BayesRandomForest: An R implementation of Bayesian Random Forest for Regression Analysis of High-dimensional Data

Oyebayo Ridwan Olaniran*, Mohd Asrul Affendi Bin Abdullah

Department of Mathematics and Statistics, Faculty of Applied Sciences and Technology, Universiti Tun Hussein Onn Malaysia, Pagoh Educational Hub, 84600 Pagoh, Johor, Malaysia.

rid4stat@yahoo.com; afendi@uthm.edu.my

iCMS 2017, Langkawi, Malaysia.

November 8, 2017

Outline

Introduction

- The growth in computer applications have enhanced collection and analysis of big datasets.
- Big datasets are often referred to as high dimensional data in statistical parlance.
- The difficulties faced while analyzing big datasets has led to development of many statistical or machine learning procedures in the recent time.

Some high dimensional data scenarios

Table: High dimensional data scenarios

Scenario	**Sample size**	**Covariates**
Microarray studies	$n \leq 100$	$p = 40K$ genes
Pattern Recognition	$n \leq 100$	$p = 65K$ pixels
Text mining	$n \leq 10K$ documents	$p = 100K$ words
Genome Wide A. studies	$n \leq 2K$ subjects	$p = 500K$ SNPs

Some past and recent solutions

- Classification and Regression Trees (CART) (Breiman et al., 1984)
- Support Vector Machine (Vapnik, 1995)
- Least Absolute Shrinkage Selection Operator (LASSO) (Tibshirani, 1996)
- Random Forests (Breiman, 2001)
- Gradient Boosting Machine (Friedman, 2002)
- Least Angle Regression (LARS) (Efron et al., 2004)
- Artificial Neural Network (ANN) (Hastie, et al., 2010)
- Naive Bayes Classifier (Barber, 2012)
- Bayesian Additive Regression Trees (BART) (Chipman et al, 2010, Pratola; 2016)
- Bayesian Forests (Taddy, 2015)

What is Random Forest?

- Random Forest (RF) is a tree based method for regression analysis of low or high dimensional data.
- RF is often used with the later because it relaxes dimensionality assumption.
- RF major strengths are distribution free property and wide applicability to most real life problems.

Graphical representation of CART

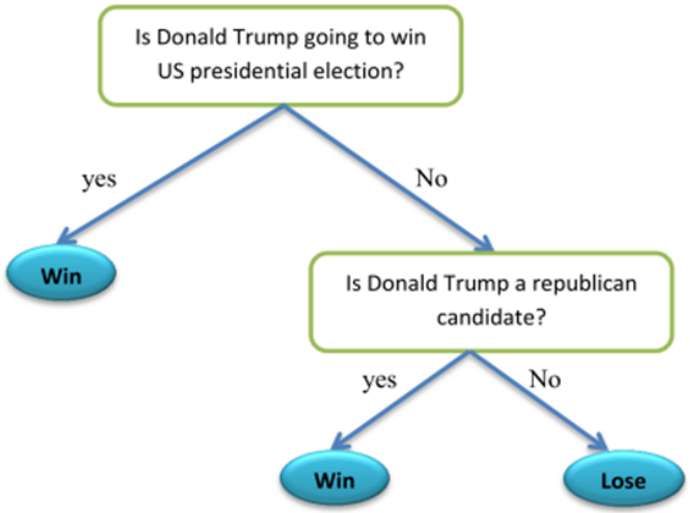

Figure: Graphical representation of CART

Random Forest Algorithm

Algorithm 1: Random Forest for Regression or Classification

1. For iteration $b = 1, 2, ..., B$ do:
   a. Draw a bootstrap sample $D^*$of size N from the training data D with one response variable Y and p predictor variables X.
   b. Grow a CART tree T_b to the bootstrapped data D^*, by iteratively repeating the following steps for each terminal node or leaf of the tree, until the minimum node size n_{min} is reached.
      i. Draw $m = \sqrt{p}$ variables at random from the p variables.
      ii. Pick the best variable/split-point among the m.
      iii. Split the node into two daughter nodes.
2. Print the ensemble of trees T_b over B iterations. To predict test data x, apply: $Regression: \hat{f}_{rf}^{B} = 1/B \sum_{b=1}^{B} \hat{T}_b(x)$
   $Classification: \hat{C}_{rf}^{B} = majorityvote\, \hat{C}_b(x)^{B}$,where $\hat{C}_b(x)$ is the class prediction of random forest tree.

