean(0.5)	334	492.5	0.75	0.25	8

QaSIS(0.75)	318.5	341.25	0.625	0.25	8
PQaSIS_mean(0.75)	210	342.25	0.75	0.15625	8
PQaSIS_median(0.75)	238.5	435.75	0.6875	0.125	8
PQaSIS_Rmean(0.75)	212.5	377	0.75	0.125	8
PQaSIS_Rmedian(0.75)	219.5	402.125	0.75	0.125	8
PQaSIS_Bmean(0.75)	340	490	0.625	0.125	8
NIS	65.5	153.75	0.875	0.125	8
PNIS_mean	87	166	0.875	0.15625	8
PNIS_median	95	169.5	0.875	0.125	8
PNIS_Rmean	82.5	131.75	0.875	0.125	8
PNIS_Rmedian	95.75	153.25	0.875	0.25	8
PNIS_Bmean	101.5	315.5	0.875	0.15625	8
SIRS	36.5	74	0.9375	0.125	8
PSIRS_mean	49.5	124	0.875	0.125	8
PSIRS_median	51.5	123.5	0.875	0.125	8
PSIRS_Rmean	57	117.25	0.875	0.125	8
PSIRS_Rmedian	58	114.75	0.875	0.125	8
PSIRS_Bmean	126	404	0.875	0.125	8
SIS	24.5	44	1	0.125	8
PSIS_mean	27	44.25	1	0.125	8
PSIS_median	27	53.5	1	0.125	8
PSIS_Rmean	29	55.5	1	0.125	8
PSIS_Rmedian	28.25	53.5	1	0.125	8
PSIS_Bmean	29	53.75	1	0.125	8

Table 4: Example 1, Quantile Ensemble, Error= Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.25)	354	429	0.75	0.25	8
QaSIS(0.5)	210	514.5	0.75	0.25	8
QaSIS(0.75)	322	427.75	0.75	0.25	8
NIS	43	115	0.875	0.125	8
SIRS	22	67.5	1	0.125	8
QES_mean	89	179	0.875	0.25	8
QES_median	97	224.5	0.875	0.25	8
QES_Rmean	69	182.25	0.875	0.25	8
QES_Rmedian	99.5	219.125	0.875	0.25	8
QES_Bmean	163	1108.25	0.875	0.25	8

Table 5: Example 1, Quantile Ensemble, Error= Cauchy

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.25)	546.5	628.75	0.5	0.125	8
QaSIS(0.5)	327.5	423.75	0.75	0.15625	8
QaSIS(0.75)	581.5	675.25	0.625	0.125	8
NIS	1291.5	1107.25	0.125	0.5	8
SIRS	66	132.5	0.875	0.125	8
QES_mean	714.5	507.25	0	0.125	8
QES_median	163.5	340	0.75	0.25	8
QES_Rmean	164.5	263.875	0.75	0.25	8
QES_Rmedian	201.75	263	0.75	0.25	8
QES_Bmean	994	1403	0.75	0.25	8

Table 6: Example 1, Quantile Ensemble, Error= Mix Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.25)	340.5	461	0.625	0.125	8
QaSIS(0.5)	306.5	565.25	0.75	0.25	8
QaSIS(0.75)	594.5	686	0.625	0.25	8
NIS	111	251.5	0.875	0.125	8
SIRS	42	141	0.875	0.125	8
QES_mean	173.5	367.5	0.875	0.125	8
QES_median	190.5	308.75	0.875	0.125	8
QES_Rmean	146	296.75	0.875	0.125	8
QES_Rmedian	146.5	364.75	0.8125	0.125	8
QES_Bmean	358.5	1203	0.75	0.125	8

Table 7: Example 1, Multi Ensemble, Error= Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.75)	322	427.75	0.75	0.25	8
NIS	43	115	0.875	0.125	8
SIRS	22	67.5	1	0.125	8
MultiSIS_Rmean	45.5	101	0.875	0.125	8
MultiSIS_Rmedian	29.5	79	1	0.125	8

