

## Article

# Super RaSE: Super Random Subspace Ensemble Classification

Jianan Zhu<sup>1</sup> and Yang Feng<sup>2,\*</sup> <sup>1</sup> Department of Biostatistics, School of Global Public Health, New York University, New York, NY 10003; jz4698@nyu.edu<sup>2</sup> Department of Biostatistics, School of Global Public Health, New York University, New York, NY 10003; yang.feng@nyu.edu

\* Correspondence: yang.feng@nyu.edu

**Abstract:** We propose a new ensemble classification algorithm, named *Super Random Subspace Ensemble* (Super RaSE), to tackle the sparse classification problem. The proposed algorithm is motivated by the Random Subspace Ensemble algorithm (RaSE) (Tian and Feng 2021b). The RaSE method was shown to be a flexible framework that can be coupled with any existing base classification. However, the success of RaSE largely depends on the proper choice of the base classifier, which is unfortunately unknown to us. In this work, we show that Super RaSE avoids the need to choose a base classifier by randomly sampling a collection of classifiers together with the subspace. As a result, Super RaSE is more flexible and robust than RaSE. In addition to the vanilla Super RaSE, we also develop the iterative Super RaSE, which adaptively changes the base classifier distribution as well as the subspace distribution. We show the Super RaSE algorithm and its iterative version perform competitively for a wide range of simulated datasets and two real data examples. The new Super RaSE algorithm and its iterative version are implemented in a new version of the R package **RaSEn**.

**Keywords:** classification; ensemble; subspace; sparsity; feature ranking

## 1. Introduction

Ensemble learning is a popular machine learning framework, which combines multiple learning algorithms to obtain better prediction performance and increase the stability of any single algorithm (Dietterich 2000; Rokach 2010). Some popular examples include bagging (Breiman 1996) and random forests (Breiman 2001), which aggregates a collection of weak learners formed by decision trees. More recent ensemble learning methods include the random subspace method (Ho 1998), super learner (Van der Laan et al. 2007), model averaging (Feng et al. 2021; Raftery et al. 1997), random rotation (Blaser and Fryzlewicz 2016), random projection (Cannings and Samworth 2017; Durrant and Kabán 2015), and random subspace ensemble classification (Tian and Feng 2021ab).

This paper is largely motivated by the Random Subspace Ensemble (RaSE) classification framework (Tian and Feng 2021b), which we will briefly review. Suppose we want to predict the class label  $y$  from the feature vector  $x$ . For a given base classifier, the RaSE algorithm aims to construct  $B_1$  weak learner, where each classifier is formed by applying the specified base classifier on a properly chosen subspace. To choose each subspace,  $B_2$  random subspaces are generated and the optimal one is selected according to certain criteria (e.g., cross-validation error). In the end, the predicted labels from the  $B_1$  weak learners are averaged and compared to a data-driven threshold, forming the final classifier. Tian and Feng (2021b) also proposed an iterative version of RaSE which updates the random subspace distribution according to the selected proportions of each feature.

Powerful the RaSE algorithm and its iterative versions are, one major limitation is that one needs to specify a single base classifier *prior* to using the RaSE framework. The success of the RaSE algorithms largely depends on whether the base classifier is suitable

for the application scenario. As shown in the numerical experiments in [Tian and Feng \(2021b\)](#), the RaSE algorithm could fail to work well if the base classifier is not properly set. In this regards, blindly applying the RaSE algorithm has the risk of wrongly set the base classifier, leading to a poor performance. In this work, we relax this requirement by replacing a single base classifier with a *collection of base classifiers*. We call the new ensemble classification framework the *Super Random Subspace Ensemble (Super RaSE)*.

The working mechanism of Super RaSE is that in addition to randomly generating the subspaces, it also generates the base classifiers to be used together with the subspaces. More specifically, instead of fixing a base classifier and using it for all subspaces, each time the Super RaSE randomly generates the base classifier (from a collection of base classifiers) and the subspace as a pair, and then picks the best performing pair among the  $B_2$  ones via five-fold cross-validation to form one of the  $B_1$  weak learners. Then, the predictions of the  $B_1$  weak learners are averaged and compared to a data-driven threshold, forming the final prediction of Super RaSE.

The main contribution of the paper is three-fold. First, the Super RaSE algorithm adaptively chooses the base classifier and subspace pair, which makes it a fully model-free approach. Second, in addition to the accurate prediction performance, the Super RaSE computes the selected proportion for each base classifier among the  $B_1$  weak learners, implying the appropriateness of each base classifier under the specific scenario, and for each of the base classifier, the selected proportion of each feature, measuring the importance of each feature. Third, we propose an iterative Super RaSE algorithm, which updates the sampling distribution of base classifiers as well as the sampling distribution of the subspaces for each base classifier.

The rest of the paper is organized as follows. In Section 2, we introduce the super random subspace ensemble (SRaSE) classification algorithm as well as its iteration version. Section 3 conducts extensive simulation studies to show the superior performance of SRaSE and its iterative version by comparing them with competing methods including the original RaSE algorithm. In Section 4, we evaluate the SRaSE algorithms with two real datasets and show they perform competitively. Lastly, we conclude the paper with a short discussion in Section 5.

## 2. Methods

Suppose that we have  $n$  pairs of observations  $\{(x_i, y_i), i = 1, \dots, n\} \stackrel{i.i.d.}{\sim} (x, y) \in \mathbb{R}^p \times \{0, 1\}$ , where  $p$  is the number of predictors and  $y \in \{0, 1\}$  is the class label. We use  $S_{\text{Full}} = \{1, \dots, p\}$  to represent the whole feature set. We assume the marginal densities of  $x$  for class 0 ( $y = 0$ ) and 1 ( $y = 1$ ) exist and are denoted as  $f^{(0)}$  and  $f^{(1)}$ , respectively. Thus, the joint distribution of  $(x, y)$  can be described in the following mixture model

$$x|y = y_0 \sim (1 - y_0)f^{(0)} + y_0f^{(1)}, y_0 = 0, 1,$$

where  $y$  is a Bernoulli variable with success probability  $\pi_1 = 1 - \pi_0 \in (0, 1)$ . For any subspace  $S$ , we use  $|S|$  to denote its cardinality. When restricting to the feature subspace  $S$ , the corresponding marginal densities of class 0 and 1 are denoted as  $f_S^{(0)}$  and  $f_S^{(1)}$ , respectively.

Here, we are concerned with a high-dimension classification problem where the dimensional  $p$  is comparable or even larger than the sample size  $n$ . In high-dimensional problems, we usually believe there are only a handful of features that contribute to the response, which is usually referred to as the *sparse classification problem*. For sparse classification problems, it is of significance to accurately separate signals from noises. Following [Tian and Feng \(2021b\)](#), we introduce the definition of a discriminant set.

A feature subset  $S$  is called a *discriminative set* if  $y$  is conditionally independent with  $x_{S^c}$  given  $x_S$ , where  $S^c = S_{\text{Full}} \setminus S$ . We call  $S$  a *minimal discriminative set* if it has minimal cardinality among all discriminative sets, and we denote it as  $S^*$ .

### 2.1. Super Random Subspace Ensemble classification (SRaSE)

Here, to train each weak learner (e.g., the  $j$ -th one),  $B_2$  independent random subspaces are generated as  $\{S_{j1}, \dots, S_{jB_2}\}$  from subspace distribution  $\mathcal{D}$  and  $B_2$  base classifiers  $\{T_{j1}, \dots, T_{jB_2}\}$  are sampled with replacement from base classifier distribution  $\mathbb{D}$  on the candidate base classifier set  $\mathcal{T} = \{T_1, \dots, T_M\}$ . By default, we will be using a uniform distribution for  $\mathbb{D}$ . However, users can use other distributions in the algorithm if they have some prior belief on which classifiers may work better. Then, for  $k = 1, \dots, B_2$ , we train the classifier  $T_{jk}$  using only the features in  $S_{jk}$ . We then choose the optimal subspace  $S_{j*}$  and base classifier  $T_{j*}$  pair using 5-fold cross-validation. We denote the base classifier  $T_{j*}$  applied on subspace  $S_{j*}$  as  $C_n^{T_{j*}-S_{j*}}$ , where the subscript  $n$  is used to emphasize the classifier depends on the sample with  $n$  observations. Finally, we aggregate outputs of  $\{C_n^{T_{j*}-S_{j*}}\}_{j=1}^{B_1}$  to form the final decision function by taking a simple average. The whole procedure can be summarized in Algorithm 1.

