Subscribe to DSC Newsletter

All Blog Posts My Blog + Add

Random Forests explained intuitively

Posted by Manish Kumar Barnwal on June 1, 2017 at 12:30am View Blog

Random Forests algorithm has always fascinated me. I like how this algorithm can be easily explained to anyone without much hassle. One quick example, I use very frequently to explain the working of random forests is the way a company has multiple rounds of interview to hire a candidate. Let me elaborate.

Say, you appeared for the position of Statistical analyst at WalmartLabs. Now like most of the companies, you don't just have one round of interview. You have multiple rounds of interviews. Each one of these interviews is chaired by independent panels. Each panel assesses the candidate separately and independently. Generally, even the questions asked in these interviews differ from each other. **Randomness** is important here.

The other thing of utmost importance is **diversity**. The reason we have a panel of interviews is that we assume a committee of people generally takes better decision than a single individual. Now, this committee is not any collection of people. We make sure that the interview panel is a little diversified in terms of topics to be covered in each interview, the type of questions asked, and many other details. You don't go about asking the same question in each round of interviews.

After having all the rounds of interviews, the final call whether to select or reject the candidate is based on the majority of the decision from each panel. If out of 5 panels of interviewers, 3 recommends a hire and two against a hire, we tend to go ahead with selecting the candidate. I hope you get the gist.

If you have heard about decision tree, then you are not very far from understanding what random forests are. There are two keywords here - random and forests. Let us first understand what forest means. Random forest is a collection of many decision trees. Instead of relying on a single decision tree, you build many decision trees say 100 of them. And you know what a collection of trees is called - a forest. So you now understand why is it called a forest.

Why is it called random then?

Say our dataset has 1,000 rows and 30 columns. There are two levels of randomness in this algorithm:

- **At row level**: Each of these decision trees gets a random sample of the training data (say 10%) i.e. each of these trees will be trained independently on 100 randomly chosen rows out of 1,000 rows of data. Keep in mind that each of these decision trees is getting trained on 100 randomly chosen rows from the dataset i.e they are different from each other in terms of predictions.
- **At column level**: The second level of randomness is introduced at the column level. Not all the columns are passed into training each of the decision trees. Say we want only 10% of columns to be sent to each tree. This means a randomly selected 3 column will be sent to each tree. So for the first decision tree, may be column C1, C2 and C4 were chosen. The next DT will have C4, C5, C10 as chosen columns and so on.

Let me draw an analogy. Let us now understand how interview selection process resembles a random forest algorithm. Each panel in the interview process is actually a decision tree. Each panel gives a result whether the candidate is a pass or fail and then a majority of these results is declared as final. Say there were 5 panels, 3 said yes and 2 said no. The final verdict will be yes.

Something similar happens in the random forest as well. The results from each of the tree are taken and the final result is declared accordingly. Voting and averaging is used to predict in case of classification and regression respectively.

With the advent of huge computational power at our disposal, we hardly think for even a second before we apply random forests. And very conveniently our predictions are made. Let us try to understand other aspects of this algorithm.

随机森林的几个缺点

When is a random forest a poor choice relative to other algorithms?

1. **Random forests don't train well on smaller datasets** as it fails to pick on the pattern. To simplify, say we know that 1 pen costs INR 1, 2 pens cost INR 2, 3 pens cost INR 6. In this case, linear regression will easily estimate the cost of 4 pens but random forests will fail to come up with a good estimate.
2. **There is a problem of interpretability with random forest.** You can't see or understand the relationship between the response and the independent variables. Understand that random forest is a predictive tool and not a descriptive tool. You get variable importance but this may not suffice in many analysis of interests where the objective might be to see the relationship between response and the independent features.
3. The **time taken to train random forests** may sometimes be too huge as you train multiple decision trees. Also, in the case of a categorical variable, the time complexity increases exponentially. For a categorical column with n levels, RF tries split at 2^n -1 points to find the maximal splitting point. However, with the power of H2O we can now train random forests pretty fast. You may want to read about H2O at H2O in R explained.
4. In the case of a regression problem, **the range of values response variable can take** is determined by the values already available in the training dataset. Unlike linear regression, decision trees and hence random forest can't take values outside the training data.

What are the advantages of using random forest?