Table 8: Example 1, Multi Ensemble, Error= Cauchy

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.75)	581.5	675.25	0.625	0.125	8
NIS	1291.5	1107.25	0.125	0.5	8
SIRS	66	132.5	0.875	0.125	8
MultiSIS_Rmean	339	331	0.625	0.375	8
MultiSIS_Rmedian	242.5	480	0.75	0.25	8

Table 9: Example 1, Multi Ensemble, Error= Mixture Normal

name	R	IQR(R)	S	IQR(S)	p^*
QaSIS(0.75)	594.5	686	0.625	0.25	8
NIS	111	251.5	0.875	0.125	8
SIRS	42	141	0.875	0.125	8
MultiSIS_mean	122.5	237.25	0.875	0.125	8
MultiSIS_median	74	211.5	0.875	0.125	8

Table 10: Example 1, Mixture Ensemble, Error= Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.5)	210	514.5	0.75	0.25	8
QaSIS(0.75)	322	427.75	0.75	0.25	8
SIRS	22	67.5	1	0.125	8
NIS	43	115	0.875	0.125	8
MixSIS_Rmean	21	65.25	1	0.125	8
MixSIS_Rmedian	39.5	117.25	0.875	0.125	8

Table 11: Example 1, Mixture Ensemble, Error= Cauchy

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.5)	327.5	423.75	0.75	0.15625	8
QaSIS(0.75)	581.5	675.25	0.625	0.125	8
SIRS	66	132.5	0.875	0.125	8
NIS	1291.5	1107.25	0.125	0.5	8
MixSIS_Rmean	231	245	0.625	0.375	8
MixSIS_Rmedian	133	266	0.75	0.125	8

Table 12: Example 1, Mixture Ensemble, Error= Mixture Normal

name	R	IQR(R)	S	IQR(S)	p^*
QaSIS(0.5)	306.5	565.25	0.75	0.25	8
QaSIS(0.75)	594.5	686	0.625	0.25	8
SIRS	42	141	0.875	0.125	8
NIS	111	251.5	0.875	0.125	8
MultiSIS_Mean	55	130.75	0.875	0.125	8
MultiSIS_Median	89	190.5	0.875	0.15625	8

Table 13: Example 2(a), Parallel Ensemble, Error= Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.5)	4	0	1	0	4
PQaSIS_mean(0.5)	4	0	1	0	4
PQaSIS_median(0.5)	4	0	1	0	4
PQaSIS_Rmean(0.5)	4	0	1	0	4
PQaSIS_Rmedian(0.5)	4	0	1	0	4
PQaSIS_Bmean(0.5)	4	0	1	0	4
QaSIS(0.75)	4	1	1	0	4
PQaSIS_mean(0.75)	4	2.25	1	0	4
PQaSIS_median(0.75)	4	1	1	0	4
PQaSIS_Rmean(0.75)	4	0	1	0	4
PQaSIS_Rmedian(0.75)	4	0	1	0	4
PQaSIS_Bmean(0.75)	4	0	1	0	4
NIS	4	0	1	0	4
PNIS-mean	4	0	1	0	4
PNIS_median	4	0	1	0	4
PNIS_Rmean	4	0	1	0	4
PNIS_Rmedian	4	0	1	0	4
PNIS_Bmean	4	1	1	0	4
SIRS	4	0	1	0	4
PSIRS_mean	4	0	1	0	4
PSIRS_median	4	0	1	0	4
PSIRS_Rmean	4	0	1	0	4
PSIRS_Rmedian	4	0	1	0	4
PSIRS_Bmean	4	0	1	0	4
SIS	4	0	1	0	4

PSIS_mean	4	0	1	0	4
PSIS_median	4	0	1	0	4
PSIS_Rmean	4	0	1	0	4
PSIS_Rmedian	4	0	1	0	4
PSIS_Rmean	4	1	1	0	4