---

#### Algorithm 1: Super Random Subspace Ensemble classification (SRaSE)

---

**Input:** training data  $\{(x_i, y_i)\}_{i=1}^n$ , new data  $x$ , subspace distribution  $\mathcal{D}$ , integers  $B_1$  and  $B_2$ , the candidate base classifier set  $\mathcal{T}$ , base classifier distribution  $\mathbb{D}$

**Output:** predicted label  $C_n^{RaSE}(x)$ , the selected proportion of each base classifier  $\zeta$ , and for the base classifier  $T_i \in \mathcal{T}$  where  $i \in \{1, \dots, M\}$ , the selected proportion of each feature  $\eta_i$

- 1 Independently generate base classifiers  $T_{jk} \sim \mathbb{D}, 1 \leq j \leq B_1, 1 \leq k \leq B_2$
  - 2 Independently generate random subspaces  $S_{jk} \sim \mathcal{D}, 1 \leq j \leq B_1, 1 \leq k \leq B_2$
  - 3 **for**  $j \leftarrow 1$  **to**  $B_1$  **do**
  - 4     Select the optimal subspace and base classifier pair  $(T_{j*}, S_{j*})$  from  $\{(T_{jk}, S_{jk})\}_{k=1}^{B_2}$  using 5-fold cross-validation.
  - 5 **end**
  - 6 Construct the ensemble decision function  $v_n(x) = B_1^{-1} \sum_{j=1}^{B_1} C_n^{T_{j*}-S_{j*}}(x)$
  - 7 Set the threshold  $\hat{\alpha}$  according to (1)
  - 8 Compute the selected proportion of each method  $\zeta = (\zeta_1, \dots, \zeta_M)^T$ , where  $\zeta_i = B_1^{-1} \sum_{j=1}^{B_1} \mathbb{1}(i \in T_{j*})$
  - 9 For each method  $T_i, i = 1, \dots, M$ , compute the selected proportion of each feature  $\eta_i = (\eta_{i1}, \dots, \eta_{ip})^T$ , where  $\eta_{il} = (\zeta_i B_1)^{-1} \sum_{j=1}^{B_1} \mathbb{1}(i \in T_{j*}) \mathbb{1}(l \in S_{j*}), l = 1, \dots, p$
  - 10 Output the predicted label  $C_n^{RaSE}(x) = \mathbb{1}(v_n(x) > \hat{\alpha})$ , the selected proportion of each method  $\zeta = (\zeta_1, \dots, \zeta_M)^T$ , and the selected proportion of each feature for each method  $\eta_i = (\eta_{i1}, \dots, \eta_{ip})^T$
- 

Following Tian and Feng (2021b), by default, the subspace distribution  $\mathcal{D}$  is chosen as a *hierarchical uniform distribution* over the subspaces. In particular, with  $D$  as the upper bound of the subspace size, we first generate the subspace size  $d$  from the discrete uniform distribution over  $\{1, \dots, D\}$ . Then, the subspaces  $\{S_{jk}, j = 1, \dots, M, k = 1, \dots, p\}$  are independent and follow the uniform distribution over the set  $\{S \subseteq S_{\text{Full}} : |S| = d\}$ . In addition, in Step 7 of Algorithm 1, we choose the decision threshold to minimize the empirical classification error on the training set,

$$\hat{\alpha} = \arg \min_{\alpha \in [0,1]} [\hat{\pi}_0(1 - \hat{G}_n^{(0)}(\alpha)) + \hat{\pi}_1 \hat{G}_n^{(1)}(\alpha)], \quad (1)$$

where

$$\begin{aligned} n_r &= \sum_{i=1}^n \mathbb{1}(y_i = r), r = 0, 1, \\ \hat{\pi}_r &= \frac{n_r}{n}, r = 0, 1, \\ \hat{G}_n^{(r)}(\alpha) &= \frac{1}{n_r} \sum_{i=1}^n \mathbb{1}(y_i = r) \mathbb{1}(v_n(\mathbf{x}_i) \leq \alpha), r = 0, 1, \\ v_n(\mathbf{x}) &= B_1^{-1} \sum_{j=1}^{B_1} C_n^{T_{j*} - S_{j*}}(\mathbf{x}). \end{aligned}$$

In Algorithm 1, there are two important by-products. The first one is the selected proportion of each method  $\zeta = (\zeta_1, \dots, \zeta_M)^T$  out of the  $B_1$  weak learners. The higher the proportion for a method (e.g. KNN) is, the more appropriate it may be for the particular data. In numerical studies and real data analyses, we will provide more interpretations of the results.

Now, let's introduce the second by-product of Algorithm 1. For each method  $T_i, i = 1, \dots, M$ , we have the selected proportion of each feature  $\eta_i = (\eta_{i1}, \dots, \eta_{ip})^T$ . The feature selection proportion depends on the particular base method. The underlying reason is that when we use different base methods on the same data, different signals may be found. For example, if some predictors contribute to the response only through an interaction effect with other predictors, they may be detected using the Quadratic Discriminant Analysis (QDA) method, however, they can't be identified using the Linear Discriminant Analysis (LDA) method since the LDA only considers the linear effects of features. This base-classifier-dependent feature selection frequencies will produce a better understanding of the working mechanism of each base classifier as well as the nature of the importance for each feature.

## 2.2. Iterative Super RaSE

Motivated by the fact that some base classifiers may be more appropriate under the given scenario, it could be a good idea to leverage the information we learned during the Super RaSE algorithm. In particular, the selected proportion of each method  $\zeta$  could be used to adjust the sampling distribution of the methods. In the new iterative algorithm, we will update the base classifier distribution  $\mathbb{D}$  by setting the probability mass function as the selected proportion of each method from the last iteration. This updating scheme will make the better-performing classifiers have a higher chance of being sampled in the next iteration, which could potentially increase the quality of the  $B_1$  classifiers. On the other hand, if a certain base classifier was rarely selected in the  $B_1$  classifiers, perhaps we want to down-weight it in the next iteration.

In addition, regarding the feature selection proportion for each base method, we will adopt a similar updating strategy as that in the original RaSE (Tian and Feng 2021b). However, in Super RaSE, we are updating the random subspace distribution separately for each base method. For base classifier  $i$ , the initial subspace distribution is denoted by  $\mathcal{D}_i^{(0)}$ , where the subscript  $i$  represents the  $i$ -th base classifier and superscript (0) represented the 0-th iteration. In the  $t$ -th iteration, we will use  $\mathcal{D}_i^{(t)}$  to represent the distribution of the subspace distribution for base classifier  $i$ .