1. Since we are using multiple decision trees, **the bias remains same as that of a single decision tree**. However, the variance decreases and thus we decrease the chances of overfitting. I have explained bias and variance intuitively at The curse of bias and variance.
2. When all you care about is the predictions and **want a quick and dirty way out**, random forest comes to the rescue. You don't have to worry much about the assumptions of the model or linearity in the dataset.

I will add in the R code snippets as well to get an idea of how this is executed soon.

I write more on data science, machine learning and life experiences at my blog. Please stop by.

Did you find the article useful? If you did, share your thoughts in the comments. Share this post with people who you think would enjoy reading this. Let's talk more of data-science.

Most Popular Content on DSC

To not miss this type of content in the future, subscribe to our newsletter.

- Book: Classification and Regression In a Weekend - With Python
- Book: Applied Stochastic Processes
- Long-range Correlations in Time Series: Modeling, Testing, Case Study
- How to Automatically Determine the Number of Clusters in your Data
- New Machine Learning Cheat Sheet | Old one
- Confidence Intervals Without Pain - With Resampling
- Advanced Machine Learning with Basic Excel
- New Perspectives on Statistical Distributions and Deep Learning
- Fascinating New Results in the Theory of Randomness
- Fast Combinatorial Feature Selection

Other popular resources

- Comprehensive Repository of Data Science and ML Resources
- Statistical Concepts Explained in Simple English
- Machine Learning Concepts Explained in One Picture
- 100 Data Science Interview Questions and Answers
- Cheat Sheets | Curated Articles | Search | Jobs | Courses
- Post a Blog | Forum Questions | Books | Salaries | News

Archives: 2008-2014 | 2015-2016 | 2017-2019 | Book 1 | Book 2 | More

Follow us: Twitter | Facebook

Views: 41554

Tags: algorithms, data, forests, learning, machine, random, science

Next Post >

Comment

You need to be a member of Data Science Central to add comments!

Join Data Science Central

Comment by Rohit on March 8, 2018 at 10:19pm

Hi Manish,

1st, kudos on such an intuitive yet simple description of RF.

I would like to discuss #1 of the "Advantages of RF" a bit further...

I've always wondered whether the true motivation of RF's is "just variance reduction" compared to a single DT or should it rather be "both variance and bias reduction". Please note that overfitting of a single DT can be due to both variance and bias. The later will depend on the depth ofcourse. Now we know that RF's are ensembles and each tree is fully grown (no pruning). Thus deep trees reduce bias and averaging the ensemble reduces the variance.

Now, your comment from #1: "**the bias remains same as that of a single decision tree**. However, the variance decreases" ... is based on the assumption that the single DT is a deep tree, correct? In practice we don't grow single trees too deep due to the overfitting error scare. Overfitting is due to both bias and variance and RF's reduce both. Would like to read your thoughts.

Thanks

Rohit

Comment by Vinod Sharma on September 15, 2017 at 4:01am

I find this post succinct and informative for me !! Thank you

Comment by Manish Kumar Barnwal on June 4, 2017 at 8:46pm

Hi Rohan,

Glad you enjoyed reading it.

Decision trees are the base classifiers for random forests. And we know the way decision tree predicts is to take the average of all the observations at the leaf node. And so the value it predicts cannot be out of the response values in the training data.

However, the same is not true for linear regression. The beta coefficients are estimated and the predicted response value is not bounded in any way like a decision tree.

Comment by Rohan Kotwani on June 4, 2017 at 4:48pm

Thanks for this information! I find it very well thought out and intuitive.

I have a question about 4. in where random forests are a disadvantage. How does linear regression take values outside of the training dataset? I interpreted this as random forests do no extrapolate very well.

RESOURCES

- Subscribe to DSC Newsletter
- Free Books
- Forum Discussions
- Cheat Sheets
- Jobs
- Search DSC
- DSC on Twitter
- DSC on Facebook

VIDEOS

DSC Webinar Series: Scale AI/ML with Data Wrangling Featuring Forrester

Added by Tim Matteson

DSC Webinar Series: How Does Tableau Help You Ask the Right Questions?

Added by Tim Matteson

DSC Webinar Series: Adding Optimization to Your Analytics Toolbox

Added by Tim Matteson

+ Add Videos View All