Table 14: Example 2(a), Parallel Ensemble, Error= Cauchy

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.5)	4	0	1	0	4
PQaSIS_mean(0.5)	4	0	1	0	4
PQaSIS_median(0.5)	4	0	1	0	4
PQaSIS_Rmean(0.5)	4	0	1	0	4
PQaSIS_Rmedian(0.5)	4	0	1	0	4
PQaSIS_Bmean(0.5)	4	0	1	0	4
QaSIS(0.75)	4	2	1	0	4
PQaSIS_mean(0.75)	5	4	1	0	4
PQaSIS_median(0.75)	4	1	1	0	4
PQaSIS_Rmean(0.75)	4	0	1	0	4
PQaSIS_Rmedian(0.75)	4	1	1	0	4
PQaSIS_Bmean(0.75)	4	1	1	0	4
NIS	4	1	1	0	4
PNIS_mean	4	2	1	0	4
PNIS_median	4	1	1	0	4
PNIS_Rmean	4	0	1	0	4
PNIS_Rmedian	4	1	1	0	4
PNIS_Bmean	4	1	1	0	4
SIRS	4	0	1	0	4
PSIRS_mean	4	0	1	0	4
PSIRS_median	4	0	1	0	4
PSIRS_Rmean	4	0	1	0	4
PSIRS_Rmedian	4	0	1	0	4
PSIRS_Bmean	4	0	1	0	4
SIS	4	0	1	0	4
PSIS_mean	4	0	1	0	4
PSIS_median	4	0	1	0	4
PSIS_Rmean	4	0	1	0	4
PSIS_Rmedian	4	0	1	0	4
PSIS_Bmean	4	1	1	0	4

Table 15: Example 2(a), Quantile Ensemble, Error= Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.25)	4	0	1	0	4
QaSIS(0.5)	4	0	1	0	4
QaSIS(0.75)	4	0	1	0	4
NIS	4	0	1	0	4
SIRS	4	0	1	0	4
QES_mean	4	0	1	0	4
QES_median	4	0	1	0	4
QES_Rmean	4	0	1	0	4
QES_Rmedian	4	0	1	0	4
QES_Bmean	5	1	1	0	4

Table 16: Example 2(a), Quantile Ensemble, Error= Cauchy

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.25)	4	0	1	0	4
QaSIS(0.5)	4	0	1	0	4
QaSIS(0.75)	4	0	1	0	4
NIS	4	0.25	1	0	4
SIRS	4	0	1	0	4
QES_mean	4	1	1	0	4
QES_median	4	0	1	0	4
QES_Rmean	4	0	1	0	4
QES_Rmedian	4	0	1	0	4
QES_Bmean	5	2	1	0	4

Table 17: Example 2(a), Quantile Ensemble, Error= Mix Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.25)	4	0	1	0	4
QaSIS(0.5)	4	0	1	0	4
QaSIS(0.75)	4	0	1	0	4
NIS	4	0	1	0	4
SIRS	4	0	1	0	4
QES_mean	4	0	1	0	4
QES_median	4	0	1	0	4
QES_Rmean	4	0	1	0	4
QES_Rmedian	4	0	1	0	4
QES_Bmean	5	1	1	0	4

Table 18: Example 2(a), Multi Ensemble, Error= Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.75)	4	0	1	0	4
NIS	4	0	1	0	4
SIRS	4	0	1	0	4
MultiSIS_Rmean	4	0	1	0	4
MultiSIS_Rmedian	4	0	1	0	4

Table 19: Example 2(a), Multi Ensemble, Error= Cauchy

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.75)	4	0	1	0	4
NIS	4	0	1	0	4
SIRS	4	0	1	0	4
MultiSIS_Rmean	4	0	1	0	4
MultiSIS_Rmedian	4	0	1	0	4

Table 20: Example 2(a), Mixture Ensemble, Error= Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.5)	4	0	1	0	4
QaSIS(0.75)	4	0	1	0	4
SIRS	4	0	1	0	4
NIS	4	0	1	0	4
MixSIS_Rmean	4	0	1	0	4
MixSIS_Rmedian	4	0	1	0	4