The updating scheme for the subspace distribution is as follows. For method  $i$ , we again first generate the subspace size  $d$  from the uniform distribution over  $\{1, \dots, D\}$  as before. It is easy to observe that each subspace  $S$  can be equivalently represented as a binary  $p$ -dimensional vector representing whether each feature is included in  $S$ . To be more specific, the equivalent vector representation subspace  $S$  is  $\mathbf{J} = (J_1, \dots, J_p)^T$ , where  $J_l = \mathbb{1}(l \in S), l = 1, \dots, p$ . Then, we will generate  $\mathbf{J}$  from a *restrictive multinomial distribution* with parameter  $(p, d, \tilde{\eta}_i)$ , where  $\tilde{\eta}_i = (\tilde{\eta}_{i1}, \dots, \tilde{\eta}_{ip})^T, \tilde{\eta}_{il} = \eta_{il} \mathbb{1}(\eta_{il} >$

**Algorithm 2:** Iterative Super RaSE (SRaSE<sub>T</sub>)

---

**Input:** training data  $\{(x_i, y_i)\}_{i=1}^n$ , new data  $x$ , integers  $B_1$  and  $B_2$ , the candidate base classifier set  $\mathcal{T}$ , initial base classifier distribution  $\mathbb{D}^{(0)}$ , initial subspace distribution for each base classifier  $\{\mathcal{D}_i^{(0)}, i = 1, \dots, T\}$ , the number of iterations  $T$

**Output:** predicted label  $C_n^{\text{SRaSE}_T}(x)$ , the selected proportion of each base classifier  $\zeta^{(T)}$ , and for the base classifier  $T_i \in \mathcal{T}$  where  $i \in \{1, \dots, M\}$ , the selected proportion of each feature  $\eta_i^{(T)}$

---

```

1 for  $t \leftarrow 0$  to  $T$  do
2   Independently generate base classifiers  $T_{jk} \sim \mathbb{D}^{(t)}, 1 \leq j \leq B_1, 1 \leq k \leq B_2$ 
3   Independently generate random subspaces
4      $S_{jk}^{(t)} \sim \mathcal{D}_i^{(t)}, 1 \leq j \leq B_1, 1 \leq k \leq B_2$ 
5   for  $j \leftarrow 1$  to  $B_1$  do
6     Select the optimal subspace and base classifier pair  $(S_{j*}^{(t)}, T_{j*}^{(t)})$  from
7        $\{(S_{jk}^{(t)}, T_{jk}^{(t)})\}_{k=1}^{B_2}$  using 5-fold cross-validation.
8   end
9   Set  $\mathbb{D}^{(t+1)}$  to be a discrete distribution over the candidate base classifier set
10     $\mathcal{T}$ , where for each base classifier  $T_i \in \mathcal{T}$ ,  $P(T_i) = \zeta_i^{(t)}$ , where
11     $\zeta_i^{(t)} = B_1^{-1} \sum_{j=1}^{B_1} \mathbb{1}(i \in T_{j*}^{(t)})$ 
12  For each method  $T_i, i = 1, \dots, M$ , compute
13     $\eta_{il}^{(t)} = (\zeta_i^{(t)} B_1)^{-1} \sum_{j=1}^{B_1} \mathbb{1}(i \in T_{j*}^{(t)}) \mathbb{1}(l \in S_{j*}^{(t)}), l = 1, \dots, p$ 
14  Set  $\mathcal{D}^{(t+1)}$  to be a restrictive multinomial distribution with parameter
15     $(p, d, \tilde{\eta}_i^{(t)})$ , where  $\tilde{\eta}_{il}^{(t)} = \eta_{il}^{(t)} \mathbb{1}(\eta_{il}^{(t)} > C_0 / \log p) + \frac{C_0}{p} \mathbb{1}(\eta_{il}^{(t)} \leq C_0 / \log p)$ 
16    and  $d$  is sampled from the uniform distribution over  $\{1, \dots, D\}$ 
17 end
18 Set the threshold  $\hat{\alpha}$  according to (1)
19 Construct the ensemble decision function  $v_n(x) = \frac{1}{B_1} \sum_{j=1}^{B_1} C_n^{S_{j*}^{(T)} - T_{j*}^{(T)}}(x)$ 
20 Output the predicted label  $C_n^{\text{SRaSE}_T}(x) = \mathbb{1}(v_n(x) > \hat{\alpha})$ , the selected
21   proportion of each base classifier  $\zeta^{(T)} = (\zeta_1^{(T)}, \dots, \zeta_M^{(T)})^T$ , and the selected
22   proportion of each feature for each base classifier  $\eta_i^{(T)} = (\eta_{i1}^{(T)}, \dots, \eta_{ip}^{(T)})^T$ 

```

---

$C_0 / \log p) + \frac{C_0}{p} \mathbb{1}(\eta_{il} \leq C_0 / \log p)$ , and the restriction is that  $I_l \in \{0, 1\}, l = 1, \dots, p$ , where  $C_0$  is a constant. Note that here, the parameter  $\tilde{\eta}_i$  characterize the subspace distribution.

We named the iterative algorithm *Iterative Super RaSE*, the details of which are summarized in Algorithm 2.

In the iterative Super RaSE algorithm, the base classifier distribution is initially set to be  $\mathbb{D}^{(0)}$ , which is a uniform distribution over all base classifiers by default. As the iteration proceeds, the base classifiers that are more frequently selected will have a higher chance of being selected in the next step, resulting in a different  $\mathbb{D}^{(t)}$ . The adaptive nature of the iterative Super RaSE algorithm will enable us to discover the best performing base methods for each dataset and in turn, reduces its classification error.

Besides the base classifier distribution, the subspace distribution is also continuously updated during the iteration process. In our implementation of the algorithm, the initial subspace distribution for each base classifier  $\mathcal{D}_i(0)$  is the hierarchical uniform distribution as introduced in Section 2.1. After running the Super RaSE algorithm once, the features that are more frequently selected are given higher weights in  $\mathcal{D}_i(1)$ . In



this mechanism, we give an edge to the useful features, which could further boost the performance. In addition, for each given base classifier, the selected frequencies of each feature can be viewed as an importance measure of the features.

### 2.3. Parameter Specification

In the Super RaSE algorithm and its iterative version, we have discussed the base classifier distribution and the subspace distribution for each base classifier in Sections 2.1 and 2.2. There are a few additional parameters that need to be specified. We now discuss the details with the default values in the algorithms.

- $B_1$  is the number of weak classifiers we want to construct during the algorithm and average over at the end. It is set to be 200 by default. Usually, the larger the  $B_1$  is, the more robust the algorithm is.
- $B_2$  is the number of base classifier and subspace pairs among which we choose the optimal one. It is set to be 500 by default. To make sure each base classifier and the subspace combination has a fair chance of being evaluated,  $B_2$  has to be reasonably large.
- $D$  is the maximum subspace size in the subspace distribution. Following [Tian and Feng \(2021b\)](#), we set  $D = \min(p, \lfloor \sqrt{n_0} \rfloor, \lfloor \sqrt{n_1} \rfloor)$  for QDA base classifier and  $D = \min(p, \lfloor \sqrt{n} \rfloor)$  for all the other base classifiers, where  $\lfloor a \rfloor$  denotes the largest integer not larger than  $a$ .
- $\mathcal{T}$  is the candidate base classifier set. In theory, it could contain any classifier and the Super RaSE algorithm will automatically select the better-performing ones in the final  $B_1$  weak learners. On the other hand, the more base classifiers we put inside  $\mathcal{T}$ , each base classifier would have a smaller chance of being sampled in the  $B_2$  pairs of base classifier and subspace, leading to an increased risk of missing the optimal combination. In our implementation, we set the candidate base classifier set  $\mathcal{T} = \{\text{LDA}, \text{QDA}, \text{KNN}\}$ .
- $T$  is the number of iterations in the iterative Super RaSE. From our limited numerical experience, the first two rounds of iterations lead to the most performance improvement. As a result, we use  $T = 2$  for simulation and  $T = 1$  for real data analysis.

## 3. Simulation Studies

In this section, we conduct extensive simulation studies on the proposed Super RaSE algorithm (Algorithm 1) and its iterative version (Algorithm 2) with candidate base classifiers set  $\mathcal{T} = \{\text{LDA}, \text{QDA}, \text{KNN}\}$ . In addition, we compare their performances with several competing methods including the original RaSE with LDA, QDA, and KNN as the base classifier ([Tian and Feng 2021b](#)), as well as LDA, QDA, KNN, and Random Forest (RF). We use the default values for all parameters in the Super RaSE algorithm and its iterative version ( $B_1 = 200$ ,  $B_2 = 500$ ).

For all experiments, we conducted 200 replicates, and report the summary of test errors (in percentage) in terms of mean and the standard deviation. We use boldface to highlight the method with minimal test error for each setting, and use italics to highlight the methods that achieve test errors within one standard deviation of the smallest error.

