Statistical Methods for Bioinformatics

II-5: Trees, Bagging and Boosting

We will have a look at the Heart dataset here, which contains data from a study into heart disease. A set of medical parameters are measured and the response variable is the occurrence of myocardial infarction (chd in the dataset). The main goal is to find a sufficient, but minimal model to predict the occurrence of a myocardial infarction. We will try Trees, Bagging and Boosting on the dataset. Start by loading the necessary libraries and the dataset.

```
library(tree)
library(randomForest)
library(gbm)
heart = read.csv('{PATH}SAheart.data', header=T,row.names=1)
```

The response variable is chd, and is binary. We need to cast it to a factor so the tree analysis makes a Classification tree. We also make a training and test set:

```
chdOriginal=heart$chd
heart$chd=as.factor(heart$chd)
set.seed(1)
train=sample(1:nrow(heart), nrow(heart)*2/3) test=(-train)
Now make the tree, look at the summary and plot it.
tr=tree(chd~.,data=heart, subset=train)
summary(tr)
plot(tr)
text(tr,pretty=0,cex=0.5)
```

- Do some splits provide the same outcome for both leafs? Explain your observation.

Now prune the tree to reduce its complexity using cross-validation. Select the classification error rate as criterium, and choose your number of terminal nodes.

```
cv.tr=cv.tree(tr,FUN=prune.misclass)
plot(cv.tr)
```

- What does the scale on the top refer to?

Now prepare the pruned tree and plot it. Then calculate the performance for both trees

```
prune.tree=prune.misclass(tr,best=6)
plot(prune.tree)
text(prune.tree,pretty=0,cex=0.65)
```

```
summary(prune.tree)
predtree = predict(tr,newdata=heart[test,],type="class")
table(heart$chd[test],predtree)
performanceTree1=length(which(predtree==heart$chd[test]))/length(heart$chd[test])

predtree2 = predict(prune.tree,newdata=heart[test,],type="class")

table(heart$chd[test],predtree2)
performanceTree2=length(which(predtree2==heart$chd[test]))/length(heart$chd[test])
```

- Explain any difference in performance
- Is this a good score? Explain your answer.

Continue by fitting bagged trees and a random forest. The procedure for both methods is the same, with the only difference the number of variables available for every tree split. This is the mtry variable

```
#do a randomForest
set.seed(1)
rf=randomForest(chd~.,mtry=(ncol(heart[train,])/3),data=heart, subset=train,
importance=TRUE, ntree=1000, nodesize=20)
predRF = predict(rf,newdata=heart[test ,])
table(heart$chd[test],predRF)
performanceRF=length(which(predRF==heart$chd[test]))/length(heart$chd[test])

performanceRF #do a bagged tree
set.seed(1)
bag=randomForest(chd~.,mtry=ncol(heart[,])-1,data=heart, subset=train,
importance=TRUE, ntree=1000,nodesize=20)
predbag = predict(bag,newdata=heart[test ,])
table(heart$chd[test],predbag)
performanceBag=length(which(predbag==heart$chd[test])) / length(heart$chd[test])

performanceBag
```

- Which performs better? Can you improve performance by doubling the number of trees? By changing the nodesize?

Now have a look at the Variable Importance for both procedures:

```
importance(rf)
varImpPlot(rf)
importance(bag)
varImpPlot (bag)
```

- Are there any differences between the approaches? Interpret what you find.

- Some variables are not very important, do you increase performance if you drop them from the analysis?

Now perform a boosted tree. We need to change the chd parameter back to 0 and 1, and specify the proper error family.

```
heart$chd=chdOriginal
gbm1 <-gbm(chd~.,data=heart[train,], distribution="bernoulli", n.trees=1000,
shrinkage=0.01, interaction.depth=3, n.minobsinnode = 10, cv.folds
= 5, keep.data=TRUE, verbose=FALSE,n.cores=1)
```

You can print some trees, but in a rather technical format!

```
print(pretty.gbm.tree(gbm1,1))
print(pretty.gbm.tree(gbm1,gbm1$n.trees))
```

Now look at what actually underlies the model, the predicted effect (marginal effect) of one variable to the response, and for two variables to the response:

```
plot(gbm1,7,best.iter)
plot(gbm1,9,best.iter)
plot(gbm1,c(7,9),best.iter)
```

- What does this plot show? Do you see evidence of non-linear effects?