Accelerated Bayesian Additive Regression Trees

Hytham Farah

Supervisor: Prof. Russell Steele

Introduction

BACKGROUND

A common problem in statistics is to predict the value of an unknown response variable y given a set features $x_1, \dots, x_p$. BART and XBART are prediction machines, which learn from labeled datasets to predict y from $x_1, \dots, x_p$.

HISTORY

In 2010 the BART algorithm was proposed and in 2019 XBART was proposed as alternative algorithm which can preserve the predictive accuracy of BART while improving speed.

GOAL

Our goal is to test weather XBART manages to both improve the speed and preserve the accuracy of BART.

STRATEGY

- Generate our own data, where we know the true relationship between the variable y and the features $x_1, \dots, x_p$.
- Train BART and XBART on half the data.
- Use BART and XBART to generate predictions on the other half.
- Calculate the mean squared error (MSE) of the prediction.

If XBART is at least as accurate as BART, then the MSE of XBART should be, on average, at most as high as BART's MSE.

SIGNIFIGANCE OF THE RESEARCH

It is standard practice to present a new algorithm alongside a demonstration of its performance on a generated dataset. The purpose of this research project is to test these algorithms on a much larger spectrum of generated data to see how consistent the results are across different kinds of data.

What is a Regression Tree?

SINGLE TREE MODEL

Split the predictor space into distinct regions. For each region assign a single y value to every observation that falls within it.

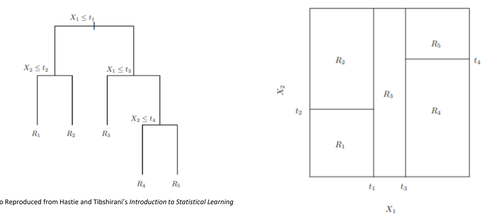

Photo Reproduced from Hastie and Tibshirani's *Introduction to Statistical Learning*

ADDITIVE TREE MODEL

The idea here is to generate a large ensemble of single trees, that are small. Each of which explain a portion of the overall relationship between the response and the features.

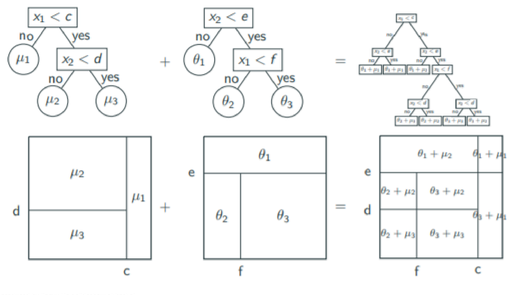

Photo reproduced by Jingyu He's slideshow at bart.ai

Workflow

	Input	Action	Output
Step 1	Training Dataset	Fit the Model	Approximation of a function
Step 2	Test Dataset	Predict using the approximate function	Predicted values of response
Step 3	Predictions and the True Value	Average the difference between prediction and truth squared	Mean Squared Error

Methodology

FEATURES

We begin by choosing a distribution for our features $x_1, \dots, x_p$. This will influence how often we see certain values.

- **Normal**: Values closer to the mean are more likely to occur.
- **Uniform:** Values are distributed equally across the spectrum.
- **Exponential:** Values are concentrated at the left tail and become increasingly less likely as they become larger.

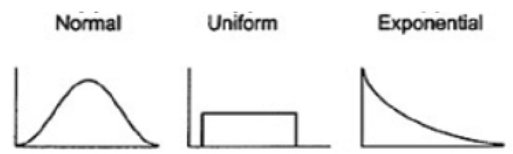

Photo Reproduced from website https://ecstep.com/the-normal-distribution/

We also decide weather or not the variables are independent of each other. Note that variable W is dependent on variable Z if knowledge of the value of Z gives us knowledge of X's value.

RESPONSE

$$y = f(x_1, \dots, x_p) + \varepsilon \qquad \epsilon \sim \text{Normal}(0, 1)$$

We are assuming that there is a functional relationship between the features and the response, plus some statistical error. Therefore, in order to generate y we need a function with our features as the arguments.

Here is an example of 3 functions used in the analysis:

1. $10\sin(\pi x_1 x_2) + 20(x_3 - .5)^2 + 10x_4 + 5x_5$
2. $\max\{x_1, x_2, x_3\}$
3. $x_1 + 2x_2 + 18x_3 + 7x_4 + 3x_5$

ASSESSING THE ERROR

Suppose we have trained two models A, and B, and we wish to test them, and suppose our (rather small) testing dataset on the left and the predictions on the right.

y (hidden)	x_1	x_2	x_3
6	.4	.12	.813
2	.21	.524	.6
3	.341	.77	.234

Model A	Model B
5	8
2	4
2	3

To calculate the MSE, we would simply subtract the predicted value of y from the true value of y. We square this error so that the number remains positive. Then we divided it by the number of observations:

$$MSE_A = (6-5)^2 + (2-2)^2 + (3-2)^2 = 2$$

The general formula is as follows:

$$\frac{1}{n}\sum_{i=1}^{n} \left| y_{true} - y_{predicted} \right|^2$$

Results

A particularly surprising result came when running the code with the following parameters:

- Features were normally distributed and serially generated
- Training Dataset Size = 1000
- Number of features = 10

The different models

	Linear Regression <dbl>	bartMachine <dbl>	dbarts <dbl>	xbartMSE <dbl>
Friedman	9.73253095	0.83919401	1.2824705	356570.4993
Linear	46.23606036	47.23631062	49.2410208	105.4869
Single Index	311.55013388	313.99955995	322.4652864	1474173.8870
Trig + Poly	2.26399299	0.17679900	0.4082993	42082.6679
Max	0.03999915	0.06141121	0.1432680	817.7554
Constant	0.01673042	0.04199380	0.1156522	37756.2483
No Interaction	0.01673042	0.31636145	0.7656812	471369.2872
Average Runtime (sec.)	0.04333333	105.42666667	73.6226667	277.1183

Function Names

Mean Squared Error (MSE)

The MSE of XBART is astoundingly higher than the MSE of all the other functions. Also, strangely enough, the runtime of XBART was more than twice as long as both the Bart Machine and dbarts implementation of the BART algorithm. Note that a standard linear model works well in cases where there are not complicated interactions between the variables and in cases where the function is linear.

Bellow is a table of the parameters that we used to generate different kinds of data:

Distribution	Sample Size	# of Features	Dependence
Uniform	100, 1000, 10000	10, 30	Independent
Normal	100, 1000	10	Serially Dependent
Exponential	100, 1000	10	.7 Correlation

Conclusion

XBART as it currently stands is not an adequate replacement for BART. It neither is as accurate and even the speed is lacking sometimes.

References

Chipman, Hugh A., et al. "BART: Bayesian Additive Regression Trees." *The Annals of Applied Statistics*, vol. 4, no. 1, 2010, pp. 266–298., doi:10.1214/09-aoas285.

He, Jingyu, et al. *XBART: Accelerated Bayesian Additive Regression Trees*. 2019.

James, Gareth, et al. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2017.

Kapelner, Adam, and Justin Bleich. "BartMachine: Machine Learning with Bayesian Additive Regression Trees." *Journal of Statistical Software*, vol. 70, no. 4, 2016, doi:10.18637/jss.v070.i04.