### 3.1. Model 1: LDA

The first model we consider is a sparse LDA model with settings in [Mai et al. \(2012\)](#) and [Tian and Feng \(2021b\)](#). In particular, assume  $\mathbf{x}|y = r \sim N(\boldsymbol{\mu}^{(r)}, \Sigma)$ ,  $r = 0, 1$ , where  $\Sigma = (\Sigma_{ij})_{p \times p} = (0.5^{|i-j|})_{p \times p}$ ,  $\boldsymbol{\mu}^{(0)} = \mathbf{0}_{p \times 1}$ ,  $\boldsymbol{\mu}^{(1)} = \Sigma \times 0.556(3, 1.5, 0, 0, 2, \mathbf{0}_{1 \times (p-5)})^T$ . The dimension  $p = 400$ , and we vary the training sample size  $n \in \{200, 400, 1000\}$ . We also independently generate a test data of size 1000.

It is easy to verify that the feature subset  $\{1, 2, 5\}$  is the minimal discriminative set  $S^*$ . In Table 1, the performances of various methods for model 1 under different sample sizes are presented.

Table 1: Summary of test classification error rates for each classifier under various sample sizes over 200 repetitions in Model 1 (LDA). The results are presented as mean values with the standard deviation values in parentheses

	$n = 200$	$n = 400$	$n = 1000$
SRaSE	12.57(1.61)	11.42(1.18)	10.78(1.02)
SRaSE <sub>1</sub>	11.44(1.36)	10.68(1.14)	10.34(0.96)
SRaSE <sub>2</sub>	11.58(1.38)	10.79(1.06)	10.37(0.95)
RaSE-LDA	12.99(1.42)	12.38(1.20)	11.16(1.10)
RaSE-QDA	13.78(1.26)	13.69(1.19)	13.23(1.04)
RaSE-KNN	13.21(1.48)	12.57(1.23)	11.19(1.10)
RaSE <sub>1</sub> -LDA	11.48(1.26)	10.42(1.07)	10.25(1.08)
RaSE <sub>1</sub> -QDA	11.28(1.47)	10.76(1.25)	10.4(1.22)
RaSE <sub>1</sub> -KNN	<b>11.11</b> (1.25)	10.58(1.09)	10.33(1.05)
RaSE <sub>2</sub> -LDA	12.62(1.45)	11.41(1.16)	10.13(1.03)
RaSE <sub>2</sub> -QDA	11.74(1.45)	<b>10.39</b> (1.01)	<b>10.09</b> (1.07)
RaSE <sub>2</sub> -KNN	11.17(1.31)	10.6(0.99)	10.34(1.02)
LDA	NA(NA)	46.39(2.62)	18.62(1.53)
QDA	NA(NA)	NA(NA)	47.74(1.73)
KNN	29.08(2.77)	26.73(1.97)	24.53(1.60)
RF	12.66(1.43)	11.77(1.04)	11.25(1.10)

As we could see, RaSE<sub>1</sub>-KNN performs the best when the sample size  $n = 200$ , and RaSE<sub>2</sub>-QDA performs the best for  $n = 400$  and  $1000$ . It is worth noting the results for LDA and QDA are NA due to the small sample size compared with the dimension. By inspecting the performances of the proposed Super RaSE algorithm and its iterative version, we can see that although they are not the best performing method, both SRaSE<sub>1</sub> and SRaSE<sub>2</sub> are within one standard error of the best performing method, showing the robustness of Super RaSE. And clearly, one iteration helps Super RaSE to have a lower test classification error.

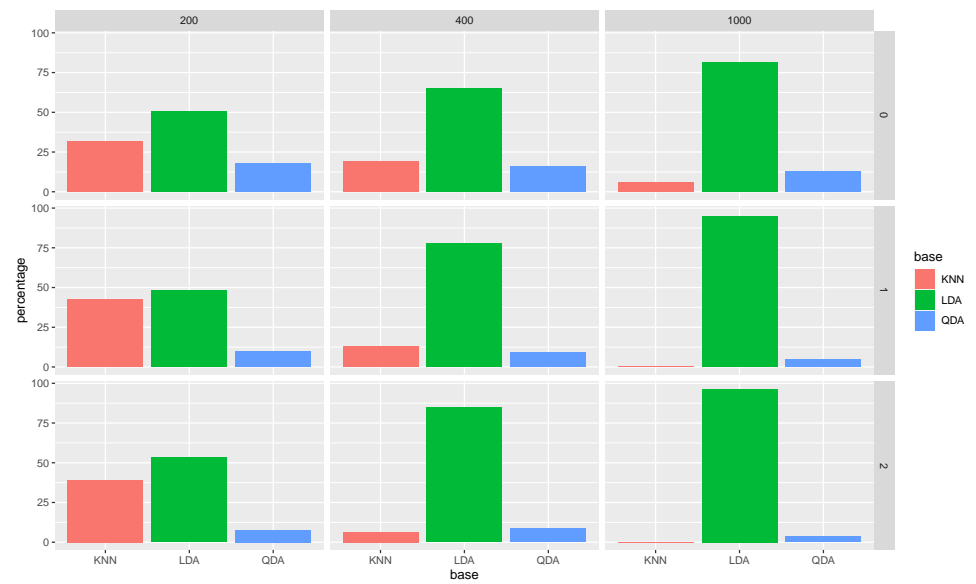
In addition to the test classification error, it is useful to investigate the two by-products of our algorithms, namely the selected proportion of each base classifier among the  $B_1$  classifiers, and the selected proportion of each feature among the weak learners that use a particular classifier.

Let's take a look at Figure 1. The first row shows the bar charts for the selection percentage of each base classifier in the Super RaSE algorithm among the 200 repetitions, when the sample size  $n$  varies in  $\{200, 400, 1000\}$ . It shows when  $n = 200$ , the percentage of LDA is around 50%. As the sample size increases, the percentage of LDA also increases, showing that having a larger sample size helps us to select the model from which the data is generated.

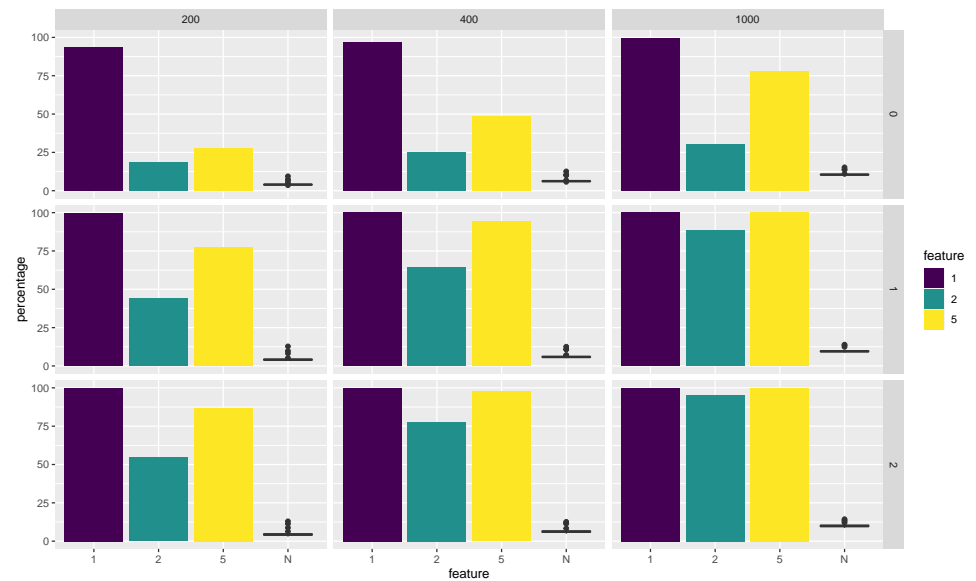
Now, let's look at the column of  $n = 1000$ , we can see that as the iteration process moves on, the percentage of LDA is increasing as well, leading to almost 100% for SRaSE<sub>2</sub>.

