A demonstration of the RaSEn package

Ye Tian and Yang Feng

2020-10-20

We provide a detailed demo of the usage for the `RaSEn` package.

- Introduction
- Installation
- How to Fit a RaSE Classifier for Prediction
- How to Use RaSE for Feature Ranking

Introduction

Suppose we have training data $\{\mathbf{x}_i, y_i\}_{i=1}^n \in \{\mathbb{R}^p, \{0,1\}\}$, where each $\mathbf{x}_i$ is a $1 \times p$ vector.

Based on training data, RaSE algorithm aims to generate B_1 weak learners $\{C_n^{S_{j*}}\}_{j=1}^{B_1}$, each of which is constructed in a feature subspace $S_{j*} \subseteq \{1, ..., p\}$ instead using all p features. To obtain each weak learner, B_2 candidates $\{C_n^{S_{jk}}\}_{k=1}^{B_2}$ are trained based in subspaces $\{S_{jk}\}_{k=1}^{B_2}$, respectively. To choose the optimal one among these B_2 candidates, some criteria need to be applied, including minimizing ratio information criterion (RIC, Tian and Feng (2020)), minimizing extended Bayes information criterion (eBIC, Chen and Chen (2008), Chen and Chen (2012)), minimizing the training error, minimizing the validation error (if validation data is available), minimizing the cross-validation error, minimizing leave-one-out error etc. And the type of weak learner can be quite flexible.

To better adapt RaSE into the sparse setting, we can update the distribution of random feature subspaces according to the selected percentage of features in B_1 subspaces in each round. This can be seen as an adaptive strategy to increase the possibility to cover the signals that contribute to our model, which can improve the performance of RaSE classifiers in sparse settings.

The selected percentage of each of p features in B_1 subspaces can be used for feature ranking as well. And we could plot the selected percentage to intuitively rank the importance of each feature in a RaSE model.

Installation

`RaSEn` can be installed from CRAN.

```
install.packages("RaSEn", repos = "http://cran.us.r-project.org")
```

Then we can load the package:

```
library(RaSEn)
```

How to Fit a RaSE Classifier for Prediction

We will show in this section how to fit RaSE classifiers based on different types of base classifiers. First we generate the data from a binary guanssian mixture model (Model 1 in Tian and Feng (2020))

$$\mathbf{x} \sim (1-y)N(\mu^{(0)}, \Sigma) + yN(\mu^{(1)}, \Sigma),$$

where $\mu^{(0)}, \mu^{(1)}$ are both $1 \times p$ vectors, Σ is a $p \times p$ symmetric positive definite matrix. Here y follows a bernoulli distribution:

$$y \sim Bernoulli(\pi_1),$$

where $\pi_1 \in (0, 1)$ and we denote $\pi_0 = 1 - \pi_1$.