Issues with Random Forests

When p is large and the number of relevant variables is small:

- The subsample size $m \approx \sqrt{p}$ or $p/3$ used by RF does not take into account the number of relevant variables, thus the chance of selecting irrelevant variables increases with increased p.
- Choosing m by cross validation increases RF computational time and do not often results in optimal m.
- Increasing m towards p increases the correlation between adjacent trees which eventually results in loss of efficiency.

Current Research

- To develop Bayesian Random Classification and Regression Forests for high dimensional data.

The Idea

- The goal of this research is to update every facet of the random forest algorithm using Bayesian reasoning in order to obtain a statistical learning model rather than a machine learning algorithm. Thus starting with the first step of Algorithm 1, which is bootstrapping, our intention is to provide a Bayesian framework that does simple random sampling with replacement. Also, the step b(ii) in algorithm 1 refers to random subset of m predictors out of a total of p predictors. RF uses traditional simple random sampling without replacement, but we intend to update with a Bayesian counterpart.

contd'

Bayesian simple random sampling with replacement

Suppose we let $G = (G_1, G_2, ..., G_N)$ be the actual variable of interest in the population with size N and $g = (g_1, g_2, ..., g_n)$ is the sample drawn with size $n = N$. If we define $I = (I_1, I_2, ..., I_N)$ be the set of inclusion indicator for G_i in g. Thus under simple random sampling with replacement $p(I = h)$ converges in probability to $bernoulli(1, \pi)$. Where π is the probability of inclusion of G_i in g. Formally;

$$p(I = h) = \pi^h (1 - \pi)^{(1-h)}, h = 0, 1$$

contd'

Bayesian simple random sampling with replacement

To perform Bayesian inference we need to specify the likelihood of inclusion indicator and their corresponding prior. However, the use of conjugate $beta(a, b)$ prior distribution for the binomial or Bernoulli likelihood is well documented in the literature (Lee, 2012; Lesaffre and Lawson, 2013). Thus, the posterior inference for the probability of inclusion π is;

$$p(\pi|I) = \frac{L(\pi|I) \times p(\pi)}{\int L(\pi|I) \times p(\pi)\, d\pi} \tag{1}$$

contd'

Bayesian simple random sampling with replacement

After little algebra, the posterior distribution is given by;

$$p(\pi|I = h, a, b) = \frac{\Gamma(n + a + b)}{\Gamma(h + a)\Gamma(n - h + b)} \times \pi^{a-1} \times (1 - \pi)^{b-1}, 0 \leq \pi \leq 1 \tag{2}$$

Therefore the Bayesian estimate of π is;

$$\hat{\pi}_B = \frac{a + h}{a + b + n} \tag{3}$$

For $h = 1$, implies;

$$\hat{\pi}_B = \frac{a + 1}{a + b + n} \tag{4}$$

contd'

Bayesian simple random sampling without replacement

Now for SRSWOR, where I assumes an hypergeometric distribution with parameters N, R, n, and R is the total number of target sample point of G in g and $N-R$ is its complement (excluded sample point). Thus if n random samples are to chosen from N with $n \leq N$. The probability distribution of the random variable I is given by as;

$$p(I = h|N, R, n) = \frac{\binom{R}{h}\binom{N-R}{n-h}}{\binom{N}{n}}; max(0, n - N + R) \leq h \leq min(n, R) \quad (5)$$

contd'

Bayesian simple random sampling without replacement

Peskun (2016) defined a discrete $ABC(N, a, b)$ conjugate distribution as a special case of polya or beta-binomial distribution (Dyer and Pierce, 1993). Thus, for an hypergeometric likelihood with R target outcomes the $ABC(N, a, b)$ conjugate distribution for $R - h$ is given by;

$$p(R|N, a, b) = \frac{\binom{a+R}{a}\binom{b+N-R}{b}}{\binom{a+b+N+1}{a+b+1}}; R = 0, 1, ..., N \tag{6}$$

contd'

The posterior distribution is thus;

$$p(R|I=h,N,a,b)=\frac{\frac{\binom{R}{h}\binom{N-R}{n-h}}{\binom{N}{n}}\frac{\binom{a+R}{a}\binom{b+N-R}{b}}{\binom{a+b+N+1}{a+b+1}}}{\sum_{R=1}^{N-n+h}\frac{\binom{R}{h}\binom{N-R}{n-h}}{\binom{N}{n}}\frac{\binom{a+R}{a}\binom{b+N-R}{b}}{\binom{a+b+N+1}{a+b+1}}};h\leq R\leq N-n+h \tag{7}$$