The second product of the Super RaSE algorithm and its iterative version is the selected frequencies for each feature among the weaker learner with a particular classifier. Figure 2 visualizes the selected proportions of features among all the  $B_1$  classifiers that use LDA as the base classifier. In particular, we show the selected proportions for each feature in the minimum discriminative set  $S^* = \{1, 2, 5\}$ . In the same figure, we also show a boxplot of the selected proportion of all the noisy features, as a way to verify whether the Super RaSE algorithms can distinguish the important features from the noisy features.

From Figure 2, we observe that when  $n = 200$ , the vanilla Super RaSE algorithm doesn't select the important variables 2 and 5 with a high percentage. However, the



**Figure 1.** The average selected proportion for each base method for different sample sizes (corresponding to each column) and iteration number (corresponding to each row) in Model 1 (LDA).



**Figure 2.** The average selected proportion for each feature for different sample sizes (corresponding to each column) and iteration number (corresponding to each row) in Model 1 (LDA).

iteration greatly helps the algorithm to increase the selected percentages for features 2 and 5. In addition, the increase of sample size leads to the selection of all important features with almost 100% percentage. It is also worth noting that the noise features all have a relatively small selection frequency, showing the power of feature ranking in the Super RaSE algorithms. Similar figures can also be generated for the selected proportions of features among the  $B_1$  classifiers that use QDA and the KNN as the base classifier, respectively. For simplicity, we omit these figures in our presentation.

### 3.2. Model 2: QDA

Now, we consider the case where the data is generated from a sparse QDA model with the following setting (Fan et al. 2015; Tian and Feng 2021b).



$x|y = r \sim N(\mu^{(r)}, \Sigma^{(r)})$ ,  $r = 0, 1$ , where  $\Omega^{(0)} = (\Sigma^{(0)})^{-1}$  is a  $p \times p$  banded matrix with  $(\Omega^{(0)})_{ii} = 1$  and  $(\Omega^{(0)})_{ik} = 0.3$  for  $|i - k| = 1$  and all other entries zero. The inverse covariance matrix for class 1 is  $\Omega^{(1)} = \Omega^{(0)} + \Omega$ , where  $\Omega$  is a  $p \times p$  sparse symmetric matrix with  $\Omega_{10,10} = -0.3758$ ,  $\Omega_{10,30} = 0.0616$ ,  $\Omega_{10,50} = 0.2037$ ,  $\Omega_{30,30} = -0.5482$ ,  $\Omega_{30,50} = 0.0286$ ,  $\Omega_{50,50} = -0.4614$  and all other entries zero. The dimension is  $p = 200$  and we again vary the sample size  $n \in \{200, 400, 1000\}$ .

We can easily verify that the feature subset  $\{1, 2, 10, 30, 50\}$  is the minimal discriminative set  $S^*$ . In Table 2, we present the performance of different methods for model 2 under different sample sizes. From the table, when  $n = 400$  and  $n = 1000$ , SRaSE<sub>2</sub>

Table 2: Summary of test classification error rates for each classifier under various sample sizes over 200 repetitions in Model 2 (QDA). The results are presented as mean values with the standard deviation values in parentheses

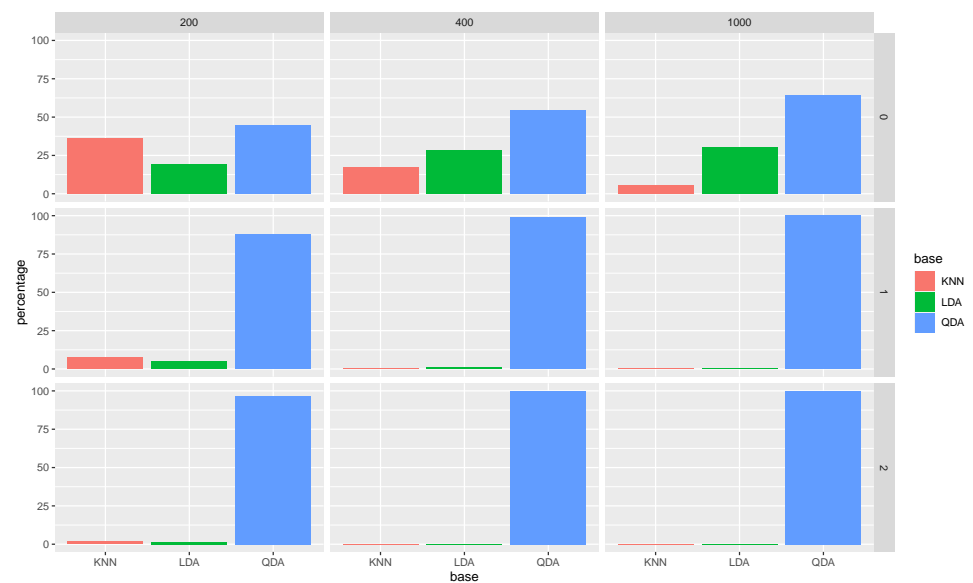
	$n = 200$	$n = 400$	$n = 1000$
SRaSE	30.82(3.29)	28.68(3.23)	26.12(2.66)
SRaSE <sub>1</sub>	27.58(2.33)	24.64(1.85)	23.03(1.38)
SRaSE <sub>2</sub>	27.36(2.67)	<b>24.04</b> (1.74)	<b>22.63</b> (1.41)
RaSE-LDA	37.3(3.17)	36.11(1.97)	35.67(1.73)
RaSE-QDA	32.52(2.90)	30.44(2.60)	29(1.97)
RaSE-KNN	31.1(3.23)	27.83(2.41)	25.22(1.56)
RaSE <sub>1</sub> -LDA	36.09(2.87)	32.82(1.74)	32.68(1.49)
RaSE <sub>1</sub> -QDA	<b>26.83</b> (2.47)	25.07(1.89)	23.53(1.50)
RaSE <sub>1</sub> -KNN	28.76(2.60)	25.88(1.98)	24.18(1.47)
RaSE <sub>2</sub> -LDA	38.09(2.48)	33.69(1.83)	32.71(1.55)
RaSE <sub>2</sub> -QDA	26.99(2.68)	24.87(1.99)	23.11(1.60)
RaSE <sub>2</sub> -KNN	28.73(2.56)	25.46(1.82)	23.76(1.54)
LDA	49.03(1.94)	42.88(1.82)	38.68(1.70)
QDA	NA	NA	45.13(1.58)
KNN	45.67(1.78)	44.63(2.02)	43.43(1.63)
RF	37.34(2.91)	31.61(2.19)	27.42(1.60)

is the best performing method, achieving the smallest test classification error. When  $n = 200$ , SRaSE<sub>2</sub> is within one standard error away from the optimal method. This result is very encouraging in the sense that the Super RaSE could improve the performance of the original RaSE coupled with the true model from which the data is generated. This shows the Super RaSE algorithms are extremely robust and avoid the need to choose a base classifier as needed in the original RaSE algorithm.

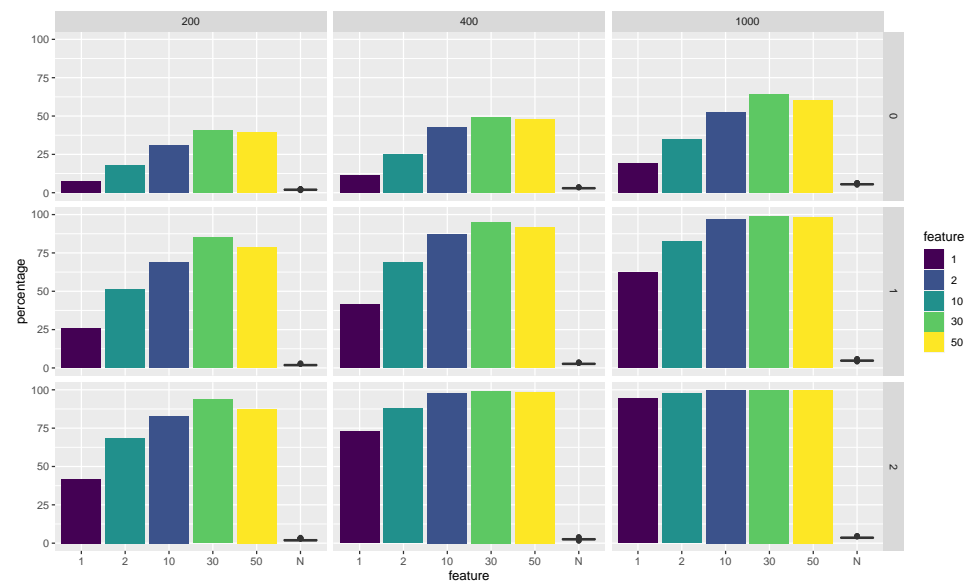
As in Model 1, we again present the average selected proportion of each base classifier in Figure 3 as well as the average selected proportion of each feature among the  $B_1$  weak learners with the chosen base classifier being QDA in Figure 4. In Figure 4, we also show a boxplot of the selected proportion of all the noisy features.