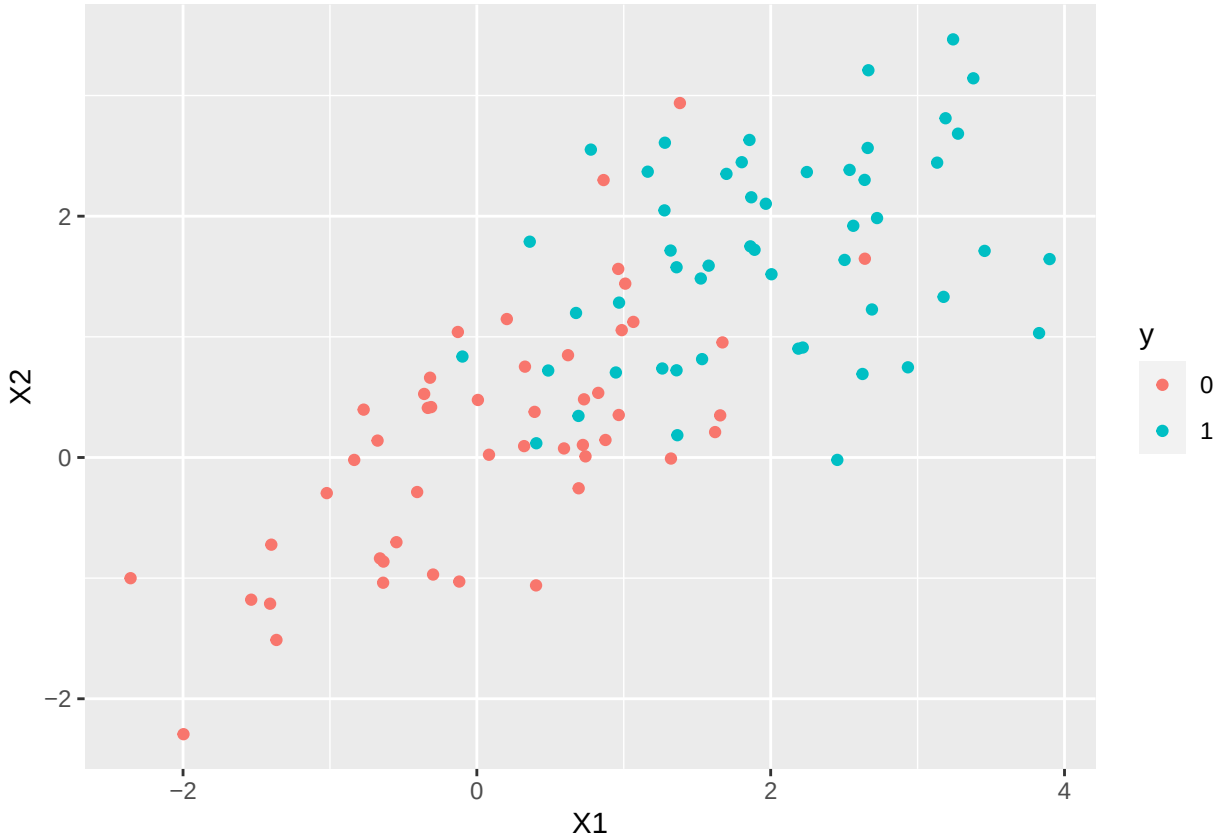
Here we follow from the setting of Mai, Zou, and Yuan (2012), letting $\Sigma = (0.5^{|i-j|})_{p\times p}, \mu^{(0)} = \mathbf{0}_{p\times 1}, \mu^{(1)} = \Sigma^{-1} \times 0.556(3, 1.5, 0, 0, 2, \mathbf{0}_{1\times(p-5)})^T$. Let $n = 100, p = 50$. According to the definition of minimal discriminative set in Ye Tian and Yang Feng (2020), here the minimal discriminative set $S^* = \{1, 2, 5\}$, which contribute to the classification.

Apply function `RaModel` to generate training data and test data of size 100 with dimension 50.

```
set.seed(0, kind = "L'Ecuyer-CMRG")
train.data <- RaModel(1, n = 100, p = 50)
test.data <- RaModel(1, n = 100, p = 50)
xtrain <- train.data$x
ytrain <- train.data$y
xtest <- test.data$x
ytest <- test.data$y
```

