Chapter 13

Random Forests DRAFT

A random forest is just a collection (or **ensemble**) of decision trees whose "votes" are uncorrelated. The trees vote to produce a final prediction.

Two details are important to the construction of random forests:

1. Each tree is built using a subset of training examples sampled with replacement from the original training set. This is called **bagging** (bootstrap aggregating). Typically around 2/3 of training examples are used per tree. Note that bagging is a general-purpose procedure that can be used for other models besides random forests.

2. For each split, the tree considers not all m predictor variables, but only a randomly-chosen subset, usually of size approximately $\sqrt{m}$ (for classification problems) or $m/3$ (for regression problems). This keeps you from building the same tree over and over again and ensures that the votes from different trees are uncorrelated.

Here are two bagged samples of size 6 from the dataset in Table **??**.

ID	friends (X_1)	money (X_2)	free time (X_3)	happy (Y)
5	1	0	0	0
4	0	0	0	0
2	1	1	1	0
10	1	0	0	1
8	1	0	1	1
10	1	0	0	1

ID	friends (X_1)	money (X_2)	free time (X_3)	happy (Y)
5	1	0	0	0
6	0	0	0	0
2	1	1	1	0
5	1	0	0	0
9	0	0	1	1
7	1	2	1	1

Question 3.11: Use a random forest to fit the data from the low birthweight example used in the logistic regression model, above. Use the following commands exactly as shown to ensure it all runs smoothly and you can view the output:

```
library(randomForest)
d <- read.delim("../data/logistic-lowbwt-data.tsv")
d$RACE <- as.factor(d$RACE)  # <- ensure RACE coded as
    factor
d$LOW <- as.factor(d$LOW)    # <- ensure LOW coded as
    factor
r <- randomForest(LOW ~ AGE + LWT + RACE + SMOKE + PTL
    + HT + UI + FTV, data = d, ntree = 100, do.trace =
    TRUE)
plot(r)
```

The random forest will report a number called the **out-of-bag (OOB)** error as it runs. To calculate OOB error, the trees are allowed to vote on the points that were *not* used in their construction. This provides an ongoing estimate of the generalization error of the algorithm, so you can see if adding more trees is likely to help.

What is the (approximate) overall OOB error? What is it for the positive outcome class only? The negative outcome class only?