R package for Weighted Random Forest? classwt option?

I'm trying to use Random Forest to predict the outcome of an extremely imbalanced data set (the minority class rate is about only 1% or even less). Because the traditional Random Forest algorithm minimizes the overall error rate, rather than paying special attention to the minority classes, it is not directly applicable on imbalanced data. So I want to assign a high cost to misclassification of the minority class (cost sensitive learning).

I read several sources that we can use the option `classwt` of `randomForest` in R, but I don't know how to use this. And do we have any other alternatives to the `randomForest` funtion?

r random-forest

edited Aug 17 '15 at 13:11
Antoine
1,454 7 25

asked Jun 19 '15 at 11:50
Matemattica
529 4 20

1 Answer

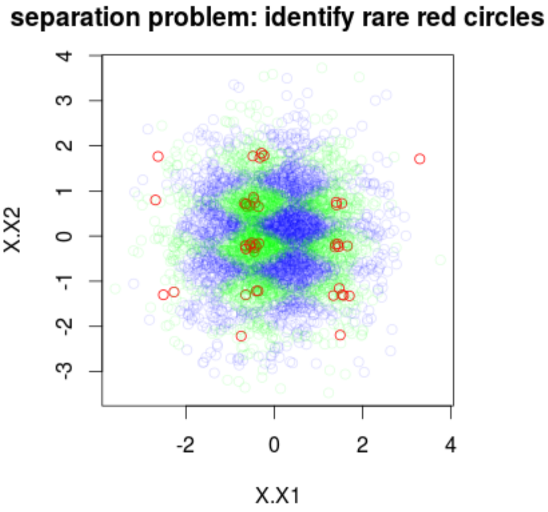
This thread refers to two other threads and a fine article on this matter. It seems classweighting and downsampling are equally good. I use downsampling as described below.

Remember the training set must be large as only 1% will characterize the rare class. Less than 25~50 samples of this class probably will be problematic. Few samples characterizing the class will inevitably make the learned pattern crude and less reproducible.

RF uses majority voting as default. The class prevalences of the training set will operate as some kind of effective prior. Thus unless the rare class is perfectly separable, it is unlikely this rare class will win a majority voting when predicting. Instead of aggregating by majority vote, you can aggregate vote fractions.

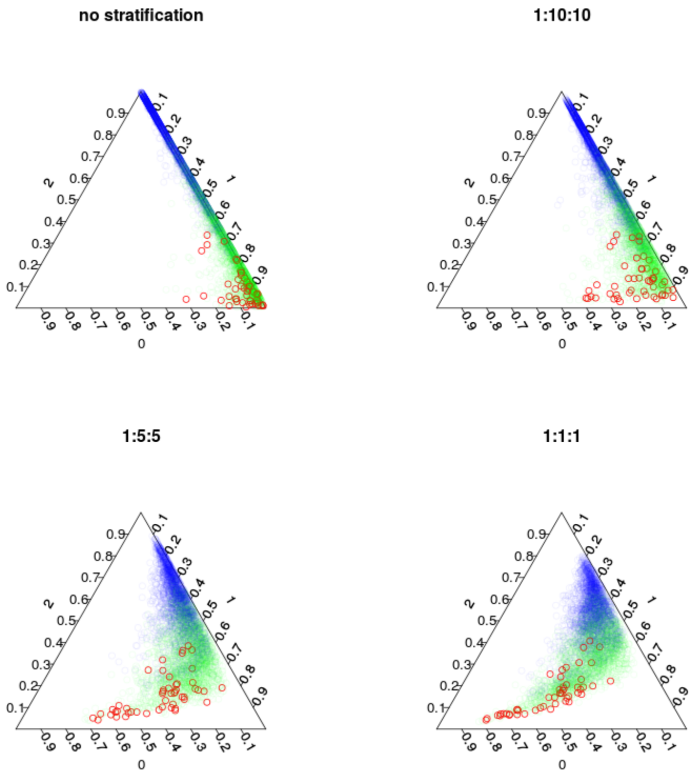
Stratified sampling can be used to increase the influence of the rare class. This is done on the cost on downsampling the other classes. The grown trees will become less deep as much fewer samples need to be split therefore limiting the complexity of the potential pattern learned. The number of trees grown should be large e.g. 4000 such that most observations participate in several trees.

In the example below, I have simulated a training data set of 5000 samples with 3 class with prevalences 1%, 49% and 50% respectively. Thus there will 50 samples of class 0. The first figure shows the true class of training set as function of two variables x1 and x2.

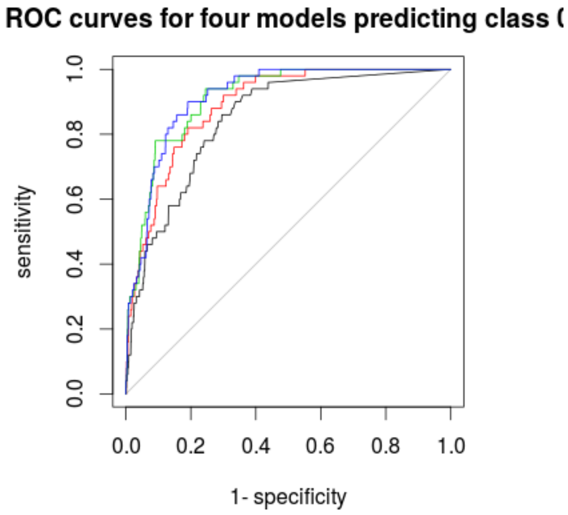


Four models were trained: A default model, and three stratified models with 1:10:10 1:2:2 and 1:1:1 stratification of classes. Main while the number of inbag samples(incl. redraws) in each tree will be 5000, 1050, 250 and 150. As I do not use majority voting I do not need to make a perfectly balanced stratification. Instead the votes on rare classes could be weighted 10 times or some other decision rule. Your cost of false negatives and false positives should influence this rule.

