# Does it makes sense to use feature selection before Random Forest?

Everything is in the title, does it makes sense to use feature selection before using random forest?
Thank you

machine-learning | feature-selection | random-forest

asked Mar 9 at 19:12
marcL
17 4



## 1 Answer

Yes it does and it is quite common. If you expect more than ~50% of your features no
are redundant but utterly useless. E.g. the randomForest package has the wrapper fu
rfcv() which will pretrain a randomForest and omit the least important variables. rfcv function refer to this chapter. Remember to embed feature selection + modeling in a outer cross-validation loop to avoid over optimistic results.

**[edit below]**

I could moderate "utterly useless". A single random forest will most often not as e.g. regression with lasso regularization completely ignore features, even if these (in simulated hindsight) were random features. Decision tree splits by features are chosen by local criteria in any of the thousands or millions of nodes and cannot later be undone. I do not advocate cutting features down to one superior selection, but it is for some data sets possible to achieve substantial increase in prediction performance (estimated by a repeated **outer** cross-validation) using this variable selection. A typical finding would be that keeping 100% of features or only few percent work less well, and then there can be a broad middle range with similar estimated prediction performance.

Perhaps a reasonable thumb rule: When one expect that lasso-like regularization would serve better than a ridge-like regularization for a given problem, then one could try pre-training a random forest and rank the features by the **inner** out-of-bag cross-validated *variable importance* and try drop some of the least *important* features. *Variable importance* quantifies how much the cross-validated model prediction decreases, when a given feature is permuted(values shuffled) after training, before prediction. One will never be certain if one specific feature should be included or not, but it likely much easier to predict by the top 5% features, than the bottom 5%.

From a practical point of view, computational run time could be lowered, and maybe some resources could be saved, if there is a fixed acquisition cost per feature.

edited Mar 10 at 22:16

answered Mar 10 at 13:28
Soren Havelund Welling
2,871 5 17

3 The ability of data to tell you that a feature is useless is severely limited, and I hope the option to which you refer is integrated into the random forest algorithm. It would not be appropriate to do up-front deletion of features before sending the candidate features to the random forest algorithm. – Frank Harrell Mar 10 at 13:47

@FrankHarrell, I have tried to elaborate my answer – Soren Havelund Welling Mar 10 at 22:02

I am suspicious about variable selection improving accuracy. Are you using a strictly proper accuracy scoring rule? – Frank Harrell Mar 10 at 22:42

Accuracy scoring rule should reflect intended use of model. Different rules may lead to different drop ratios and selections. The method has been described in several papers. Sometimes the actual selection is the purpose, such as in gene identification in GWAS studies or parcel selection associated with brain stimuli in fMRI. Again drop ratio may depend if one want predict or try to come with a narrowed relationship model. One may bootstrap aggregate the entire selection+model process. Why RF itself can't ignore features is related to the random variable subspace regularisation(mtry). – Soren Havelund Welling Mar 11 at 0:18

If one is new to RF or when N observations >> p features, I would not bother much to drop features. – Soren Havelund Welling Mar 11 at 0:25

1 I disagree that you choose different scoring rules for different purposes. An improper accuracy scoring rule leads to selection of the wrong features and giving them the wrong weights. More apparent is the arbitrariness in certain scoring rules. It is far better to choose an optimum predictive model and then using solid decision theory to make optimum decisions using that model. This is done by applying a utility function to the continuous predictions. – Frank Harrell Mar 11 at 12:22

I still think winning contributions on kaggle are likely to contain modest feature selection. I recognize your point of you from many other answers. Do you have a good chapter or paper, so I can get to learn your point of view better? – Soren Havelund Welling Mar 11 at 14:11

Sections 4.3, 4.5, 5.4 and Chapter 10 of my book *Regression Modeling Strategies* for which detailed Course Notes may help. Kaggle probably used force-choice classification without proper incorporation of a utility function. In many cases, feature selection takes away some of the energy in the system that would better be reserved for pure prediction. Think Maxwell's demon. – Frank Harrell Mar 11 at 15:34

@FrankHarrell - can you give a detailed answer to this question? apparently you have some strong arguments against doing feature selection... – ihadanny Mar 21 at 12:30

1 The best way to learn about this is to do rigorous bootstrap internal validation of a procedure that tries to do feature selection vs. one that does not. Quite often the predictive discrimination (when measured using a proper accuracy scoring rule or even with the $c$-index (ROC area)) is better when feature selection is not attempted. Feature selection is almost always arbitrary. – Frank Harrell Mar 21 at 12:52

Add Another Answer