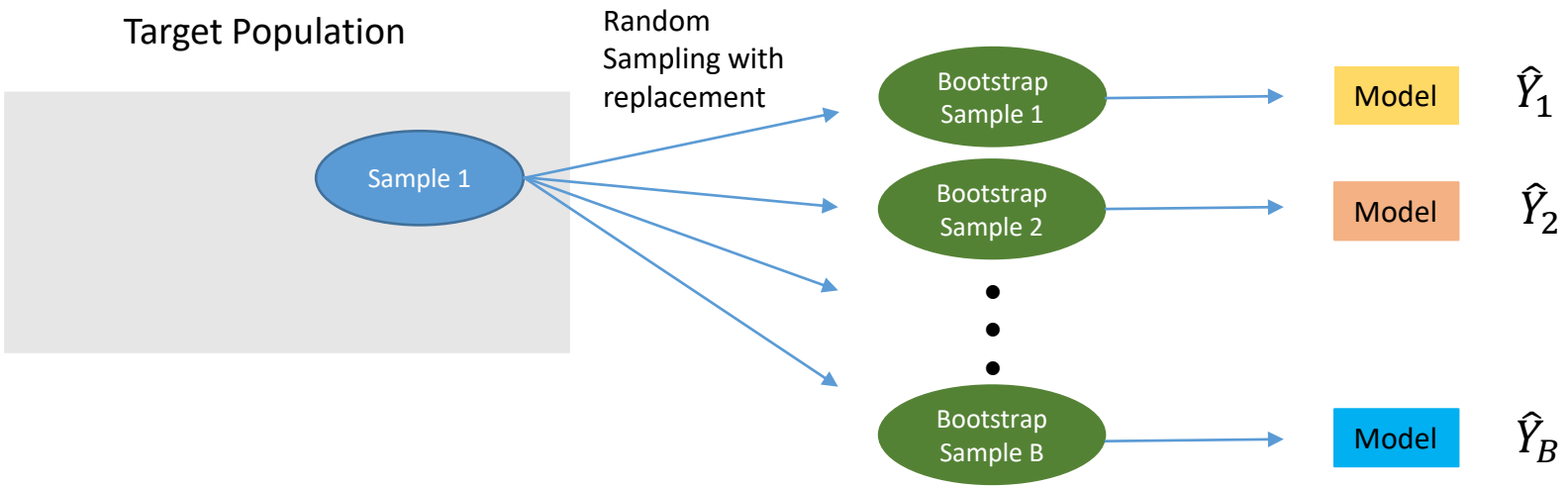
Answers in Random Forests

BC2407 Seminar 8

Bootstrap Aggregating (Bagging) Errors



- To get trainset error in Bagging, just take the average of the errors within the B Bootstrap samples.
- Q: How to get testset error in Bagging?
  - Breiman (1996a) did 90-10 train-test split from original sample 1 repeatedly (100 times).
  - A more efficient and natural testset procedure was subsequently devised in Breiman (2001) – OOB Error.

Sources: Chew C.H. (2020) AI, Analytics and Data Science, Vol. 2: Advanced Predictive Techniques, Chap. 8 & Breiman (1996a)

Activity 1: Prove that $E(Z^2) \geq [E(Z)]^2$ and explain when Bagging will perform well. [10 mins]

- The proof of the benefit of Bagging (and hence Random Forest too) in reducing MSE, given in Breiman (1996a) relies on the inequality proof: $E(Z^2) \geq [E(Z)]^2$.
- Hint: Proof of the inequality by treating Z as a random variable is a standard result in Statistics.
- Z is the random variable and interpreted as the model predicted value of Y, when trainset changes.

$$Let\ E(Z) \equiv\ \mu\ and\ E((Z-\mu)^2) \equiv\ \sigma^2$$

$$\begin{aligned}
&E(Z^2)\\
&= E((Z-\mu+\mu)^2)\\
&= E((Z-\mu)^2 + 2(Z-\mu)(\mu) + \mu^2)\\
&= E((Z-\mu)2) + 2\mu E(Z-\mu) + \mu^2\\
&= \sigma^2 + 0 + [E(Z)]^2\\
&\geq [E(Z)]^2
\end{aligned}$$

The bigger the variance of Z (i.e. σ^2), the greater the inequality.

Thus, this result shows the power and limitation of Bagging.

If the variance is low, Bagging will not improve single-model generalisation error by much.

If the variance is high, Bagging will improve single-model generalisation error by a lot.

Results of Random Forest on Heart data

```
Call:
 randomForest(formula = AHD ~ ., data = heart.df, importance = T,      na.action = na.omit)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 18.18%
Confusion matrix:
     No Yes class.error
No  136  24   0.1500000
Yes  30 107   0.2189781
```

- B = 500, RSF size = 3; OOB overall error = (24 + 30)/297 = 18.18%.
- Q1: How are the confusion matrix results determined?
  - Ans: From OOB data and majority rule.
- Q2: Why did the confusion matrix contain only 297 cases when the dataset has 303 cases?
  - Ans: 6 cases has missing values and were omitted in na.action= na.omit
- Q3: Verify if confusion matrix rows or columns represent actuals? [10 mins]
  - Ans: Check the distribution of Y in the data and compare against C.M.
- Q4: What is the meaning of the two class errors (0.15, 0.2189781)?
  - Ans: False Positive Rate and False Negative Rate.