In Figure 3, we again observe that as we iterate the Super RaSE algorithm, the proportion of QDA greatly increases, reaching almost 100 percent for RaSE<sub>2</sub> over all sample sizes. This shows the iterative Super RaSE algorithm is able to identify the true model.

In Figure 4, we can observe the average selected proportions of each important feature are pretty high, showing that the Super RaSE is able to pick up the important features. In particular, when  $n = 1000$ , the iteration helps all the features to have a higher selected proportion, reaching nearly 100 percent for SRaSE<sub>2</sub>.



**Figure 3.** The average selected proportion for each base method for different sample sizes (corresponding to each column) and iteration number (corresponding to each row) in Model 2 (QDA).



**Figure 4.** The average selected proportion for each feature for different sample size (corresponding to each column) and iteration number (corresponding to each row) in Model 2 (QDA).

### 3.3. Model 3: KNN

Having considered two parametric models, let's move to an example where the label is generated in a nonparametric way (Tian and Feng 2021b). We will be following the idea of nearest neighbors when assigning the labels.

The detailed data generation process is as follows. First, 10 initial points  $z_1, \dots, z_{10}$  are generated i.i.d. from  $N(\mathbf{0}_{p \times 1}, I_p)$ . Out of the 10 points, five are labeled as 0 and the other five are labeled as 1. Then, each observation  $(x, y)$  pair is generated as follows. First, we randomly select one element from  $\{z_1, \dots, z_{10}\}$ . Let's suppose it is  $z_{k_i}$ , corresponding to the  $i$ -th observation. Then, the corresponding  $y_i$  will take the label as  $z_{k_i}$ , and the feature vector  $x_i$  is generated as  $x_i \sim N((z_{k_i, S^*}^T, \mathbf{0}_{1 \times (p-5)})^T, 0.5^2 I_p)$ . The general idea is like we are generating a mixture of ten Gaussian clusters that surround each of the  $\{z_1, \dots, z_{10}\}$  when they are embedded in a  $p$ -dimensional space.

From the data generation process, the minimal discriminative set is  $S^* = \{1, 2, 3, 4, 5\}$ . We consider  $p = 200$ , and  $n \in \{200, 400, 1000\}$ . The Summary of test classification error rates over 200 repetitions is presented in Table 3.

Table 3: Summary of test classification error rates for each classifier under various sample sizes over 200 repetitions in Model 3 (KNN). The results are presented as mean values with the standard deviation values in parentheses

	$n = 200$	$n = 400$	$n = 1000$
SRaSE	16.89(6.10)	13.54(5.39)	10.68(4.74)
SRaSE <sub>1</sub>	8.18(4.18)	6.86(3.58)	6.12(3.24)
SRaSE <sub>2</sub>	7.22(3.82)	6.39(3.42)	5.78(3.09)
RaSE-LDA	27.38(9.53)	25.08(8.40)	24.92(9.05)
RaSE-QDA	24.24(7.20)	22.62(6.77)	22.04(6.73)
RaSE-KNN	13.26(5.03)	10.67(4.44)	8.85(4.05)
RaSE <sub>1</sub> -LDA	25.89(10.13)	23.05(8.49)	23.42(8.95)
RaSE <sub>1</sub> -QDA	13.83(5.88)	12.54(5.79)	12.7(5.51)
RaSE <sub>1</sub> -KNN	7.51(3.80)	6.16(3.48)	5.9(3.12)
RaSE <sub>2</sub> -LDA	27.49(10.13)	23.39(8.51)	23.39(9.05)
RaSE <sub>2</sub> -QDA	13.15(5.00)	11.9(5.38)	12.15(5.22)
RaSE <sub>2</sub> -KNN	<b>7.06</b> (3.62)	<b>5.89</b> (3.32)	<b>5.74</b> (3.02)
LDA	47.51(2.66)	33.27(7.65)	27.89(8.97)
QDA	NA	NA	36.7(4.83)
KNN	24.49(6.64)	21.04(6.60)	19.07(6.50)
RF	23.64(8.05)	17.39(6.19)	14.84(5.73)

From Table 3, we can see that Super RaSE and its iterative versions still have a very competitive performance compared with other methods. In particular, both SRaSE<sub>1</sub> and SRaSE<sub>2</sub> reach test errors that are within one standard deviation of the best performing method RaSE<sub>2</sub>-KNN, which uses the knowledge of the data generation process. As a method-free algorithm, Super RaSE algorithms and their iterative versions are very appealing, having similar performance as the best-performing ones without the need to specify which base classifier to use.

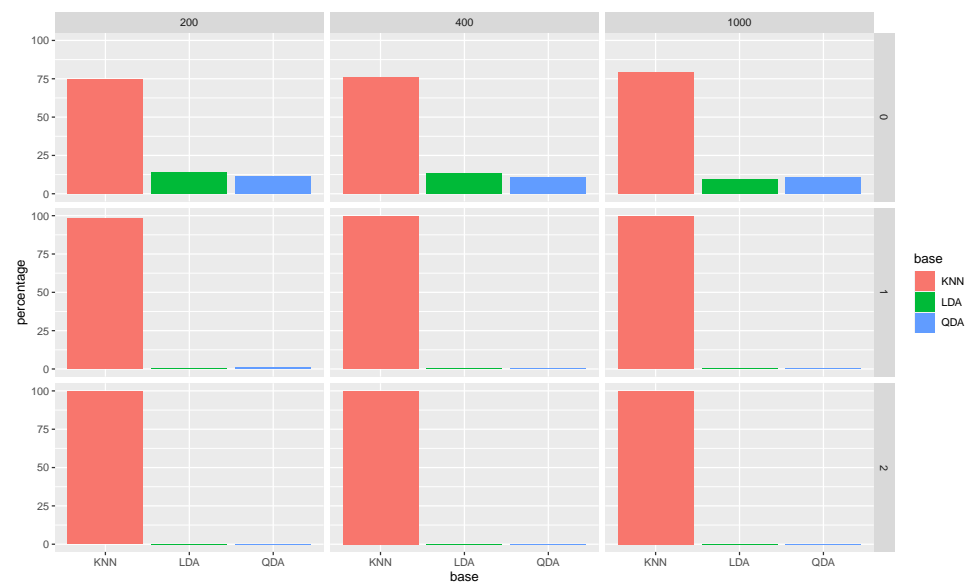
Similar to models 1 and 2, we again present the average selected proportion of each base method in Figure 5. Figure 6 visualizes the selected proportions of features among all the  $B_1$  classifiers that use KNN as the base classifier. In addition to the bar chart for the average selected proportion of each important feature, Figure 6 also includes a boxplot of the selected proportion of all the noisy features.

In Figure 5, we have a similar story as Figures 1 and 3. We can see that the average selection percentage of KNN is almost 100 percent for both SRaSE<sub>1</sub> and SRaSE<sub>2</sub>, showing that one step iteration is enough to almost always find the best classifier for this particular model.