Table 21: Example 2(a), Mixture Ensemble, Error= Cauchy

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.5)	4	0	1	0	4
QaSIS(0.75)	4	0	1	0	4
SIRS	4	0	1	0	4
NIS	4	0	1	0	4
MixSIS_Rmean	4	0	1	0	4
MixSIS_Rmedian	4	0	1	0	4

Table 22: Example 2(b), Parallel Ensemble, Error= Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.5)	6	12.5	1	0	4
PQaSIS_mean(0.5)	5	4	1	0	4
PQaSIS_median(0.5)	5	5.25	1	0	4
PQaSIS_Rmean(0.5)	4.5	4.25	1	0	4
PQaSIS_Rmedian(0.5)	5	7	1	0	4
PQaSIS_Bmean(0.5)	5	9	1	0	4
QaSIS(0.75)	21.5	79.5	1	0.25	4
PQaSIS_mean(0.75)	16	62	1	0.25	4
PQaSIS_median(0.75)	18	54.25	1	0	4
PQaSIS_Rmean(0.75)	13.25	42.75	1	0	4
PQaSIS_Rmedian(0.75)	16	58.25	1	0.0625	4

PQaSIS_Bmean(0.75)	18.5	64	1	0.25	4
NIS	4	1	1	0	4
PNIS_mean	5	3	1	0	4
PNIS_median	4	1.25	1	0	4
PNIS_Rmean	4	1	1	0	4
PNIS_Rmedian	4	2.125	1	0	4
PNIS_Bmean	4	1	1	0	4
SIRS	4	0	1	0	4
PSIRS_mean	4	1	1	0	4
PSIRS_median	4	0.25	1	0	4
PSIRS_Rmean	4	0	1	0	4
PSIRS_Rmedian	4	1	1	0	4
PSIRS_Bmean	4	1	1	0	4
SIS	4	0	1	0	4
PSIS_mean	4	0	1	0	4
PSIS_median	4	0	1	0	4
PSIS_Rmean	4	0	1	0	4
PSIS_Rmedian	4	0	1	0	4
PSIS_Bmean	4	0	1	0	4

Table 23: Example 2(b), Parallel Ensemble, Error= Cauchy

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.5)	7	16.25	1	0	4
PQaSIS_mean(0.5)	6	6.25	1	0	4
PQaSIS_median(0.5)	6	9.75	1	0	4
PQaSIS_Rmean(0.5)	5	10.25	1	0	4
PQaSIS_Rmedian(0.5)	6	10	1	0	4
PQaSIS_Bmean(0.5)	5	10.25	1	0	4
QaSIS(0.75)	33.5	92.25	1	0.25	4
PQaSIS_mean(0.75)	29.5	102.5	1	0.25	4
PQaSIS_median(0.75)	25	105.25	1	0.25	4
PQaSIS_Rmean(0.75)	24	91.25	1	0.25	4
PQaSIS_Rmedian(0.75)	26.5	93.5	1	0.25	4
PQaSIS_Bmean(0.75)	26.5	135	1	0.25	4
NIS	7.5	35.5	1	0	4
PNIS_mean	10.5	63.25	1	0.25	4
PNIS_median	7.5	37	1	0	4

PNIS_Rmean	5	20.25	1	0	4
PNIS_Rmedian	7	34.5	1	0	4
PNIS_Bmean	7	30.75	1	0	4
SIRS	4	1	1	0	4
PSIRS_mean	4	1	1	0	4
PSIRS_median	4	1	1	0	4
PSIRS_Rmean	4	1	1	0	4
PSIRS_Rmedian	4	1	1	0	4
PSIRS_Bmean	4	1	1	0	4
SIS	4	4	1	0	4
PSIS_mean	4	3	1	0	4
PSIS_median	4	4	1	0	4
PSIS_Rmean	4	4	1	0	4
PSIS_Rmedian	4	4	1	0	4
PSIS_Bmean	4	4.25	1	0	4