After little algebra;

$$p(R|I=h,N,a,b)=\frac{\binom{a+R}{a+h}\binom{b+N-R}{b+n-h}}{\binom{a+b+N+1}{a+b+1}};h\leq R\leq N-n+h \tag{8}$$

contd'

Bayesian simple random sampling without replacement

The posterior estimate of R is given by;

$$\hat{R}_B = \frac{n(a+R)}{a+b+N+1} \tag{9}$$

Also, the probability of inclusion of a member of R in n subsample of N is;

$$\hat{\pi}_{B2} = \frac{a+\hat{R}_B}{a+b+N+1} \tag{10}$$

Bayesian Random Forests

Algorithm 2: Proposed Bayesian Random Forest for Regression or Classification

1. For iteration $b = 1, 2, ..., B$ do:
   a. Draw a BSRSWR $D^*$of size N from the training data D with one response variable Y and p predictor variables X.
   b. Grow a Bayesian CART tree T_b^* to the BSRSWR data D^*.
      i. Draw $m = \sqrt{p}$ variables using BSRSWOR from the p variables.
      ii. Pick the best variable/split-point among the m.
      iii. Split the node into two daughter nodes.
2. Print the ensemble of trees T_b^* over B iterations. To predict test data x, apply: $Regression: \hat{f}_{rf}^B = 1/B \sum_{b=1}^{B} \hat{T}_b^*(x)$
   $Classification: \hat{C}_{rf}^B = majorityvote \hat{C}_b^*(x)$,where $\hat{C}_b^*(x)$ is the class prediction of Bayesian random forest tree.

Application in R

To facilitate the applicability of Bayesian Random Forests algorithm, we implemented the procedure in R via package "BayesRandomForest". Specifically, we modified the R package "randomForestSRC" using the scheme explained earlier.

Illustration

To illustrate BayesRandomForest we adapted the simulation scheme of (Friedman, 1992; Chipman et al., 2010; 2017; Hernandez et al., 2015). The simulation scheme involves $x_1, \ldots, x_p$ which are $iid \approx U(0,1)$ random variables and $\epsilon \approx N(0,1)$. The model used was formulated as;

$$y = 10sin(x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \epsilon$$

where the first five variables are relevant and $p - 5$ are irrelevant.

R Lab: Load library

```
> library(BayesRandomForest)
Loading required package: randomForestSRC

 randomForestSRC 2.5.0

 Type rfsrc.news() to see new features, ch
anges, and bug fixes.


> ## Example 1
>
> ## n = 100,  p = 1000
>
> friedman = friedmandata(100,1000)
>
> x = friedman[,-1]
>
> y = friedman[,1]
```

R Lab: Example 1, n = 100, p = 1000

```
> BayesRandomForest(x,y)
                           Sample size: 100
                    Number of trees: 500
          Forest terminal node size: 5
      Average no. of terminal nodes: 19.792
No. of variables tried at each split: 334
              Total no. of variables: 1000
                           Analysis: RF-R
                             Family: regr
                     Splitting rule: mse
               % variance explained: 75.49
                           Error rate: 222.72

>
> rfsrc(y~., data = friedman,ntree = 500)
                           Sample size: 100
                    Number of trees: 500
          Forest terminal node size: 5
      Average no. of terminal nodes: 18.62
No. of variables tried at each split: 334
              Total no. of variables: 1000
                           Analysis: RF-R
                             Family: regr
                     Splitting rule: mse
               % variance explained: 63.13
                           Error rate: 335
```

R Lab: Example 2, n = 100, p = 10000

```
> friedman = friedmandata(100,10000)
>
> x = friedman[,-1]
>
> y = friedman[,1]
>
> BayesRandomForest(x,y)
                         Sample size: 100
                      Number of trees: 500
           Forest terminal node size: 5
       Average no. of terminal nodes: 18.872
No. of variables tried at each split: 3334
              Total no. of variables: 10000
                            Analysis: RF-R
                              Family: regr
                     Splitting rule: mse
                % variance explained: 69.66
                          Error rate: 275.66

>
> rfsrc(y~., data = friedman,ntree = 500)
                         Sample size: 100
                      Number of trees: 500
           Forest terminal node size: 5
       Average no. of terminal nodes: 18.262
No. of variables tried at each split: 3334
              Total no. of variables: 10000
                            Analysis: RF-R
                              Family: regr
                     Splitting rule: mse
                % variance explained: 56.13
                          Error rate: 398.62
```

