Bagging & Boosting

Andrea Spano’

2023-12-18

Random Forest

Introduction

- Random Forest is often considered to be a panacea of all data science problems. On a funny note, when you can’t think of any algorithm (irrespective of situation), use random forest!
- Random Forest is a versatile machine learning method capable of performing both *regression* and *classification* tasks.
- Five Basic concepts:

  1. Ensemble
  2. Bagging
  3. Sampling
  4. OOB
  5. Importance

Ensembling

- When building a tree model, the algorithm works by making the best possible choice at each particular stage, without any consideration of whether those choices remain optimal in future stages.
- That is, the algorithm makes a locally optimal decision at each stage
- It is thus quite possible that such a choice at one stage turns out to be sub-optimal in the overall scheme
- We can try to learn from multiple trees. Just be sure they do not just learn all the same

Bagging

- **Bagging: Bootstrap aggregating** is a method to learn from multiple trees
- Take m samples with replacement from original data of dimension $n \times k$ so that each sample is of dimensions $n \times k$
- Fit a fully grown tree model (not pruned) on each sample. Minimum size of terminal nodes is usually one for classification trees and five for regression trees.
- When predicting, we calculate the predictions on a the test set along all trees
- Combine the predictions results into a single prediction by
    - average for regression trees
    - majority vote for classification tree

Bagging

- Pros
  - Combing multiple results reduce the prediction variances
  - Easy to paralelize
- Cons
  - Higly correlated trees. Bagging builds trees based on the same set of predictors, few strong predictors will be repeatedly selected. This leads to generating similar trees that produce highly correlated predictions. One problem with correlated predictions is that taking the average of those does not decrease variance as expected.

Bagging

- An average of B i.i.d random variables, each with variance σ^2, has variance: $\frac{\sigma^2}{B}$
- In case of B i.d., identical but not independent, with ρ pair correlation, then the variance is:

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

- As B increases the second term disappears but the first term remains

Sampling

- As in bagging, we build a number of decision trees on bootstrapped training samples (bagging)
- Each time a split in a tree is considered, a random sample of m predictors is chosen as split candidates from the full set of p predictors.
- In Random forest: $m << p$ usually:
  - $m = \frac{1}{2}\sqrt{p}$
  - $m = \sqrt{p}$
  - $m = 2\sqrt{p}$
- Note that if $m = p$, then this is bagging.

OOB

- Remember, in bootstrapping we sample with replacement, and therefore not all observations are used for each bootstrap sample. On average $\frac{1}{3}$ of them are not used
- We call them out-of-bag samples (OOB)
- We can predict the response for the i_{th} observation using each of the trees in which that observation was OOB and do this for n observations
- Calculate overall $OOB\ MSE$ or classification error

Variable Importance Measure

- Bagging results in improved accuracy over prediction using a single tree
- Unfortunately, the resulting model becomes difficult to interpret.
- Bagging improves prediction accuracy at the expense of interpretability.
- A proxy for model interpretation consists of calculating the total amount that the RSS or Gini index is decreased due to splits over a given predictor and averaged over all B trees.
- We do this for each tree over all its corresponding OOB samples to get the mean and SD
- This importance score gives an indication of how useful the variables are for prediction

Random Forests Tuning

- The inventors of random Forest make the following recommendations:
  - For classification, the default value for m is $\sqrt(p)$ and the minimum node size is one.
  - For regression, the default value for m is $\frac{p}{3}$ and the minimum node size is $five$.
- In practice the best values for these parameters will depend on the problem, and **they should be treated as tuning parameters**.
- We can use OOB data to perform cross-validation along the way.
- Once the OOB error stabilizes, the training can be terminated: *caret*

Overfitting

- Random forests *cannot overfit* data with regards to to the number of trees.
- The number of trees does not mean increase in the flexibility of the model

Boosting

Recap

- Boosting is a general approach that can be applied to many statistical learning methods for regression or classification.
- Bagging: Generate multiple trees from bootstrapped data and average the trees. Bagging results in i.d. trees and not i.i.d.

Random Forest produces i.i.d trees by randomly selecting a subset of predictors at each step

Boosting

- Boosting works very differently
- Boosting does not involve bootstrap sampling
- Trees are grown **sequentially**:
- Each tree is grown using information from previously grown trees
- Like bagging, boosting involves combining a large number of decision trees

Sequential fitting

- Given a tree model, we fit a decision tree to the residuals from the model. Response variable now is the residuals and not Y
- We then add this new decision tree into the fitted function in order to update the residuals
- The learning rate has to be controlled in order to avoid over fitting

Sequential fitting Example

Assume we have a simple model M with acciracy A_M so that:

$$y = M(x) + \epsilon_M$$

Assuming that ϵ_M is noy simple white noise but has some correlation with y, we can develope a model like:

$$\epsilon_M = G(x) + \epsilon_G$$

that may has have better accuracy A_G where $A_G \geq A_M$

walking by :

$$\epsilon_G = H(x) + \epsilon_H$$

that may has have better accuracy A_H where $A_H \geq A_G$

Sequential fitting Example

After fitting M, G amd H we can combine them together:

$$y = M(x) + G(x) + H(x) + \epsilon_h$$

The newly created model may even have better accuracy that A_H.

That we can further improve if we can find a *optimal* combination of parameters: $[\alpha, \beta, \gamma]$ so that:

$$y = \alpha M(x) + \beta G(x) + \gamma H(x) + \epsilon$$

Boosting is generally made on *weak leanrnes*. that do not have the ability to return white noise residuals

Boosting can easily lead to overfitting. A stop rule is strongly required