Making Trees

Rian Fantozzi

10/17/2017

Name(s):

**Data setup**

We will be using data about patients with acute hepatitis and trying to classify which explanatory variables predict death of a patient.

Load the cleaned data.

library(readr)  
hep <- read\_csv("hepatitisCleanNoNA.csv")

## Parsed with column specification:  
## cols(  
## ID = col\_integer(),  
## Class = col\_integer(),  
## Age = col\_integer(),  
## Sex = col\_integer(),  
## Steriod = col\_integer(),  
## Antivirals = col\_integer(),  
## Fatigue = col\_integer(),  
## Malaise = col\_integer(),  
## Anorexia = col\_integer(),  
## LiverBig = col\_integer(),  
## LiverFirm = col\_integer(),  
## SpeelPalpable = col\_integer(),  
## Spiders = col\_integer(),  
## Acvites = col\_integer(),  
## Varacises = col\_integer(),  
## Bilirubis = col\_double(),  
## Sgot = col\_integer(),  
## Albumin = col\_double(),  
## Histology = col\_integer()  
## )

Now you'll need to take all of the quantitative variables that are categorical and use as.factor like in the example from class to make R treat them like categories called factors (needed for randomForests).

hep=mutate(hep, Class=as.factor(Class),Sex=as.factor(Sex), Steriod=as.factor(Steriod), Antivirals = as.factor(Antivirals), Fatigue=as.factor(Fatigue),Malaise=as.factor(Malaise),Anorexia=as.factor(Anorexia),LiverBig=as.factor(LiverBig),LiverFirm=as.factor(LiverFirm),SpeelPalpable=as.factor(SpeelPalpable),Spiders=as.factor(Spiders),Acvites=as.factor(Acvites),Varacises=as.factor(Varacises))  
head(hep)

## # A tibble: 6 x 19  
## ID Class Age Sex Steriod Antivirals Fatigue Malaise Anorexia  
## <int> <fctr> <int> <fctr> <fctr> <fctr> <fctr> <fctr> <fctr>  
## 1 1 2 30 2 1 2 2 2 2  
## 2 2 2 50 1 1 2 1 2 2  
## 3 3 2 78 1 2 2 1 2 2  
## 4 4 2 34 1 2 2 2 2 2  
## 5 5 2 34 1 2 2 2 2 2  
## 6 6 2 39 1 2 2 1 2 2  
## # ... with 10 more variables: LiverBig <fctr>, LiverFirm <fctr>,  
## # SpeelPalpable <fctr>, Spiders <fctr>, Acvites <fctr>,  
## # Varacises <fctr>, Bilirubis <dbl>, Sgot <int>, Albumin <dbl>,  
## # Histology <int>

Now let's pick a training set and a test set.

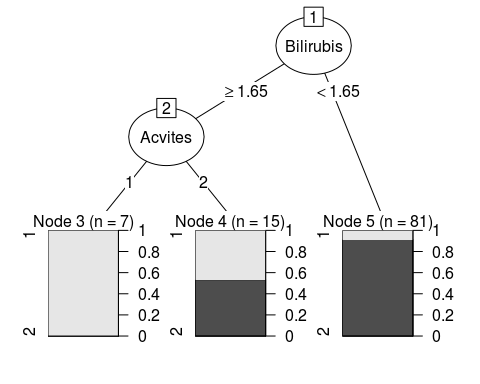
set.seed(1337334651)  
n=nrow(hep)  
testSample=sample.int(n,size=round(.2\*n))  
train=hep[-testSample,]  
test=hep[testSample,]

Our response variable will be Class. It has two values, 2=LIVE and 1=DIE. Note for sex, 2-male and 1=female. For yes/no categorical variables 2=No and 1=yes.

