Inferring variable contribution to variance explained for random forests

I am trying to understand if there is a way to approximate what portion of the variance explained is being contributed by each independent variable in a random forest model. Just for illustration, I am borrowing the following model from the Stanford StatLearning class notes. This builds a random forest model for predicting median housing prices in Boston using the dataset provided with the `MASS` package.

```
require(randomForest)
require(MASS)
set.seed(101)
dim(Boston)
train=sample(1:nrow(Boston),300)
```

Fitting the model (just using a simple model here without any validation just for illustration)

```
rf.boston=randomForest(medv~.,data=Boston,subset=train)
rf.boston
```

I get the following output

```
Call:
 randomForest(formula = medv ~ ., data = Boston, subset = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 4

          Mean of squared residuals: 12.34243
                    % Var explained: 85.09
```

Now `R` tells me that this model explains 85.09% variance in median housing prices. Additionally, I can run the `importance` command to figure out what variables turned out to be "significant" in my model.

```
importance(rf.boston)

        IncNodePurity
crim        1487.1777
zn           142.0280
indus        965.7756
chas         234.6918
nox         1741.9305
rm          7435.3378
age          655.6031
dis         1357.3411
rad          316.3278
tax          794.0953
ptratio     1858.7183
black        455.5382
lstat       6947.9121
```

Is there a way to use these two pieces of information (or using some other approach) to tell us what percentage of `85.09` was explained by `crim`, `zn` and so on.

My goal here is to show this as a 100% stacked bar graph ordered by variable importance illustrating major drivers of the dependent variable (median housing prices in this example). Overall, I want to see if we can get outputs akin to shapely value regression as shown here (esp slide 21) using random forests.

r | machine-learning | random-forest | importance

asked Mar 7 at 5:31

sriramn
106 3

1 Answer

Great question, unfortunately the answer is not so straight forward. Pieces of information are not necessarily additive but can also be redundant or complimentary. Thus, it is somewhat crude to assign each variable a fixed percentage of useful information/explained variance etc. Nevertheless I guess it should be possible to exactly relate IncNodePurity to %explainedVar. I guess you would need to correct IncNodePurity for nodesizes. In worst case recursively revisit all nodes to do this.

PermutationVariableImportance is often preferred over IncNodePurity (importance=TRUE). Permutation variable importance is in a particular sense a uni-variate variable selection method, because only one variable is permuted at the time. Vivian W. Ng and Leo Breiman wrote a paper on "Bivariate variable selection for classification problem" which emphasizes this problem.

Speaking of the linked slide show, I'm a little skeptic to this consultancy style presentation, when it comes to what kind of conclusions could be made from a "***%explained variance attributed to each variable analysis***". Remember relation does not imply causation (xkcd.com/552). It would be failure of logic to infer that because some variable contribute by some metric ~80% to predict e.g. costumer satisfaction, then the company should allocate ~80% or some other proportion of the resources to improve costumer satisfaction. That said,

citing the above XKCD comic: *"Correlation doesn't imply causation, but it does waggle its eyebrows suggestively and gesture furtively while mouthing 'look over there' "*

[EDIT] Oh it turns out you do not have to correct for node size as the IncNodePurity metric is sum of squares for regression, but IncNodePurity is still not exactly related %Explained variance. %Explained is out-of-bag cross-validated, whereas IncNodePurity is sum of squares of the inbag samples. You could redefine the IncNodePurity as a OOB term, but you would have to recompute sum of squares of out-of-bag samples recursively. R is not very fast to execute millions of recursive function calls, so it is better implemented with Rcpp, and then one spend hours hacking something, which is not very great in the first place.

Because IncNodePurity is not cross-validated and tend to answer a less central question, you should really get to know permutation variable importance. It is not that abstract and can actually be used with virtually any model. For regression variable importance is typically the change of out-of-bag %explained variance, when a given variable is permuted after training, but before prediction.

I think the most honest % visualization you could do, would be to normalize the most important feature to 100% and set absolute 0 variable importance to 0%. Any other feature would have a percentage in between(perhaps slightly negative).

Anyways here is a code example on IncNodePurity, where the training set have some $y_{ss} = \sum_{i=1}^{N} (\overline{y} - y_i)^2$ where N is train set size. In this special case all samples participate in all trees and the terminal nodesize=1, therefore the squared residuals $(\hat{y} - y)_{ss} = 0$. Then, %explained variance is not-defined as no observation is ever OOB, but then the sum of IncNodePurities(INP) do relate exactly to y_{ss}

$\sum_{j=1}^{p} INP_j = y_{ss} - (\hat{y} - y)_{ss} = y_{ss} - 0 = y_{ss}$ where p is number of features.

```
set.seed(1)
library(randomForest)
ss = function(x) sum((mean(x)-x)^2)
#data
X = data.frame(replicate(3,rnorm(500)))
y = with(X,X1^2+X2-X3)
#model
rf = randomForest(X,y,replace=F,sampsize = length(y),nodesize=1)
ss(y)-sum(rf$importance) #equals some number smaller than 1e-9
-----
[1] -3.001333e-11
```

edited Mar 7 at 21:05

answered Mar 7 at 14:22

Soren Havelund Welling
2,871 5 17

2 It is also worth bootstrapping the entire process. I'll bet that the variable contributions are quite unstable, and a bootstrap confidence interval for the rank of each variable's contribution will reveal the true difficulty of the task. – Frank Harrell Mar 7 at 14:43

@Soren: Thanks! I share your skepticism on such metrics and would usually prefer showing elasticities by running a bunch of sensitivity tests. However, I was going for a "quick win" due to time constrains. Can you explain in a bit more detail how I can should be correcting for node sizes? Also, I have seen the PermutationVariableImportance score but really did not understand what goes into it - seems like that would be a better approach? – sriramn Mar 7 at 17:12

You're welcome :) Variable importance is worth the effort to learn. You could reasonably normalize variable importance (or IncNodePurity) to percentages as explained in edit, but the values would still not add up to anything particular. – Soren Havelund Welling Mar 7 at 20:56

Add Another Answer