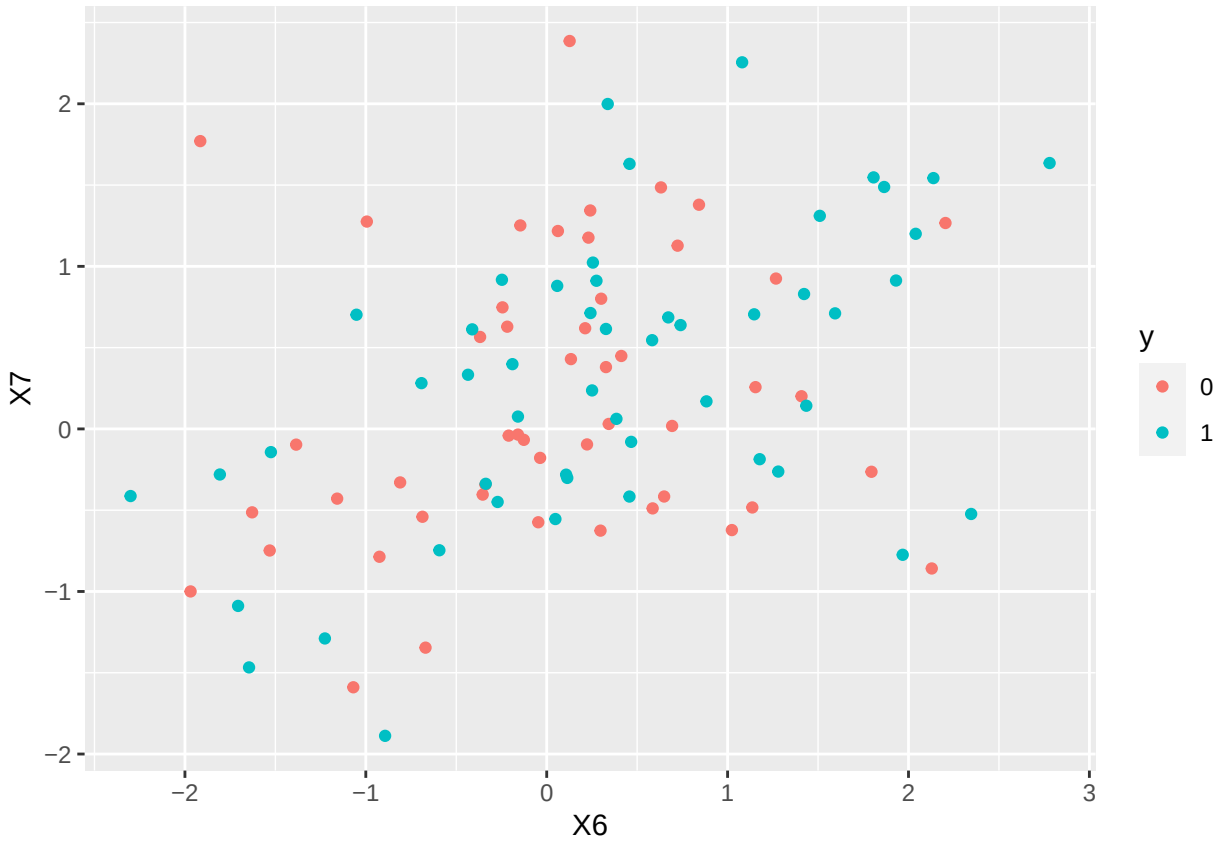
We can visualize the first two dimensions or feature 1 and 2 as belows:

```
library(ggplot2)
ggplot(data = data.frame(xtrain, y = factor(ytrain)), mapping = aes(x = X1,
    y = X2, color = y)) + geom_point()
```



Similarly, we can also visualize the feature 6 and 7:

```
ggplot(data = data.frame(xtrain, y = factor(ytrain)), mapping = aes(x = X6, y = X7,
    color = y)) + geom_point()
```



It's obvious to see that in dimension 1 and 2 the data from two classes are more linearly seperate than in dimension 6 and 7. Then we call `Rase` function to fit the RaSE classifier with LDA, QDA and logistic regression base classifiers with criterion of minimizing RIC and RaSE classifier with knn base classifier with criterion of minimizing leave-one-out error. To use different types of base classifier, we set `base` as "lda", "qda", "knn" and "logistic", repectively. `B1` is set to be 100 to generate 100 weak learners and `B2` is set to be 50 as well to generate 50 subspace candidates for each weak learner. Without using iterations, we set `iteration` as 0. `criterion` is set to be "ric" for RaSE classifier with LDA, QDA and logistic regression while it is "loo" for RaSE classifier with knn base classifier. To speed up the computation, we apply parallel computing with 2 cores by setting `cores = 2`.

```
fit.lda <- Rase(xtrain, ytrain, B1 = 100, B2 = 50, iteration = 0, base = "lda",
    cores = 2, criterion = "ric")
fit.qda <- Rase(xtrain, ytrain, B1 = 100, B2 = 50, iteration = 0, base = "qda",
    cores = 2, criterion = "ric")
fit.knn <- Rase(xtrain, ytrain, B1 = 100, B2 = 50, iteration = 0, base = "knn",
    cores = 2, criterion = "loo")
fit.logistic <- Rase(xtrain, ytrain, B1 = 100, B2 = 50, iteration = 0,
    base = "logistic", cores = 2, criterion = "ric")
```