Comparison

The predictive performance of BayesRandomForest (BRF) was compared with Random Forests (RF), Gradient Boosting Machine (GBM), Bayesian Additive Regression Trees (BART) and Bayesian Forests (BF). The methods were assessed using 10 folds cross validation of Root Mean Squared Error (RMSE) at sample size $n = 50, 100$ and variables $p = [100, 500, 1000, 5000, 10$
The default parameters were assumed for the methods.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n_{test}} (y_i - \hat{y}_i)^2}{n_{test}}}$$

R Lab: Comparison 1, n = 50

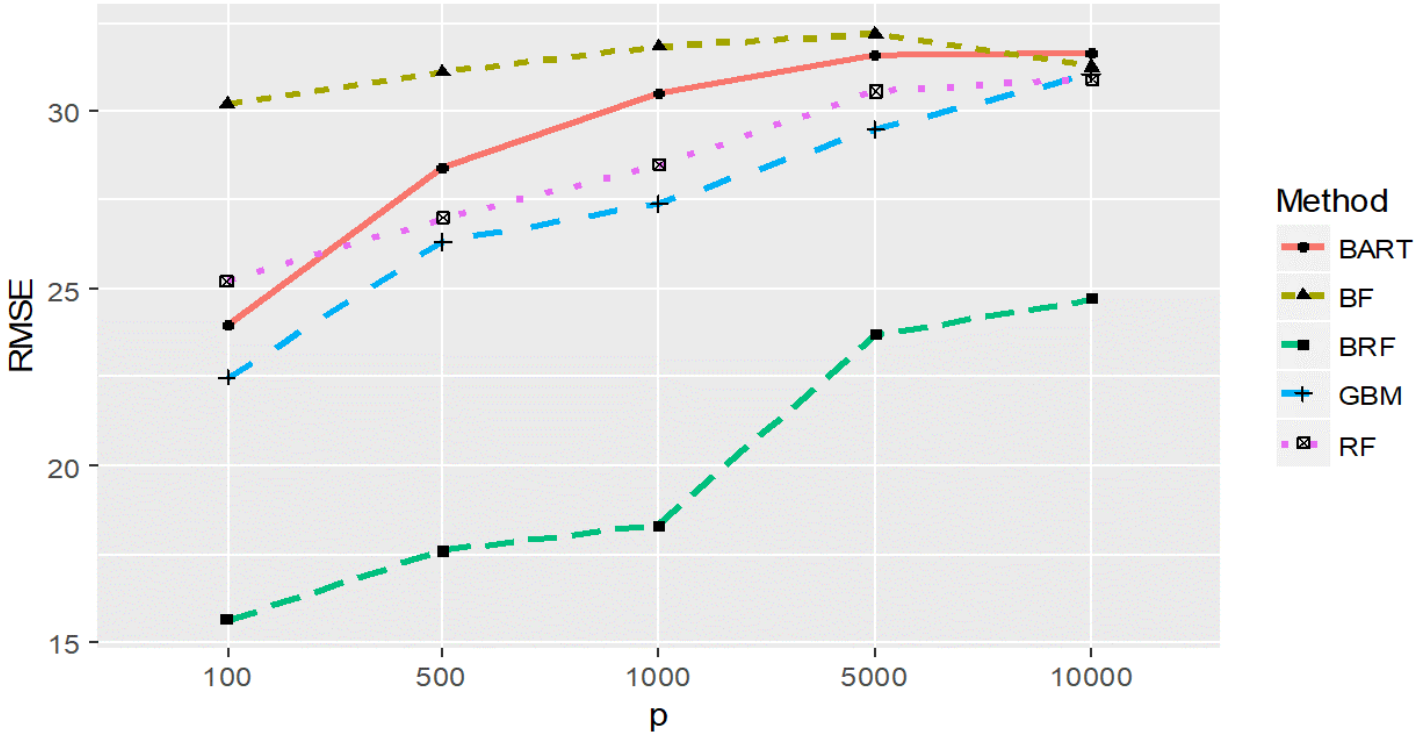

Figure: RMSE with increasing p at sample size $n = 50$

R Lab: Comparison 2, n = 100

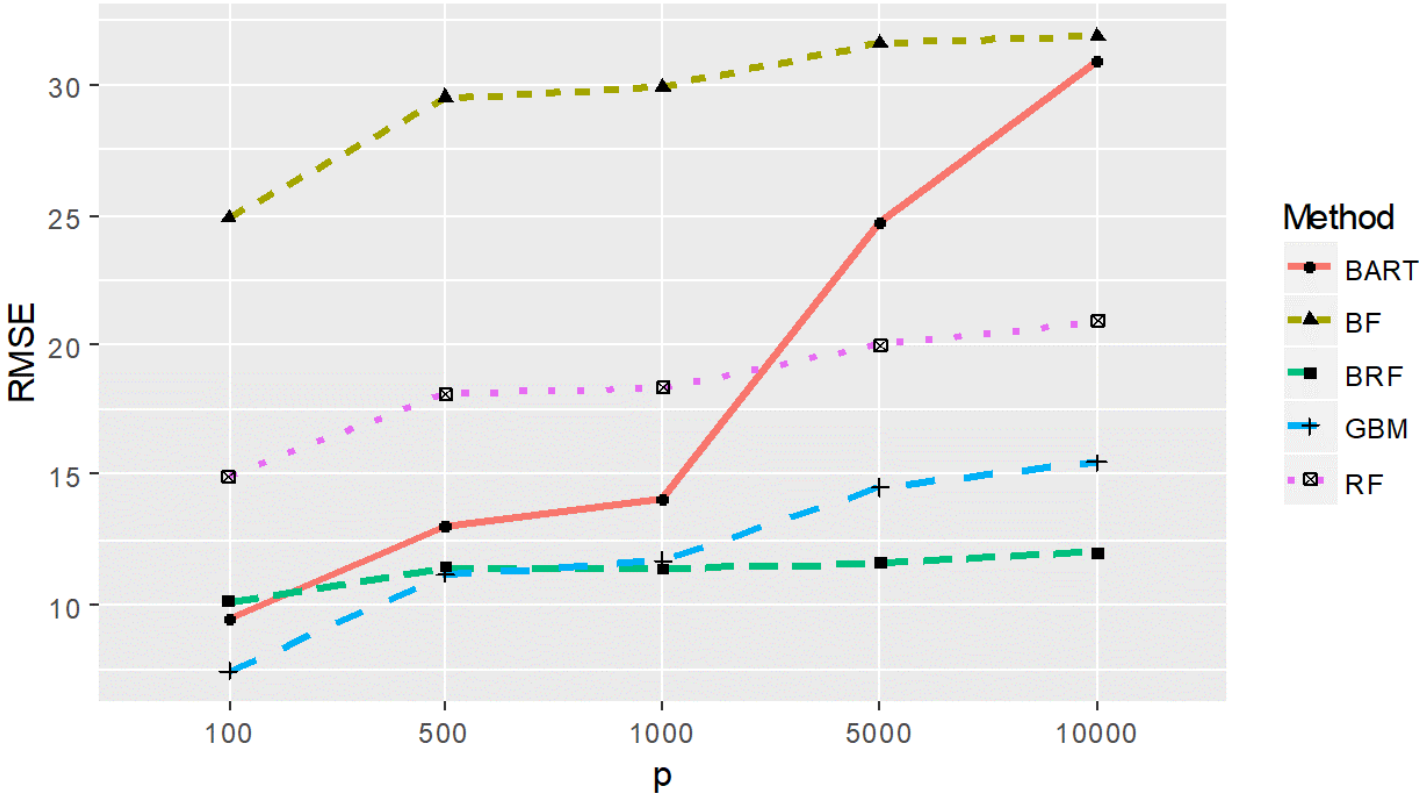

Figure: RMSE with increasing p at sample size $n = 100$

Conclusion

Concluding Remark

The results observed from the simulation study revealed that Bayesian Random Forest is highly robust to large p small relevant variable issue at a reasonable sample size n when compared with its competitors.

Grant

Appreciation

We will like to appreciate Universiti Tun Hussein Onn (UTHM), Malaysia for supporting this research with grant [*Vot*, *U*607].

References

Barber, D. (2012).

Bayesian Reasoning and Machine Learning.

Cambridge University Press, United Kingdom

Breiman, L. (1996a).

Stacked regressions

Machine Learning 24, 41–64.

Breiman, L. (1996b).

Bagging predictors

Machine Learning 26, 123–140.

References

Breiman, L. (2001).

Random forests

Machine Learning 45, 5–32.

Breiman, L., J., F., R., O., and Stone, C. (1984).

Classification and Regression Trees

Wadsworth

Chipman,H.A. George, E. I. and McCulloch, R. E. (2010).

BART: Bayesian Additive Regression Trees.

Annals Applied Statistics 4, 266–298

Dyer, D. and Pierce, R. L. (1993)

On the choice of the prior distribution in hypergeometric sampling

Communications in Statistics - Theory and Methods 22(8), 2125–2146

Hastie, T., James, G., Witten, D., and Tibshirani, R. (2013)

An introduction to statistical learning

Springer, New York

Thank You