Table 24: Example 2(b), Parallel Ensemble, Error= Mixture Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.5)	6	16.25	1	0	4
PQaSIS_mean(0.5)	5.5	8.75	1	0	4
PQaSIS_median(0.5)	6	12	1	0	4
PQaSIS_Rmean(0.5)	5	10.25	1	0	4
PQaSIS_Rmedian(0.5)	6	13	1	0	4
PQaSIS_Bmean(0.5)	5	10	1	0	4
QaSIS(0.75)	35.5	99.75	1	0.25	4
PQaSIS_mean(0.75)	25.5	70.75	1	0.25	4
PQaSIS_median(0.75)	28	80.5	1	0.25	4
PQaSIS_Rmean(0.75)	16	64	1	0.25	4
PQaSIS_Rmedian(0.75)	27	72.625	1	0.25	4
PQaSIS_Bmean(0.75)	27	71.25	1	0.25	4
NIS	4	1	1	0	4
PNIS_mean	4	3	1	0	4
PNIS_median	4	1	1	0	4
PNIS_Rmean	4	0	1	0	4
PNIS_Rmedian	4	1.125	1	0	4
PNIS_Bmean	4	1	1	0	4
SIRS	4	0	1	0	4

PSIRS_mean	4	1	1	0	4
PSIRS_median	4	1	1	0	4
PSIRS_Rmean	4	0	1	0	4
PSIRS_Rmedian	4	1	1	0	4
PSIRS_Bmean	4	1	1	0	4
SIS	4	0	1	0	4
PSIS_mean	4	0	1	0	4
PSIS_median	4	0	1	0	4
PSIS_Rmean	4	0	1	0	4
PSIS_Rmedian	4	0	1	0	4
PSIS_Bmean	4	0	1	0	4

Table 25: Example 2(b), Quantile Ensemble, Error= Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.25)	5.5	9.25	1	0	4
QaSIS(0.5)	6	12.5	1	0	4
QaSIS(0.75)	21.5	79.5	1	0.25	4
NIS	4	1	1	0	4
SIRS	4	0	1	0	4
QES_mean	12.5	29.25	1	0	4
QES_median	4	3	1	0	4
QES_Rmean	4	1	1	0	4
QES_Rmedian	4	2	1	0	4
QES_Bmean	4	1	1	0	4

Table 26: Example 2(b), Quantile Ensemble, Error= Cauchy

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.25)	5	12	1	0	4
QaSIS(0.5)	7	16.25	1	0	4
QaSIS(0.75)	33.5	92.25	1	0.25	4
NIS	7.5	35.5	1	0	4
SIRS	4	1	1	0	4
QES_mean	26.5	40.25	1	0	4
QES_median	4	4.25	1	0	4
QES_Rmean	4	2	1	0	4
QES_Rmedian	4	1.125	1	0	4
QES_Bmean	4	2	1	0	4

Table 27: Example 2(b), Quantile Ensemble, Error= Mix Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.25)	5	5	1	0	4
QaSIS(0.5)	6	16.25	1	0	4
QaSIS(0.75)	35.5	99.75	1	0.25	4
NIS	4	1	1	0	4
SIRS	4	0	1	0	4
QES_mean	16.5	29.75	1	0	4
QES_median	4	5	1	0	4
QES_Rmean	4	2	1	0	4
QES_Rmedian	4	1	1	0	4
QES_Bmean	4	1	1	0	4

Table 28: Example 2(b), Multi Ensemble, Error= Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.75)	21.5	79.5	1	0.25	4
NIS	4	1	1	0	4
SIRS	4	0	1	0	4
MultiSIS_Rmean	4	2	1	0	4
MultiSIS_Rmedian	4	0	1	0	4

Table 29: Example 2(b), Multi Ensemble, Error= Cauchy

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.75)	33.5	92.25	1	0.25	4
NIS	7.5	35.5	1	0	4
SIRS	4	1	1	0	4
MultiSIS_Rmean	5	8.5	1	0	4
MultiSIS_Rmedian	4	2	1	0	4

