Random Forest

Random Forest is a small extension of Bagging, in which the bootstrap trees are decorrelated

The idea is, we draw a bootstrap sample and start to build a tree. At each split, we randomly select m of the possible p features as candidates for the split; A new sample of size m of the features is taken at each split. Usually, we use about $m = \sqrt{p}$

In other words, at each split, we aren't even allowed to consider the majority of possible features!

Suppose there is 1 really strong feature and many mediocre ones.

- Then each tree will have this one feature in it,
- Therefore, each tree will look very similar (i.e. highly correlated).
- Averaging highly correlated things leads to much less variance reduction than if they were uncorrelated.

If we don't allow some trees/splits to use this important feature, each of the trees will be much less similar and hence much less correlated.

Bagging is Random Forest when $m = p$, that is, when we can consider all the features at each split.

An average of B i.i.d random variables has variance

$$\frac{\sigma^2}{B}$$

An average of B random variables has variance

$$\rho\sigma^2 + \frac{(1-\rho)\sigma^2}{B}$$

for correlation ρ

As $B \to \infty$, the second term goes to zero, but the first term remains. Hence, correlation of the trees limits the benefit of averaging

Sensitivity and specificity

Sensitivity:	The proportion of times we label recession, given that recession is the correct answer.
Specificity:	The proportion of times we label no recession, given that no recession is the correct answer.

We can think of this in terms of hypothesis testing. If

$$H_0: \text{ no recession},$$

then

Sensitivity:	P(reject $H_0 \| H_0$ is false), [1 - P(Type II error)]
Specificity:	P(accept $H_0 \| H_0$ is true), [1 - P(Type I error)]

Confusion matrix

We can report our results in a matrix:

		Truth	
		Recession	No Recession
Our	Recession	(A)	(B)
Predictions	No Recession	(C)	(D)

The total number of each combination is recorded in the table.

The overall miss-classification rate is

$$\frac{(B) + (C)}{(A) + (B) + (C) + (D)} = \frac{(B) + (C)}{\text{total observations}}$$

Sensitivity is (A)/[(A) + (C)], Specificity is (D)/[(B) + (D)]

Tree results: Confusion matrices

			Truth		
			Growth	Recession	Mis-Class
Our Preds	Null	Growth	111	26	
		Recession	0	0	18.9%
	Tree	Growth	99	3	
		Recession	12	23	10.9%
	Random Forest	Growth	102	5	
		Recession	9	21	10.2%
	Bagging	Growth	104	3	
		Recession	7	23	7.3%

Tree results: Sensitivity & specificity

	Sensitivity	Specificity
Null	0.000	1.000
Tree	0.884	0.891
Random Forest	0.807	0.918
Bagging	0.884	0.936

Out-of-bag error estimation for bagging

		Truth		
		Growth	Recession	Miss-Class
OOB Bagging	Growth	400	10	
	Recession	21	46	6.5%
Test Bagging	Growth	104	3	
	Recession	7	23	7.3%

Random Forest in R

require(randomForest)
out.rf = randomForest(X,Y,importance=T,mtry=p) class.rf = predict(out.rf,X_0)

Notes:

- The importance statement tells it to produce the variable importance measures
- the mtry = p tells randomForest to consider all the covariates at each split
  This particular choice corresponds to bagging
- randomForest also supports formulae out.rf = randomForest(Y .,data=X) However, it can take much longer to run

```
> out.rf

Call:
randomForest(formula = Y~.,data = X, import = T, mtry = p)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 56

        OOB estimate of  error rate: 7.33%
Confusion matrix:
    0  1 class.error
0 508 13  0.02495202
1  32 61  0.34408602


#Permutation variable importance
> head(importance(out.rf,type=1))
           MeanDecreaseAccuracy
Alabama               3.7277511
Alaska                1.7941463
Arizona               2.9659623
Arkansas              0.8341577
California            7.2973572
#Mean decrease variable importance
> head(importance(out.rf,type=2))
           MeanDecreaseGini
Alabama           0.4551073
```

```
Alaska          1.6440170
Arizona         0.7025527
Arkansas        0.3503138
California      1.4616203

#variable importance plot:
varImpPlot(out.rf,type=2)
```

Additional random forest topics

Claim: Random forest cannot overfit.

This is and isn't true. Write

$$\hat{f}_{rf}^{B} = \frac{1}{B}\sum_{b=1}^{B} T(x;\Theta_b)$$

where Θ_b characterizes the b^{th} tree

That is, the split variables, cutpoints of each node, terminal node values.

Increasing B does not cause Random forest to overfit, rather removes the Monte-Carlo-like approximation error

$$\hat{f}_{rf}(x) =_{\Theta} T(x,\Theta) = \lim_{B\to\infty} \hat{f}_{rf}^{B}$$

However, this limit can overfit the data, the average of fully grown trees can result in too complex of a model

Note that Segal (2004) shows that a small benefit can be derived by stopping each tree short, but thus induce another tuning parameter