Random forests

Jeffrey Leek, Assistant Professor of Biostatistics

May 18, 2016

Random forests

1. Bootstrap samples
2. At each split, bootstrap variables
3. Grow multiple trees and vote

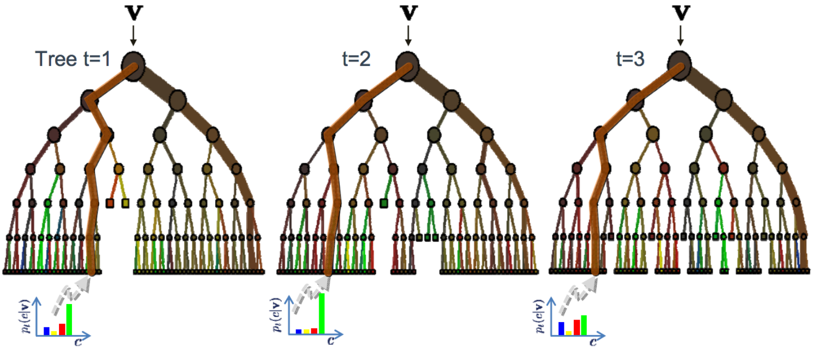
Pros:

1. Accuracy

Cons:

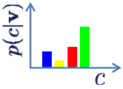
1. Speed
2. Interpretability
3. Overfitting

Random forests



The ensemble model

Forest output probability $p(c|\mathbf{v}) = \frac{1}{T}\sum_{t}^{T} p_t(c|\mathbf{v})$



http://www.robots.ox.ac.uk/~az/lectures/ml/lect5.pdf

Iris data

```r
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```r
data(iris); library(ggplot2)
inTrain <- createDataPartition(y=iris$Species,
                               p=0.7, list=FALSE)
training <- iris[inTrain,]
testing <- iris[-inTrain,]
```

Random forests

```
library(caret)
modFit <- train(Species~ .,data=training,method="rf",prox=T
```

```
## Loading required package: randomForest

## randomForest 4.6-12

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

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
modFit
```

```
## Random Forest
```

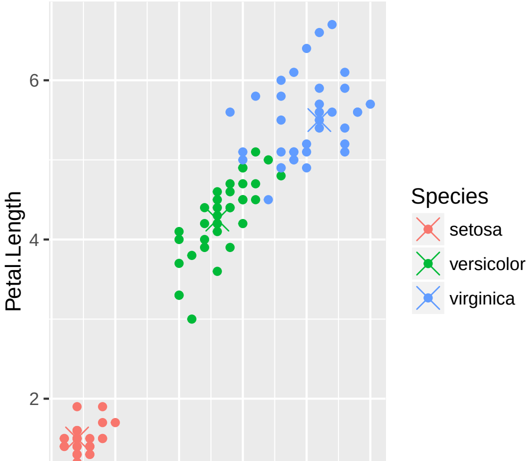
Getting a single tree

```r
getTree(modFit$finalModel,k=2)
```

```
##   left daughter right daughter split var split point sta
## 1             2              3         4        0.75
## 2             0              0         0        0.00
## 3             4              5         4        1.75
## 4             6              7         3        5.35
## 5             8              9         3        4.85
## 6             0              0         0        0.00
## 7             0              0         0        0.00
## 8             0              0         0        0.00
## 9             0              0         0        0.00
```

Class “centers”

```
irisP <- classCenter(training[,c(3,4)], training$Species, m
irisP <- as.data.frame(irisP); irisP$Species <- rownames(ir
p <- qplot(Petal.Width, Petal.Length, col=Species,data=trai
p + geom_point(aes(x=Petal.Width,y=Petal.Length,col=Species
```



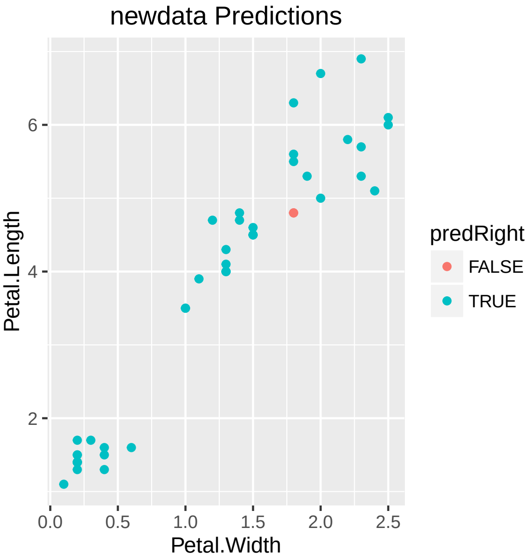
Predicting new values

```
pred <- predict(modFit,testing); testing$predRight <- pred=
table(pred,testing$Species)
```

```
##
## pred         setosa versicolor virginica
##   setosa         15          0         0
##   versicolor      0         15         2
##   virginica       0          0        13
```

Predicting new values

```
qplot(Petal.Width,Petal.Length,colour=predRight,data=testi
```



Notes and further resources

Notes:

- Random forests are usually one of the two top performing algorithms along with boosting in prediction contests.
- Random forests are difficult to interpret but often very accurate.
- Care should be taken to avoid overfitting (see rfcv funtion)

Further resources:

- Random forests
- Random forest Wikipedia
- Elements of Statistical Learning