Random forests can overfit

Carl Ehrett

April 30, 2019

First, load the `randomForest` library.

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.5.3
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

Generate artificial data

Here we generate data with `p` predictors and `n` observations, for some large `p < n`. Only two of the predictors will be in the true model; the other `p-2` predictors are mere noise.

We also set aside some large `n.train < n` samples to be the training set. The remainder of the observations will be the test set.

```
# Number of samples and number of predictors
n<- 1e4
p<- 1e2
n.train <- 1e2

# Generate the data, first the predictors:
X <- rnorm(n*p)
X <- matrix(X,nrow = n)

# Now the response:
Y <- (X[,1]+X[,2])>0

# Put it all in data frames, separate training and test for convenience
dat <- data.frame(X=X, Y=as.factor(Y))
dat.train <- dat[1:n.train,]
dat.test <- dat[-(1:n.train),]

# Also for convenience, get separate training and test response vectors
Y.test <- dat.test[,p+1]
Y.train <- dat.train[,p+1]
```

Train the random forest model

Now we use the artificial data to fit a random forest model, using default settings.

```
rfmod <- randomForest(Y ~ . , data = dat.train)
```

Get training set error and test set error

Now we get the training set error, the test set error, and the OOB error. Notice that the training set error is far lower than the test set error, indicating that our random forest model did indeed overfit. However, notice too that this overfitting is easily detected simply by checking the OOB error. That is, even though the model massively overfit the training data, the OOB error (like the test error) is far above the training error. This is a big red flag that overfitting has occurred. So while random forests can overfit, it is in general easy to use OOB error to detect that overfitting has occurred, without relying on a test set.

```
# Get predictions of training set response and test set response
preds.train <- predict(rfmod,dat.train[,-(p+1)])
preds.test <- predict(rfmod,dat.test[,-(p+1)])

# Get training set error, test set error, and OOB error
training.error <- 1-sum(Y.train==preds.train)/length(Y.train)
test.error <- 1-sum(Y.test==preds.test)/length(Y.test)
oob.error <- rfmod$err.rate[500,1]

# Display errors
errors <- c(training.error,test.error,oob.error)
names(errors) <- c("Training error", "Test error", "OOB error")
knitr::kable(t(errors), digits = 4,caption="Error estimates")
```

Table 1: Error estimates

Training error	Test error	OOB error
0	0.2516	0.33