Table 30: Example 2(b), Multi Ensemble, Error= Mixture Normal

name	R	IQR(R)	S	IQR(S)	p^*
QaSIS(0.75)	35.5	99.75	1	0.25	4
NIS	4	1	1	0	4
SIRS	4	0	1	0	4
MultiSIS_mean	4	3.25	1	0	4
MultiSIS_median	4	0	1	0	4

Table 31: Example 2(b), Mixture Ensemble, Error= Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.5)	6	12.5	1	0	4
QaSIS(0.75)	21.5	79.5	1	0.25	4
SIRS	4	0	1	0	4
NIS	4	1	1	0	4
MixSIS_Rmean	4	0	1	0	4
MixSIS_Rmedian	4	0	1	0	4

Table 32: Example 2(b), Mixture Ensemble, Error= Cauchy

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.5)	7	16.25	1	0	4
QaSIS(0.75)	33.5	92.25	1	0.25	4
SIRS	4	1	1	0	4
NIS	7.5	35.5	1	0	4
MixSIS_Rmean	4	1	1	0	4
MixSIS_Rmedian	4	0	1	0	4

Table 33: Example 2(b), Mixture Ensemble, Error= Mixture Normal

name	R	IQR(R)	S	IQR(S)	p^*
QaSIS(0.5)	6	16.25	1	0	4
QaSIS(0.75)	35.5	99.75	1	0.25	4
SIRS	4	0	1	0	4
NIS	4	1	1	0	4
MultiSIS_Mean	4	0	1	0	4
MultiSIS_Median	4	0	1	0	4

Table 34: Example 3, Parallel Ensemble, Error= Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.5)	6	2	1	0	5
PQaSIS_mean(0.5)	8	2	1	0	5
PQaSIS_median(0.5)	6	2	1	0	5
PQaSIS_Rmean(0.5)	5	0	1	0	5
PQaSIS_Rmedian(0.5)	6	2	1	0	5
PQaSIS_Bmean(0.5)	5	1	1	0	5
QaSIS(0.75)	19	16	1	0	8
PQaSIS_mean(0.75)	81	102.25	0.75	0.5	8
PQaSIS_median(0.75)	19	14.25	1	0	8
PQaSIS_Rmean(0.75)	12.5	13.25	1	0	8
PQaSIS_Rmedian(0.75)	15	13	1	0	8
PQaSIS_Bmean(0.75)	26	45	1	0.125	8
NIS	1739.5	455.75	0.25	0.25	8
PNIS_mean	1568	668	0.25	0.25	8
PNIS_median	1481.5	642	0.25	0.25	8
PNIS_Rmean	1392	660.625	0.125	0.375	8
PNIS_Rmedian	1484	694.125	0.25	0.25	8
PNIS_Bmean	1628	669.25	0.25	0.25	8
SIRS	20.5	16.5	1	0	8
PSIRS_mean	21.5	41	1	0.125	8
PSIRS_median	15.5	22	1	0	8
PSIRS_Rmean	12	9.5	1	0	8
PSIRS_Rmedian	17	24	1	0	8
PSIRS_Bmean	534	1019.25	0.75	0.125	8
SIS	1587.5	649.5	0.25	0.375	8
PSIS_mean	1136	998.75	0.375	0.15625	8
PSIS_median	1453	717.75	0.25	0.25	8
PSIS_Rmean	1347.5	784.25	0.25	0.28125	8
PSIS_Rmedian	1423	754.25	0.25	0.25	8
PSIS_Bmean	1241	1514.5	0.375	0.15625	8

Table 35: Example 3, Parallel Ensemble, Error= Cauchy

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.5)	7	2.25	1	0	5
QaSIS_mean(0.5)	12	7.25	1	0	5

