Bernoulli Random Forests: Closing the Gap between Theoretical Consistency and Empirical Soundness

Paper by:

- Yisen Wang
- Qingtao Tang
- Shu-Tao Xia
- Jia Wu
- Xingquan Zhu

Analysis by:

- Mayur Garg (160100085)
- Malhar Jadhav (160100023)
- Shirish Shubhanker (160040013)
- Nautatava Navlakha (160040007)

What are random forest trees?

- Random Forest is a supervised ensemble learning method for classification and regression.
- It operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Why the paper

A decision tree will greedily choose the best option from the choices, whereas many other methods will use them all. Ensemble methods such as random forests negate this to a certain extent, but you lose the ease of understanding.

This paper focuses on the understandability of random forest tree to get a level of understanding of what’s really going on while maintaining the performance.

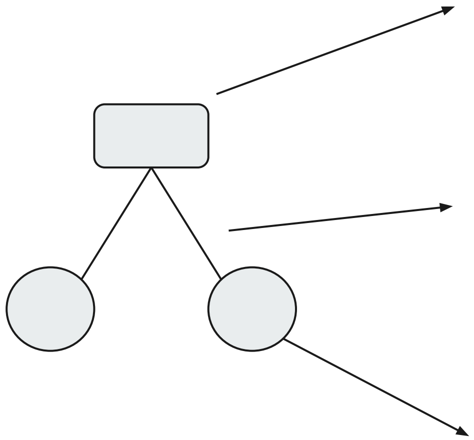
Approach

Normally, the original random forest construction is dependent on data and thus, calculation of theoretical consistency becomes difficult.

The approach here is to introduce two simplifications in tree construction: using two independent Bernoulli distributions - one used in choosing attributes for each node of the tree, the other controls splitting point used by each node

In the proposed BRF, we also replaced traditional bootstrap technique with data point partitioning.

Using this approach makes BRF less dependent on data and thus theoretical consistency is achieved. The accuracy converges to optimum as training data grows infinitely large.



Bernoulli distribution B1 with probability ‘p1’ to select attributes.
B1 = 0 => sqrt(D) attribs, B1 = 1 => 1 attribs.

Bernoulli distribution B2 with probability ‘p2’ to select splitting point.
B2 = 0 => random sampling, B2 = 1 => impurity optimisation.

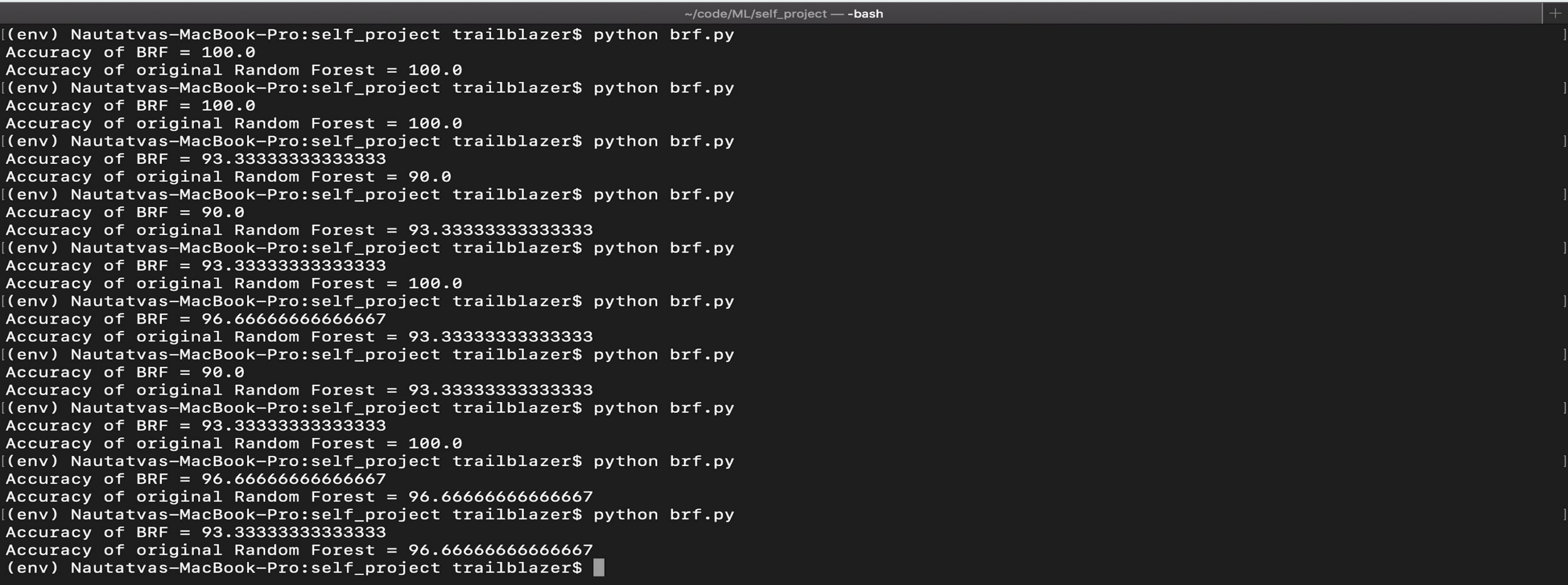
Continue as above till stopping criteria is met.