We can print the summarized results of RaSE model by calling `print` function. For instance, we print the RaSE model with LDA base classifier:

```
print(fit.lda)
```

```
## Marginal probabilities:
## class 0 class 1
##    0.49    0.51
## Type of base classifiers: lda
## Criterion: ric
```

```
## B1: 100
## B2: 50
## Cutoff: 0.6286771
## Selected percentage of each feature appearing in B1 subspaces:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
## 100  34   4  11  34  10  12  11   3   9  12  12  10  16  14  14  23  14  22  19
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##  16   9   8   8   8   8   9  11  13   6   6  10  10   7  11  19   6   8   9  16
##  41  42  43  44  45  46  47  48  49  50
##  13  18   5   6   7   8  10   6  12   3
```

To evaluate the performance of four different models, we calculate the test error on test data:

```
er.lda <- mean(predict(fit.lda, xtest) != ytest)
er.qda <- mean(predict(fit.qda, xtest) != ytest)
er.knn <- mean(predict(fit.knn, xtest) != ytest)
er.logistic <- mean(predict(fit.logistic, xtest) != ytest)
cat("LDA:", er.lda, "QDA:", er.qda, "knn:", er.knn, "logistic:", er.logistic)
```

```
## LDA: 0.03 QDA: 0.07 knn: 0.05 logistic: 0.05
```

And the output of `Rase` function is an object belonging to S3 class "RaSE". It contains:

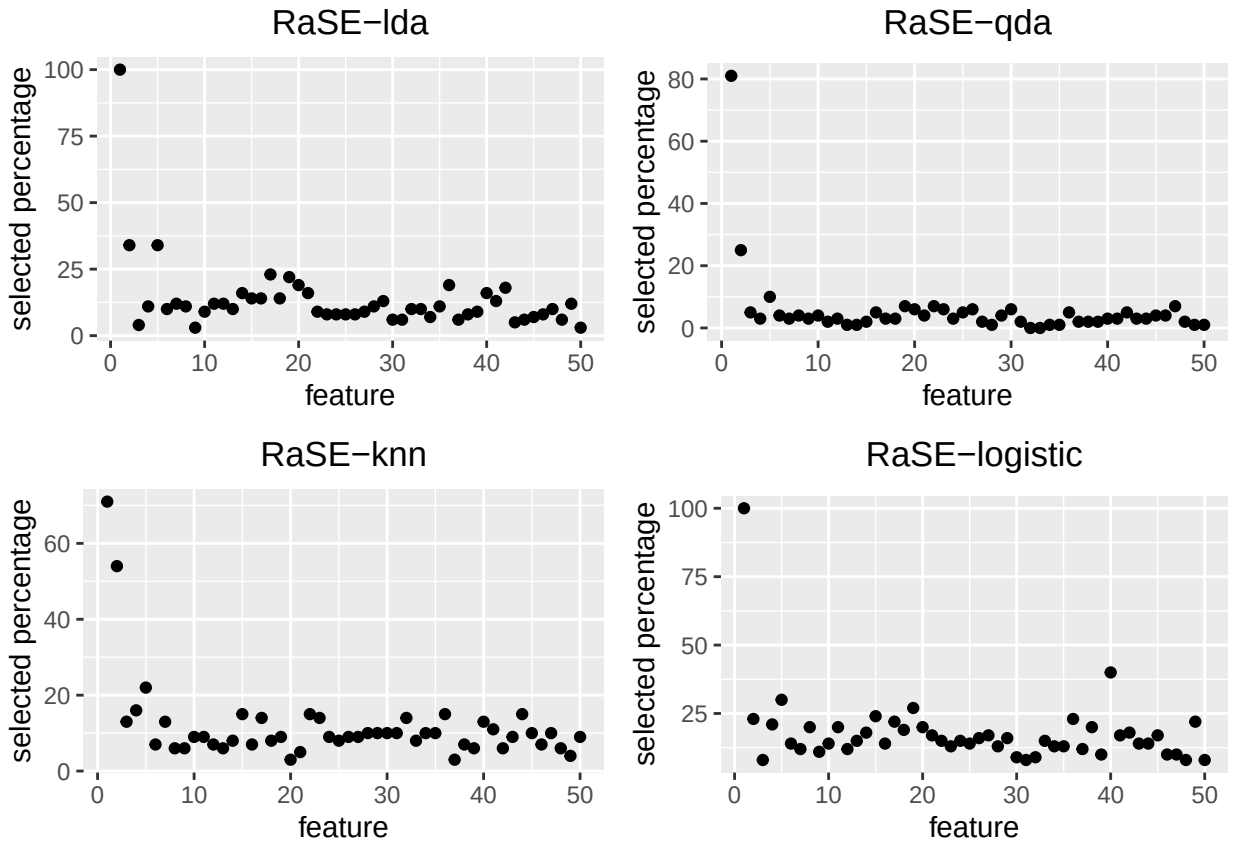
- marginal: the marginal probability for each class.
- fit.list: a list of B1 fitted base classifiers.
- B1: the number of weak learners.
- B2: the number of subspace candidates generated for each weak learner.
- base: the type of base classifier.
- cutoff: the empirically optimal threshold.
- subspace: a list of subspaces correponding to B1 weak learners.
- ranking: the selected percentage of each feature in B1 subspaces.
- C0: the threshold used to adjust the sampling probabilities of features when `iteration` > 0.