The next figure shows how stratification influences the vote-fractions. Notice the stratified class ratios always is the centroid of predictions.



Lastly you can use a ROC-curve to find a voting rule which gives you a good trade-off between specificity and sensitivity. Black line is no stratification, red 1:5:5, green 1:2:2 and blue 1:1:1. For this data set 1:2:2 or 1:1:1 seems best choice.



By the way, vote fractions are here out-of-bag crossvalidated.

And the code:

```
library(plotrix)
library(randomForest)
library(AUC)

make.data = function(obs=5000,vars=6,noise.factor = .2,smallGroupFraction=.01) {
X = data.frame(replicate(vars,rnorm(obs)))
yValue = with(X,sin(X1*pi)+sin(X2*pi*2)+rnorm(obs)*noise.factor)
yQuantile = quantile(yValue,c(smallGroupFraction,.5))
yClass = apply(sapply(yQuantile,function(x) x<yValue),1,sum)
yClass = factor(yClass)
print(table(yClass)) #five classes, first class has 1% prevalence only
Data=data.frame(X=X,y=yClass)
}

plot.separation = function(rf,...) {
triax.plot(rf$votes,...,col.symbols = c("#FF0000FF",
                                           "#00FF0010",
                                           "#0000FF10")[as.numeric(rf$y)])
}

#make data set where class "0"(red circles) are rare observations
#Class 0 is somewhat separateble from class "1" and fully separateble from class "2"
Data = make.data()
par(mfrow=c(1,1))
```

```
plot(Data[,1:2],main="separation problem: identify rare red circles",
     col = c("#FF0000FF","#00FF0020","#0000FF20")[as.numeric(Data$y)])

#train default RF and with 10x 30x and 100x upsumpling by stratification
rf1 = randomForest(y~.,Data,ntree=500, sampsize=5000)
rf2 = randomForest(y~.,Data,ntree=4000,sampsize=c(50,500,500),strata=Data$y)
    rf3 = randomForest(y~.,Data,ntree=4000,sampsize=c(50,100,100),strata=Data$y)
rf4 = randomForest(y~.,Data,ntree=4000,sampsize=c(50,50,50)  ,strata=Data$y)

#plot out-of-bag pluralistic predictions(vote fractions).
par(mfrow=c(2,2),mar=c(4,4,3,3))
plot.separation(rf1,main="no stratification")
plot.separation(rf2,main="1:10:10")
plot.separation(rf3,main="1:5:5")
plot.separation(rf4,main="1:1:1")

par(mfrow=c(1,1))
plot(roc(rf1$votes[,1],factor(1 * (rf1$y==0))),main="ROC curves for four models predicting
class 0")
plot(roc(rf2$votes[,1],factor(1 * (rf1$y==0))),col=2,add=T)
plot(roc(rf3$votes[,1],factor(1 * (rf1$y==0))),col=3,add=T)
plot(roc(rf4$votes[,1],factor(1 * (rf1$y==0))),col=4,add=T)
```

edited Oct 5 '15 at 17:57
Antoine
1,454 7 25

answered Jun 21 '15 at 21:06
Soren Havelund Welling
2,871 5 17

oups one figure caption says 1:5:5 instead of 1:2:2 – Soren Havelund Welling Jun 21 '15 at 21:36

thank you very much for your detailed answer, that will definitely help me a lot in my daily work. There is one sentence that I don't understand: "Main while the number of inbag samples(incl. redraws) in each tree will be 5000,1050, 250 and 150" .Could you please explain me where does the numbers come from? – Matemattica Jun 22 '15 at 12:26

1 my pleasure ;) in this example the rare class had 50 members. If stratifying 1:10:10 we would need to specify sampsize=c(50,500,500). 50+500+500 = 1050. A fully grown tree of 1050 samples will have 1050x2 nodes in total. – Soren Havelund Welling Jun 22 '15 at 13:07

Sorry if my question is idiot, but what is the meaning of 1:10:10, 1:2:2 and 1:1:1 stratification here? And when you said "the votes on rare classes could be weighted 10 times". Which part of the code represents that? Is it 1:10:10? Thank you very much! – Matemattica Jun 24 '15 at 10:35

1 1:10:10 are the ratios between the classes. The simulated data set was designed to have the ratios 1:49:50. These ratios were changed by down sampling the two larger classes. By choosing e.g. sampsize=c(50,500,500) the same as c(1,10,10) * 50 you change the class ratios in the trees. 50 is the number of samples of the rare class. If you furthermore set keep.inbag=TRUE and inspect rf$inbag, you will see that samples of the rare classes is inbag in ~2/3 trees whereas each non-rare class sample is included in very few trees because of down sampling. – Soren Havelund Welling Jun 24 '15 at 14:46

So, if I understand correctly, it's "down sampling" technique, not weighted random forest? – Matemattica Jun 24 '15 at 14:55

...as stated in third sentence of answer. – Soren Havelund Welling Jun 24 '15 at 14:57

Ok, perfect. Thank you very much for your kind answer:) – Matemattica Jun 24 '15 at 16:07

Add Another Answer