Implementation

We implemented the given variant of BRF and found the accuracy to be nearly equal almost every time.

The test was performed on the **iris dataset** given by sklearn. Dataset is divided into 80% training and 20% testing data.

50% of the training data goes to creating the skeleton/structure of BRF and the other 50% form the nodes for estimation while fitting the data.



```
~/code/ML/self_project — -bash
(env) Nautatvas-MacBook-Pro:self_project trailblazer$ python brf.py
Accuracy of BRF = 100.0
Accuracy of original Random Forest = 100.0
(env) Nautatvas-MacBook-Pro:self_project trailblazer$ python brf.py
Accuracy of BRF = 100.0
Accuracy of original Random Forest = 100.0
(env) Nautatvas-MacBook-Pro:self_project trailblazer$ python brf.py
Accuracy of BRF = 93.33333333333333
Accuracy of original Random Forest = 90.0
(env) Nautatvas-MacBook-Pro:self_project trailblazer$ python brf.py
Accuracy of BRF = 90.0
Accuracy of original Random Forest = 93.33333333333333
(env) Nautatvas-MacBook-Pro:self_project trailblazer$ python brf.py
Accuracy of BRF = 93.33333333333333
Accuracy of original Random Forest = 100.0
(env) Nautatvas-MacBook-Pro:self_project trailblazer$ python brf.py
Accuracy of BRF = 96.66666666666667
Accuracy of original Random Forest = 93.33333333333333
(env) Nautatvas-MacBook-Pro:self_project trailblazer$ python brf.py
Accuracy of BRF = 90.0
Accuracy of original Random Forest = 93.33333333333333
(env) Nautatvas-MacBook-Pro:self_project trailblazer$ python brf.py
Accuracy of BRF = 93.33333333333333
Accuracy of original Random Forest = 100.0
(env) Nautatvas-MacBook-Pro:self_project trailblazer$ python brf.py
Accuracy of BRF = 96.66666666666667
Accuracy of original Random Forest = 96.66666666666667
(env) Nautatvas-MacBook-Pro:self_project trailblazer$ python brf.py
Accuracy of BRF = 93.33333333333333
Accuracy of original Random Forest = 96.66666666666667
(env) Nautatvas-MacBook-Pro:self_project trailblazer$
```

Results

After the prediction, we get a average testing accuracy of as high as 94.67%.

Avg. Random forest classifier accuracy = 96.33%.

References

Paper: https://www.ijcai.org/Proceedings/16/Papers/309.pdf

https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-lea
rning/

Code link

https://drive.google.com/file/d/1wiYk-EbMI_ZWrVtvul8jUhmBZHmvIXsH/view?usp=sharing

Thank You