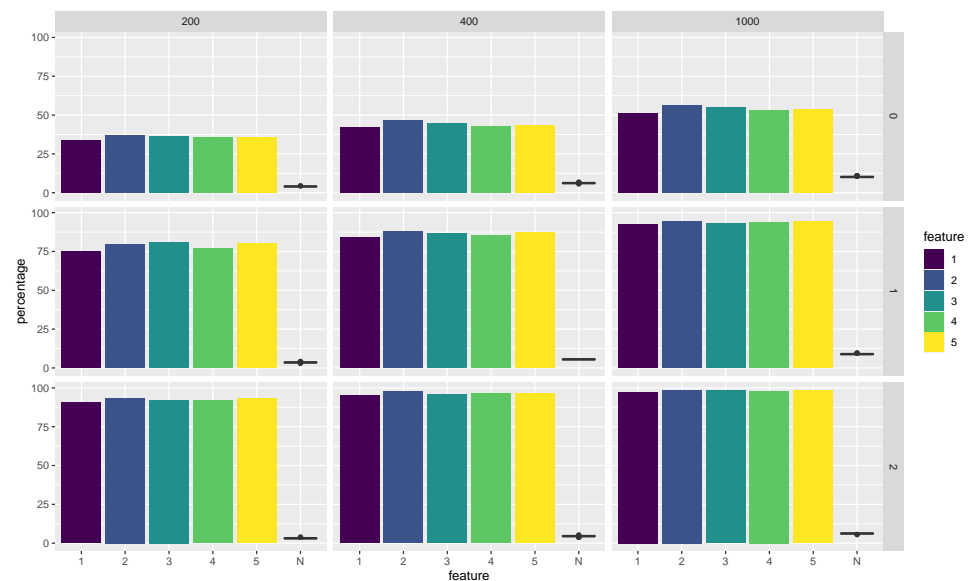
In Figure 6, we can see that without iteration, Super RaSE on average only selects the five important features with around 50 percent out of the 200 repetitions. With the help of iterations, the percentages of all five important features increase substantially, to almost 100 percent when  $n = 1000$ . This experiment shows the merit of iterative Super RaSE in terms of capturing the important features.

#### 4. Real Data Analysis

In this section, we evaluate the proposed Super RaSE algorithm and its iterative version via two real data examples. Here, we compare SRaSE and SRaSE<sub>1</sub> with the original RaSE with the base classifier is LDA, QDA, KNN along with their one-step iterated versions. We also add the classic LDA, QDA, KNN, and Random Forest (RF)



**Figure 5.** The average selected proportion for each base method for different sample sizes (corresponding to each column) and iteration number (corresponding to each row) in Model 3 (KNN).



**Figure 6.** The average selected proportion for each feature for different sample sizes (corresponding to each column) and iteration number (corresponding to each row) in Model 3 (KNN).

to the horserace. Same as in the simulation studies, we use the default values for all parameters in the Super RaSE algorithm and its iterative version ( $B_1 = 200$ ,  $B_2 = 500$ ).

#### 4.1. Mice Protein Expression

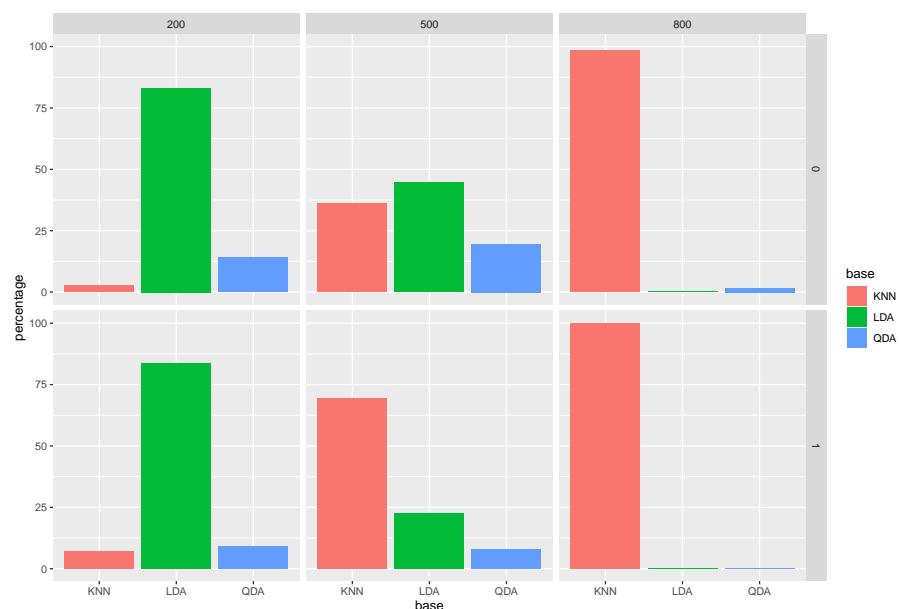
The first example we consider is the mice protein expression data set (Higuera et al. 2015). The dataset is publicly available via <https://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression>. There are 1080 mice in total, with 570 healthy (class 0) and 510 with Down's syndrome (class 1). 77 features are measured, representing the expression of 77 different proteins. Here, we vary the size of the training sample in  $\{200, 500, 800\}$ , and the remaining mice are set as the test data.

The average of test misclassification rates and standard deviations of 200 replicates are calculated with results reported in Table 4. When  $n = 200$ , our newly proposed algorithm SRaSE<sub>1</sub> achieves the lowest error among all approaches. When  $n = 1000$ , both SRaSE and SRaSE<sub>1</sub> are within one standard deviation from the best performing method: RaSE<sub>1</sub>-KNN. This shows the Super RaSE and its iterative versions are very accurate in terms of prediction performance.

Table 4: Summary of test classification error rates for each classifier under various sample sizes over 200 repetitions for the mice protein expression data set. The results are presented as mean values with the standard deviation values in parentheses

	$n = 200$	$n = 500$	$n = 800$
SRaSE	7.08(1.92)	3.33(2.54)	0.67(0.65)
SRaSE <sub>1</sub>	<b>6.76</b> (1.95)	2.56(2.39)	0.65(0.60)
RaSE-LDA	7.41(1.14)	5.7(0.93)	4.65(1.24)
RaSE-QDA	9.14(2.58)	4.81(1.17)	3.44(1.23)
RaSE-KNN	6.8(1.88)	<b>1.55</b> (0.88)	0.62(0.55)
RaSE <sub>1</sub> -LDA	7.24(1.10)	5.53(1.02)	4.49(1.23)
RaSE <sub>1</sub> -QDA	9.38(2.21)	5.16(1.16)	3.4(1.16)
RaSE <sub>1</sub> -KNN	7.43(2.00)	1.7(0.87)	<b>0.6</b> (0.56)
LDA	7.07(1.37)	3.88(0.85)	3.13(1.08)
KNN	20.53(2.47)	7.75(1.44)	2.8(1.21)
RF	8.32(1.71)	2.62(0.94)	1.04(0.73)

Next, we show the average selected proportion for each base method for different sample sizes and iteration numbers in Figure 7.



**Figure 7.** The average selected proportion for each base method for different sample sizes (corresponding to each column) and iteration number (corresponding to each row) for the Mice Protein Expression data.

From this figure, we observe a very interesting phenomenon. That is when  $n = 200$ , the selected proportion of LDA is the highest (over 75%). This choice is very reasonable since comparing the RaSE classifier with different base classifiers, RaSE<sub>1</sub>-LDA has the best performance among RaSE<sub>1</sub> combined with other base classifiers. Now, looking at the case when  $n = 800$ , our Super RaSE and SRaSE<sub>1</sub> both select KNN over 95% percent

of the time. Again, let's look at the RaSE classifier with a fixed base classifier, it is easy to observe from Table 4 that  $\text{RaSE}_1\text{-KNN}$  has a much better performance than both  $\text{RaSE}_1\text{-LDA}$  and  $\text{RaSE}_1\text{-QDA}$ . This shows that the Super RaSE algorithm is very adaptive to the specific scenario in the sense in each of the  $B_1$  weak learners, it automatically selects the base classifier among the randomly selected  $B_2$  classifiers coupled with the random subspaces.