TreeOI=rpart(Class~., data=train, parms=list(split="information")) #the . tells R to use all the other variables  
TreeOI

## n= 103   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 103 21 2 (0.20388350 0.79611650)   
## 2) Bilirubis>=1.65 22 8 1 (0.63636364 0.36363636)   
## 4) Acvites=1 7 0 1 (1.00000000 0.00000000) \*  
## 5) Acvites=2 15 7 2 (0.46666667 0.53333333) \*  
## 3) Bilirubis< 1.65 81 7 2 (0.08641975 0.91358025) \*

plot(as.party(TreeOI))



forest=randomForest(Class~.,data=train, ntree=20000,mtry=4)  
forest

##   
## Call:  
## randomForest(formula = Class ~ ., data = train, ntree = 20000, mtry = 4)   
## Type of random forest: classification  
## Number of trees: 20000  
## No. of variables tried at each split: 4  
##   
## OOB estimate of error rate: 18.45%  
## Confusion matrix:  
## 1 2 class.error  
## 1 9 12 0.57142857  
## 2 7 75 0.08536585

import=importance(forest) #this gives importance. The rest is getting to display in decreasing order  
import

## MeanDecreaseGini  
## ID 3.3547174  
## Age 3.2594417  
## Sex 0.2617156  
## Steriod 0.4104319  
## Antivirals 0.2133486  
## Fatigue 0.3924210  
## Malaise 0.6883321  
## Anorexia 0.3147762  
## LiverBig 0.4368399  
## LiverFirm 0.5355543  
## SpeelPalpable 0.6744438  
## Spiders 1.1152279  
## Acvites 3.1896815  
## Varacises 0.8818172  
## Bilirubis 7.8687310  
## Sgot 2.8460353  
## Albumin 5.5969905  
## Histology 0.7808023

import=as.data.frame(import)  
import=rownames\_to\_column(import)  
import

## rowname MeanDecreaseGini  
## 1 ID 3.3547174  
## 2 Age 3.2594417  
## 3 Sex 0.2617156  
## 4 Steriod 0.4104319  
## 5 Antivirals 0.2133486  
## 6 Fatigue 0.3924210  
## 7 Malaise 0.6883321  
## 8 Anorexia 0.3147762  
## 9 LiverBig 0.4368399  
## 10 LiverFirm 0.5355543  
## 11 SpeelPalpable 0.6744438  
## 12 Spiders 1.1152279  
## 13 Acvites 3.1896815  
## 14 Varacises 0.8818172  
## 15 Bilirubis 7.8687310  
## 16 Sgot 2.8460353  
## 17 Albumin 5.5969905  
## 18 Histology 0.7808023

arrange(import,desc(MeanDecreaseGini))

## rowname MeanDecreaseGini  
## 1 Bilirubis 7.8687310  
## 2 Albumin 5.5969905  
## 3 ID 3.3547174  
## 4 Age 3.2594417  
## 5 Acvites 3.1896815  
## 6 Sgot 2.8460353  
## 7 Spiders 1.1152279  
## 8 Varacises 0.8818172  
## 9 Histology 0.7808023  
## 10 Malaise 0.6883321  
## 11 SpeelPalpable 0.6744438  
## 12 LiverFirm 0.5355543  
## 13 LiverBig 0.4368399  
## 14 Steriod 0.4104319  
## 15 Fatigue 0.3924210  
## 16 Anorexia 0.3147762  
## 17 Sex 0.2617156  
## 18 Antivirals 0.2133486

predTest=predict(forest,test,type="class")  
confFor=tally(predTest~ test$Class)  
confFor

## test$Class  
## predTest 1 2  
## 1 2 2  
## 2 1 21

(sum(diag(confFor)))/nrow(test)

## [1] 0.8846154

**Project directions** Your job is to write a short paper talking about this data and trees made with it.

Begin with a description of the data set. You will need to look up the variables that show up in the paper to see what they mean so you can talk about them in context.

First, make a tree in R using all possible predcitors predicting Class with the Gini coefficent. For the tree you will show a picture of the tree, show how it will classify one case in the data set, and evaluate the prediction ability on the test and training data using the confusion matrix.

Second, make a tree with the information gain in R.

Third, make a tree in Weka (copy and paste the Weka text output into R). The csv file will need a column name for the first column (call it Row), which is the one you can delete. You might convert most of the variables to nominal type (Filters -> Unsupervised -> attribute -> NumericToNominal and specify the column numbers) for nicer trees. You may then remove the Row number column that is explicit in the data. Recall the decision tree algorithm is J48 under Classifiers -> Trees.