QaSIS_median(0.5)	8	3	1	0	5
QaSIS_Rmean(0.5)	5	1	1	0	5
QaSIS_Rmedian(0.5)	7.5	3	1	0	5
QaSIS_Bmean(0.5)	7	5	1	0	5
QaSIS(0.75)	51.5	53.5	0.875	0.25	8
QaSIS_mean(0.75)	363.5	339.25	0.375	0	8
QaSIS_median(0.75)	45	70	0.875	0.25	8
QaSIS_Rmean(0.75)	17.5	26.75	1	0.03125	8
QaSIS_Rmedian(0.75)	35.5	68.25	1	0.25	8
QaSIS_Bmean(0.75)	172	388.75	0.75	0.375	8
NIS	1772.5	400	0.125	0.25	8
PNIS_mean	1731.5	444	0.125	0.25	8
PNIS_median	1745.5	570.75	0.125	0.25	8
PNIS_Rmean	1728.5	459.25	0.125	0.25	8
PNIS_Rmedian	1747	412.5	0.125	0.25	8
PNIS_Bmean	1759.5	334	0.125	0.25	8
SIRS	12	6.25	1	0	8
PSIRS_mean	15	13.25	1	0	8
PSIRS_median	13.5	10	1	0	8
PSIRS_Rmean	10	3	1	0	8
PSIRS_Rmedian	14	10	1	0	8
PSIRS_Bmean	238	763.25	0.75	0.25	8
SIS	1756	487.5	0.125	0.25	8
PSIS_mean	1586.5	532.5	0.125	0.375	8
PSIS_median	1649	502.5	0.125	0.28125	8
PSIS_Rmean	1611.5	539.25	0.125	0.25	8
PSIS_Rmedian	1639.75	583.375	0.125	0.25	8
PSIS_Bmean	1659.5	537	0.125	0.375	8

Table 36: Example 3, Parallel Ensemble, Error= Mixture Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.5)	7	3	1	0	5
PQaSIS_mean(0.5)	10	6	1	0	5
PQaSIS_median(0.5)	7	3	1	0	5
PQaSIS_Rmean(0.5)	5	1	1	0	5
PQaSIS_Rmedian(0.5)	7	3	1	0	5
PQaSIS_Bmean(0.5)	7	4	1	0	5

QaSIS(0.75)	68	71.5	0.875	0.375	8
PQaSIS_mean(0.75)	384.5	335.5	0.375	0	8
PQaSIS_median(0.75)	71	80.25	0.75	0.28125	8
PQaSIS_Rmean(0.75)	30	44.75	1	0.125	8
PQaSIS_Rmedian(0.75)	55.75	76.5	0.875	0.375	8
PQaSIS_Bmean(0.75)	331	1129	0.5	0.375	8
NIS	1726.5	588.5	0.25	0.28125	8
PNIS_mean	1610	606	0.25	0.25	8
PNIS_median	1584	607.25	0.25	0.25	8
PNIS_Rmean	1607	658.75	0.25	0.28125	8
PNIS_Rmedian	1623.5	696.875	0.25	0.25	8
PNIS_Bmean	1743.5	419.75	0.25	0.25	8
SIRS	15	8.25	1	0	8
PSIRS_mean	17.5	16	1	0	8
PSIRS_median	16	11.5	1	0	8
PSIRS_Rmean	12	6	1	0	8
PSIRS_Rmedian	16	12	1	0	8
PSIRS_Bmean	127.5	383.5	0.875	0.375	8
SIS	1622.5	656	0.25	0.375	8
PSIS_mean	1378.5	854.5	0.375	0.25	8
PSIS_median	1523.5	688.75	0.25	0.25	8
PSIS_Rmean	1418.5	708.5	0.3125	0.375	8
PSIS_Rmedian	1477.5	654.5	0.25	0.25	8
PSIS_Bmean	1673	679.5	0.375	0.25	8

Table 37: Example 3, Quantile Ensemble, Error= Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.25)	19	22	1	0	8
QaSIS(0.5)	6	2	1	0	5
QaSIS(0.75)	19	16	1	0	8
NIS	1739.5	455.75	0.25	0.25	8
SIRS	20.5	16.5	1	0	8
QES_mean	1248	598.25	0.25	0.125	8
QES_median	9	2	1	0	8
QES_Rmean	31.5	23.25	1	0.125	8
QES_Rmedian	9	4	1	0	8
QES_Bmean	10	3	1	0	8