#### 4.2. Hand-written Digits Recognition

The second dataset we consider is the hand-written digits recognition data set that consists of features of hand-written numerals (0-9) extracted from a collection of Dutch utility maps (Dua and Graff 2019). The data set is publicly available at <https://archive.ics.uci.edu/ml/datasets/Multiple+Features>. Out of the 10 digits, we focus the observations corresponding to numbers 7 (class 0) and 9 (class 1). After this preprocessing, we have 400 numbers with 200 in class 0 and another 200 in class 1. We again vary the training sample size, this time in  $\{50, 100, 200\}$ . The rest of the observations is used as the test data. The average of test misclassification rates along with their standard deviations are reported in Table 5.

Table 5: Summary of test classification error rates for each classifier under various sample sizes over 200 repetitions for the hand-written digits recognition data set. The results are presented as mean values with the standard deviation values in parentheses.

	$n = 50$	$n = 100$	$n = 200$
SRaSE	2(1.06)	1.2(0.65)	0.78(0.48)
SRaSE <sub>1</sub>	1.78(1.03)	1.04(0.57)	0.62(0.37)
RaSE-LDA	1.56(0.85)	1.13(0.59)	0.8(0.54)
RaSE-QDA	2.5(1.47)	1.89(0.91)	1.47(0.96)
RaSE-KNN	1.86(0.96)	1.12(0.66)	0.75(0.45)
RaSE <sub>1</sub> -LDA	<b>1.06(0.63)</b>	0.7(0.35)	<b>0.53(0.40)</b>
RaSE <sub>1</sub> -QDA	2.18(1.66)	1.18(0.71)	0.85(0.61)
RaSE <sub>1</sub> -KNN	1.72(0.95)	1.02(0.62)	0.6(0.44)
LDA	NA	1.82(0.96)	1.01(0.56)
QDA	NA	NA	3.25(2.32)
KNN	1.42(1.32)	<b>0.67(0.41)</b>	0.6(0.47)
RF	2.34(1.24)	1.63(0.73)	1.37(0.74)

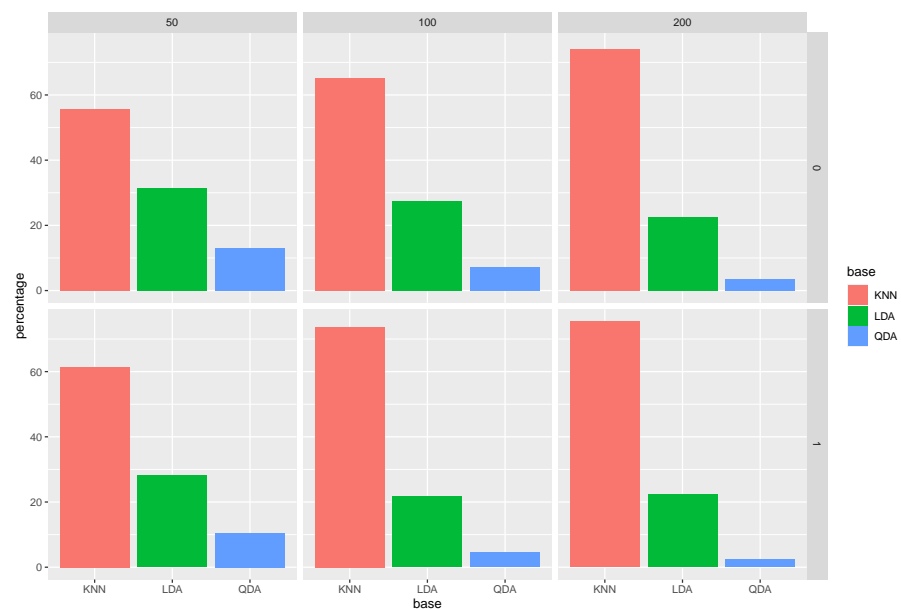
From Table 5, we can see that the Super RaSE and its iterative versions still have competitive performances, especially when  $n = 100$  and  $n = 200$ . The SRaSE<sub>1</sub> has a very similar performance as the best performing method when  $n = 100$  and  $n = 200$ . Having a careful look, it is interesting to observe that SRaSE<sub>1</sub> closely mimics the performance of RaSE<sub>1</sub>-KNN, leading us to wonder whether KNN is the most selected base method among the  $B_1$  classifiers. We confirm this is the case by presenting the average selected proportion of each base classifier for SRaSE and SRaSE<sub>1</sub> in Figure 8.

From Figure 8, we can see that KNN is indeed the most selected base classifier across all scenarios, with its selected proportion almost 100% when  $n = 200$ .

## 5. Discussion

In this work, motivated by the random subspace ensemble classification (RaSE), we propose a new ensemble classification framework, namely Super RaSE, which is a completely model-free approach. By randomly generating the pair of base classifier and subspace, it will automatically select a collection of good base classifier and subspace pairs. Besides the superb prediction performance on the test data, the Super RaSE algorithms also provide important insights as to which base classifier may be





**Figure 8.** The average selected proportion for each base method for different sample sizes (corresponding to each column) and iteration number (corresponding to each row) for the Hand-written Digits Recognition data.

more appropriate for the data and which features may be more important in terms of prediction.

There are many possible future research directions. First, this paper only considers the binary classification problem while there may be more than two classes in applications. How to extend the Super RaSE algorithm to the multiclass situation is an important topic. Second, in addition to the classification problem, it is also worthwhile to study the corresponding algorithm under a regression setting. Third, using the selected proportion of features, it is possible to develop a variable selection or variable screening algorithm.

## References

- Blaser, Rico and Piotr Fryzlewicz. 2016. Random rotation ensembles. *The Journal of Machine Learning Research* 17(1), 126–151.
- Breiman, Leo. 1996. Bagging predictors. *Machine learning* 24(2), 123–140.
- Breiman, Leo. 2001. Random forests. *Machine learning* 45(1), 5–32.
- Cannings, Timothy I and Richard J Samworth. 2017. Random-projection ensemble classification. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79(4), 959–1035.
- Dietterich, Thomas G. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pp. 1–15. Springer.
- Dua, D and C Graff. 2019. Uci machine learning repository. school of information and computer science, university of california, irvine, ca.
- Durrant, Robert J and Ata Kabán. 2015. Random projections as regularizers: learning a linear discriminant from fewer observations than dimensions. *Machine Learning* 99(2), 257–286.
- Fan, Yingying, Yinfei Kong, Daoji Li, Zemin Zheng, et al. 2015. Innovated interaction screening for high-dimensional nonlinear classification. *The Annals of Statistics* 43(3), 1243–1272.
- Feng, Yang, Qingfeng Liu, Qingsong Yao, and Guoqing Zhao. 2021. Model averaging for nonlinear regression models. *Journal of Business & Economic Statistics*, 1–14.
- Higuera, Clara, Katherine J Gardiner, and Krzysztof J Cios. 2015. Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome. *PloS one* 10(6).
- Ho, Tin Kam. 1998. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(8), 832–844.
- Mai, Qing, Hui Zou, and Ming Yuan. 2012. A direct approach to sparse discriminant analysis in ultra-high dimensions. *Biometrika* 99(1), 29–42.

- 
- Raftery, Adrian E, David Madigan, and Jennifer A Hoeting. 1997. Bayesian model averaging for linear regression models. *Journal of the American Statistical Association* 92(437), 179–191.
- Rokach, Lior. 2010. Ensemble-based classifiers. *Artificial Intelligence Review* 33(1-2), 1–39.
- Tian, Ye and Yang Feng. 2021a. Rase: A variable screening framework via random subspace ensembles. *Journal of the American Statistical Association* (just-accepted), 1–30.
- Tian, Ye and Yang Feng. 2021b. Rase: Random subspace ensemble classification. *J. Mach. Learn. Res.* 22, 45–1.
- Van der Laan, Mark J, Eric C Polley, and Alan E Hubbard. 2007. Super learner. *Statistical applications in genetics and molecular biology* 6(1).