Verify if RF confusion matrix rows represent actual Y or model predicted Y

```
> summary(na.omit(heart.df)$AHD)
 No Yes
160 137
```

```
> m.RF.1$confusion
     No Yes class.error
No  136  24   0.1500000
Yes  30 107   0.2189781
```

- Summary() shows the distribution of AHD in the missing values removed data: 160 actual No, 137 actual Yes. Total 297 cases.
- The rows in the confusion matrix total 160 and 137 respectively. Hence, rows represent actual Y values and columns represent model predicted Y values.
- False Positive Rate = 24/(136 + 24) = 0.15
- False Negative Rate = 30/(30 + 107) = 0.2189781

View RF vote for each case in the dataset via m.RF.1$votes
View RF prediction for each case via m.RF.1$predicted

```
> m.RF.1$votes
             No         Yes
1   0.56497175 0.435028249
2   0.10270270 0.897297297
3   0.06217617 0.937823834
4   0.55172414 0.448275862
5   0.98275862 0.017241379
6   0.96111111 0.038888889
7   0.28061224 0.719387755
8   0.61538462 0.384615385
9   0.16279070 0.837209302
10  0.30612245 0.693877551
11  0.65625000 0.343750000
12  0.80113636 0.198863636
13  0.32105263 0.678947368
14  0.75661376 0.243386243
15  0.82291667 0.177083333
16  0.84390244 0.156097561
17  0.72093023 0.279069767
```

Q: Consider case 1. Does this mean 56% of the 500 trees voted AHD = No and 44% of the 500 trees voted AHD = Yes?

Ans: No. Not 500 trees. Only in those trees (approx. 1/3 of 500) for which case 1 is OOB.

```
> m.RF.1$predicted
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23
 No Yes Yes  No  No  No Yes  No Yes Yes  No  No Yes  No  No  No  No  No  No  No  No  No  No
 24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46
```

votes (classification only) a matrix with one row for each input data point and one column for each class, giving the fraction or number of (OOB) 'votes' from the random forest.

Caution: m.RF.1$err.rate is not the error rate at each tree

```
> err.rate <- m.RF.1$err.rate
> View(err.rate)
```

	OOB	No	Yes
1	0.2654867	0.3015873	0.2200000
2	0.2696629	0.2755102	0.2625000
3	0.2590909	0.2333333	0.2900000
4	0.2701613	0.2388060	0.3070175
5	0.2613636	0.2291667	0.3000000
6	0.2637363	0.2482759	0.2812500
7	0.2588652	0.2432432	0.2761194
8	0.2465278	0.2287582	0.2666667

Q: Consider row 4. What is the meaning of OOB = 0.27?

Ans: OOB error using the first 4 trees is 27%

`err.rate` (classification only) vector error rates of the prediction on the input data, the i-th element being the (OOB) error rate for all trees up to the i-th.

Learning Activity: Random Forest on Heart data

Est. Duration: 40 mins

Instructor solution in RF Heart2.R

- Run RF Heart1.R.
- Execute Random Forest with different settings on Heart.csv to predict AHD and save the respective OOB error in a table. How does the default settings for B and RSF size fare in terms of OOB error?
    1. B = 25, RSF size = 1
    2. B = 25, RSF size = int(sqrt(m)) = 3
    3. B = 25, RSF size = m = 13
    4. B = 100, RSF size = 1
    5. B = 100, RSF size = int(sqrt(m)) = 3
    6. B = 100, RSF size = m = 13
    7. B = 500, RSF size = 1
    8. B = 500, RSF size = int(sqrt(m)) = 3
    9. B = 500, RSF size = m = 13
- Based on the Random Forest model with the default settings, which variables are important in predicting AHD? Explain how "variable importance" is determined.

Key Findings in Learning Activity

set.seed(1)

	B	RSF	OOB.error
7	500	1	0.1649832
8	500	3	0.1683502
4	100	1	0.1784512
5	100	3	0.1885522
2	25	3	0.1919192
6	100	13	0.1952862
9	500	13	0.2053872
1	25	1	0.2188552
3	25	13	0.2255892

set.seed(2020)

	B	RSF	OOB.error
5	100	3	0.1683502
7	500	1	0.1717172
8	500	3	0.1717172
4	100	1	0.1818182
2	25	3	0.1851852
9	500	13	0.1851852
3	25	13	0.1885522
6	100	13	0.1919192
1	25	1	0.2154882

- Trying different seeds, default settings of B and RSF size works quite well.

Learning Activity: Random Forest vs MARS on Resale Flat Price

Est. Duration: 30 mins

- Dataset: resale-flat-prices-2019.csv [from MARS seminar]
- Run flatprice-RF.R
  - set.seed(2) and do 70-30 train-test split.
  - Includes data prep.
- Construct the following models and get respective testset RMSE:
  1. MARS (degree 2) [already done in Rscript]
  2. Random Forest (default settings for B & mtry)
- Which variables are important?

- *Instructor solution in flatprice-RF solution.R*