Test mode: split 66.0% train, remainder test

=== Classifier model (full training set) ===

# RandomTree

Bilirubis < 1.65 | Acvites = 1 | | Albumin < 2.85 : 1 (2/0) | | Albumin >= 2.85 | | | Sgot < 86 | | | | Bilirubis < 0.5 : 2 (1/0) | | | | Bilirubis >= 0.5 : 1 (2/0) | | | Sgot >= 86 : 2 (3/0) | Acvites = 2 | | Bilirubis < 0.55 | | | Sex = 1 : 1 (2/0) | | | Sex = 2 : 2 (2/0) | | Bilirubis >= 0.55 | | | Fatigue = 1 | | | | Spiders = 1 | | | | | Bilirubis < 1.35 | | | | | | Albumin < 4.15 : 2 (9/0) | | | | | | Albumin >= 4.15 | | | | | | | Steriod = 1 : 2 (2/0) | | | | | | | Steriod = 2 | | | | | | | | Sgot < 46.5 : 1 (1/0) | | | | | | | | Sgot >= 46.5 : 2 (1/0) | | | | | Bilirubis >= 1.35 | | | | | | Sgot < 62.5 : 2 (2/0) | | | | | | Sgot >= 62.5 : 1 (2/0) | | | | Spiders = 2 : 2 (33/0) | | | Fatigue = 2 : 2 (41/0) Bilirubis >= 1.65 | Sex = 1 | | Acvites = 1 | | | Spiders = 1 : 1 (6/0) | | | Spiders = 2 | | | | Age < 42.5 : 2 (1/0) | | | | Age >= 42.5 : 1 (2/0) | | Acvites = 2 | | | Fatigue = 1 | | | | Malaise = 1 | | | | | Sgot < 211 | | | | | | Spiders = 1 | | | | | | | SpeelPalpable = 1 : 1 (1/0) | | | | | | | SpeelPalpable = 2 | | | | | | | | Age < 50 : 2 (3/0) | | | | | | | | Age >= 50 : 1 (1/0) | | | | | | Spiders = 2 : 1 (2/0) | | | | | Sgot >= 211 : 2 (2/0) | | | | Malaise = 2 | | | | | SpeelPalpable = 1 : 1 (2/0) | | | | | SpeelPalpable = 2 : 2 (3/0) | | | Fatigue = 2 : 1 (1/0) | Sex = 2 : 2 (2/0)

Size of the tree : 51

Time taken to build model: 0.01 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances 36 81.8182 % Incorrectly Classified Instances 8 18.1818 % Kappa statistic 0.3973 Mean absolute error 0.1818 Root mean squared error 0.4264 Relative absolute error 61.5929 % Root relative squared error 120.8479 % Total Number of Instances 44

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class  
 0.667 0.158 0.400 0.667 0.500 0.417 0.754 0.312 1  
 0.842 0.333 0.941 0.842 0.889 0.417 0.754 0.929 2

Weighted Avg. 0.818 0.309 0.867 0.818 0.836 0.417 0.754 0.845

=== Confusion Matrix ===

a b <-- classified as 4 2 | a = 1 6 32 | b = 2

Fourth, make a random forest in R.

Finally, compare the first tree to the trees to those generated in the second, third and fourth steps. Your comparison will include what variables are new to the tree/importance list of the forest and the prediction ability on the trainig and test set.

Conclude with what variables seem to be important to predicting Class and which tree you think should be used and why. Essay The data set includes 20 attribute information. We can classify the status of the patient (dead or live) by looking at the class variables. Age and sex variables also provide the general information about the patients. The variables from 4 to 20 give us the detailed information about the health condition of the patient and the medical treatment they have used. The tree in R has Bilirubis as the root. It has two chilren: >=1.85 or =<1.85, which are the level measure of bilirubin in blood. The tree has the the activities as the branch. For the level measure of bilirubin is smalleer than 1.85, the tree classify the people have fluid buildup in the abdominal cavity. If it's larger or equal 1.5, the tree classsify the level of bilirubin in blood. Node 2, 4,6,7 are the leaves of the tree, which are the final result. The root is presented both with the observed data from the bilirubin in the R's tree and Weka' tree. However, the tree made in Weka has 2 children, which are activities and sex. The tree in Weka contains variables that are not shown in R, such as ALBUMIN, FATIGUE, SPIDERS, Age, MALAISE, SGOT, SPLEEN PALPABLE.