Introduction to the bartMachine R package

Saint Louis R User Group

John Snyder

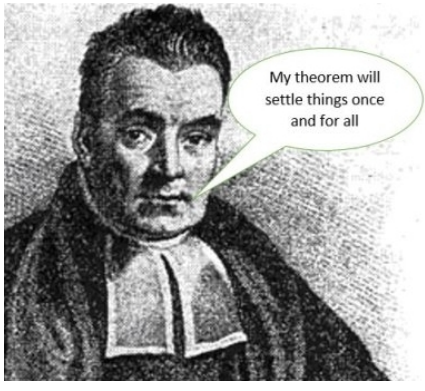
April 11, 2019

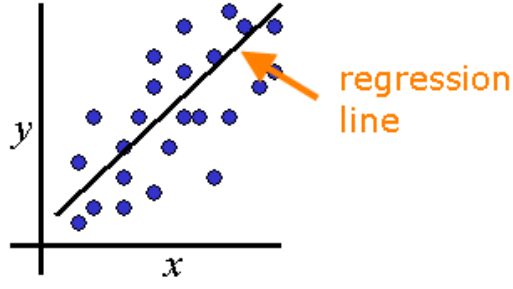
Outline

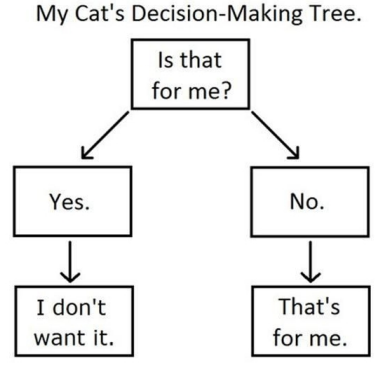
1. Brief BART overview
2. Installation and features
3. Demo
4. Further Considerations

What is BART?

Bayesian **A**dditive **R**egression **T**rees







How it works

- Emsamble method which is the sum of many shallow trees.
- Complexity is regularized via Bayesian "priors."
  - This frees us from ad hoc decisions
- Uses "Bayesian Backfitting"
  - Each tree is sequentially exposed to the residuals when all other trees are used to predict

Results:

- Each tree describes a tiny amount of the structure
- The Bayesian structure means variation is fully quantified.
  - Intervals, p-values, and model selection oh my!
- Outperforms many common models in out of sample prediction.

Powerful Predictive Performance

- Test RMSE of 100 random datasets simulated from various nonlinear functions (added noise with s=1)

Function	BART	XGBoost*	Random Forest*	Linear Reg(lol)
Friedman	1.08	1.21	1.64	2.61
Mirsha's Bird	1.53	2.78	2.90	26.59
Weird Exp	1.04	1.05	1.07	6.08
Linear	1.025	1.032	1.034	1.004

- bartMachine is relatively unknown
  - xgboost: ~43k downloads per month
  - randomForest: ~88k downloads per month
  - bartMachine: ~2k downloads per month

Package Features:

- Functions for Cross Validation
- Model fitting:
  - Is done in parallel[1]
  - Can incorporate missing data
- Lots of fun statistical things
  - Credible iterval calculation
  - Diagnostic plots/tests
- Variable selection
- Interaction detection
- Export fit trees

[1] MCMC sampling is used, so speedups during model fitting aren't great

Installation and loading steps

1. Google "How to install rJava on [your OS]"
2. Do that
3. Run the following

```
install.packages("bartMachine")
```

To load the package with:

- 10GB of memory
- All but one core available for compute

```
options(java.parameters = "-Xmx10g")
library(bartMachine)
numcores <- parallel::detectCores()
set_bart_machine_num_cores(numcores - 1)
```

Code Time

Coding demo

John's Final Thought

- BART is a powerful technique which brings many advantages
  - At the expense of computational efficiency.

```
# n is 10000, p is 100
bart.model <- bartMachine(X,y,
                          num_trees = 100,
                          num_burn_in = 1000,
                          num_iterations_after_burn_in = 5000,
                          mem_cache_for_speed = TRUE)
```

- Statistical advantages are numerous
- Great for small to mid sized data
- Good results with removing expected variation and feeding residuals into BART.