How to Use RaSE for Feature Ranking

The selected percentage of features in B_1 subspaces for four RaSE classifiers are contained in the output, which can be used for feature ranking. We can plot them by using `RaPlot` function:

```
library(gridExtra)
plot_lda <- RaPlot(fit.lda)
plot_qda <- RaPlot(fit.qda)
plot_knn <- RaPlot(fit.knn)
plot_logistic <- RaPlot(fit.logistic)

grid.arrange(plot_lda, plot_qda, plot_knn, plot_logistic, ncol = 2)
```



From four figures, it can be noticed that feature 1, 2 and 5 obtain higher selected percentage than the others among all $p = 50$ features when the base classifier is chosen to be LDA, QDA and kNN, implying their importance in classification model. We can set a positive iteration number to increase the selected percentage of three signals among B_1 subspaces as follows, which may improve the performance.

```
library(gridExtra)
fit.lda <- Rase(xtrain, ytrain, B1 = 100, B2 = 50, iteration = 1, base = "lda",
    cores = 2, criterion = "ric")
fit.qda <- Rase(xtrain, ytrain, B1 = 100, B2 = 50, iteration = 1, base = "qda",
    cores = 2, criterion = "ric")
fit.knn <- Rase(xtrain, ytrain, B1 = 100, B2 = 50, iteration = 1, base = "knn",
    cores = 2, criterion = "loo")
fit.logistic <- Rase(xtrain, ytrain, B1 = 100, B2 = 50, iteration = 1,
    base = "logistic", cores = 2, criterion = "ric")
```

Reference

Chen, Jiahua, and Zehua Chen. 2008. "Extended Bayesian Information Criteria for Model Selection with Large Model Spaces." *Biometrika* 95 (3): 759–71.

———. 2012. "Extended Bic for Small-N-Large-P Sparse Glm." *Statistica Sinica*, 555–74.

Mai, Qing, Hui Zou, and Ming Yuan. 2012. "A Direct Approach to Sparse Discriminant Analysis in Ultra-High Dimensions." *Biometrika* 99 (1): 29–42.

Tian, Ye, and Yang Feng. 2020. "RaSE: Random Subspace Ensemble Classification." *arXiv Preprint arXiv:2006.08855*.