Table 38: Example 3, Quantile Ensemble, Error= Cauchy

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.25)	63	95.25	0.875	0.15625	8
QaSIS(0.5)	7	2.25	1	0	5
QaSIS(0.75)	51.5	53.5	0.875	0.25	8
NIS	1772.5	400	0.125	0.25	8
SIRS	12	6.25	1	0	8
QES_mean	1621	464.75	0.125	0.25	8
QES_median	14	7	1	0	8
QES_Rmean	52	41.5	0.875	0.25	8
QES_Rmedian	11	3.625	1	0	8
QES_Bdmean	12	3	1	0	8

Table 39: Example 3, Quantile Ensemble, Error= Mix Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.25)	33.5	123.5	1	0.125	8
QaSIS(0.5)	7	3	1	0	5
QaSIS(0.75)	68	71.5	0.875	0.375	8
NIS	1726.5	588.5	0.25	0.28125	8
SIRS	15	8.25	1	0	8
QES_mean	1416	695.25	0.1875	0.125	8
QES_median	11	5	1	0	8
QES_Rmean	40	42	0.875	0.125	8
QES_Rmedian	10	3.5	1	0	8
QES_Bmean	10.5	4	1	0	8

Table 40: Example 3, Multi Ensemble, Error= Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.75)	19	16	1	0	8
NIS	1739.5	455.75	0.25	0.25	8
SIRS	20.5	16.5	1	0	8
MultiSIS_Rmean	302	165.5	0.625	0.375	8
MultiSIS_Rmedian	11	7	1	0	8

Table 41: Example 3, Multi Ensemble, Error= Cauchy

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.75)	51.5	53.5	0.875	0.25	8
NIS	1772.5	400	0.125	0.25	8
SIRS	12	6.25	1	0	8
MultiSIS_Rmean	319	159.5	0.5	0.25	8
MultiSIS_Rmedian	16.5	13.25	1	0	8

Table 42: Example 3, Multi Ensemble, Error= Mixture Normal

name	R	IQR(R)	S	IQR(S)	p^*
QaSIS(0.75)	68	71.5	0.875	0.375	8
NIS	1726.5	588.5	0.25	0.28125	8
SIRS	15	8.25	1	0	8
MultiSIS_mean	305.5	200.75	0.5	0.25	8
MultiSIS_median	25.5	25.75	1	0.125	8

Table 43: Example 3, Mixture Ensemble, Error= Normal

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.5)	6	2	1	0	5
QaSIS(0.75)	19	16	1	0	8
SIRS	20.5	16.5	1	0	8
NIS	1739.5	455.75	0.25	0.25	8
MixSIS_Rmean	232	193.25	0.625	0.25	8
MixSIS_Rmedian	11	6	1	0	8

Table 44: Example 3, Mixture Ensemble, Error= Cauchy

name	$\mathcal{R}$	IQR($\mathcal{R}$)	$\mathcal{S}$	IQR($\mathcal{S}$)	p^*
QaSIS(0.5)	7	2.25	1	0	5
QaSIS(0.75)	51.5	53.5	0.875	0.25	8
SIRS	12	6.25	1	0	8
NIS	1772.5	400	0.125	0.25	8
MixSIS_Rmean	310.5	152.75	0.5	0.25	8
MixSIS_Rmedian	10	4	1	0	8

Table 45: Example 3, Mixture Ensemble, Error= Mixture Normal

name	R	IQR(R)	S	IQR(S)	p^*
QaSIS(0.5)	7	3	1	0	5
QaSIS(0.75)	68	71.5	0.875	0.375	8
SIRS	15	8.25	1	0	8
NIS	1726.5	588.5	0.25	0.28125	8
MultiSIS_Mean	283	206.25	0.5	0.15625	8
MultiSIS_Median	10	5	1	0	8