BART:
Bayesian additive regression trees

Hedibert F. Lopes & Paulo Marques
INSPER Institute of Education and Research
São Paulo, Brazil

Most of the notes were kindly provided by Rob McCulloch (Arizona State University), which are based on Chipman, George and McCulloch (2010) BART: Bayesian additive regression trees, *The Annals of Applied Statistics*, 4(1), 266-298.

Outline

Nonlinear multiple regression

- Example of nonlinear function
- `R` package `BayesTree`
- Comparing 42 datasets
- Ensemble methods

A regression tree model

- A coordinate view of $g(x;\theta)$
- The BART model
- BART MCMC
- Connections to other modeling ideas
- Some Distinguishing Feastures of BART
- Example: Friedman simulated exercise
- Example: Drug Discovery
- `motorcycle` dataset

Abstract of the BART paper

We develop a Bayesian "sum-of-trees" model where each tree is constrained by a regularization prior to be a weak learner, and fitting and inference are accomplished via an iterative Bayesian backfitting MCMC algorithm that generates samples from a posterior.

Effectively, BART is a nonparametric Bayesian regression approach which uses dimensionally adaptive random basis elements.

Motivated by ensemble methods in general, and boosting algorithms in particular, BART is defined by a statistical model: a prior and a likelihood.

By keeping track of predictor inclusion frequencies, BART can also be used for model-free variable selection.

Outline

Nonlinear regression

We want to "fit" the fundamental model:

$$y_i = g(x_i; \theta) + \epsilon_i$$

BART is a Markov Monte Carlo Method[1] that draws from

$$g(x; \theta) \mid (x, y)$$

We can then use the draws as our inference for $g(x; \theta)$.

[1] Monte Carlo tools were crucial to popularize Bayesian estimation/inference/tools over the last 30 or so years across a wide range of sciences and industry.

Turning the Bayesian crank

To get the draws, we will have to:

- Put a prior on $g(x;\theta)$.
- Specify a Markov chain whose stationary distribution is $p\left(g(x;\theta) \mid (x,y)\right)$.

Example of nonlinear function

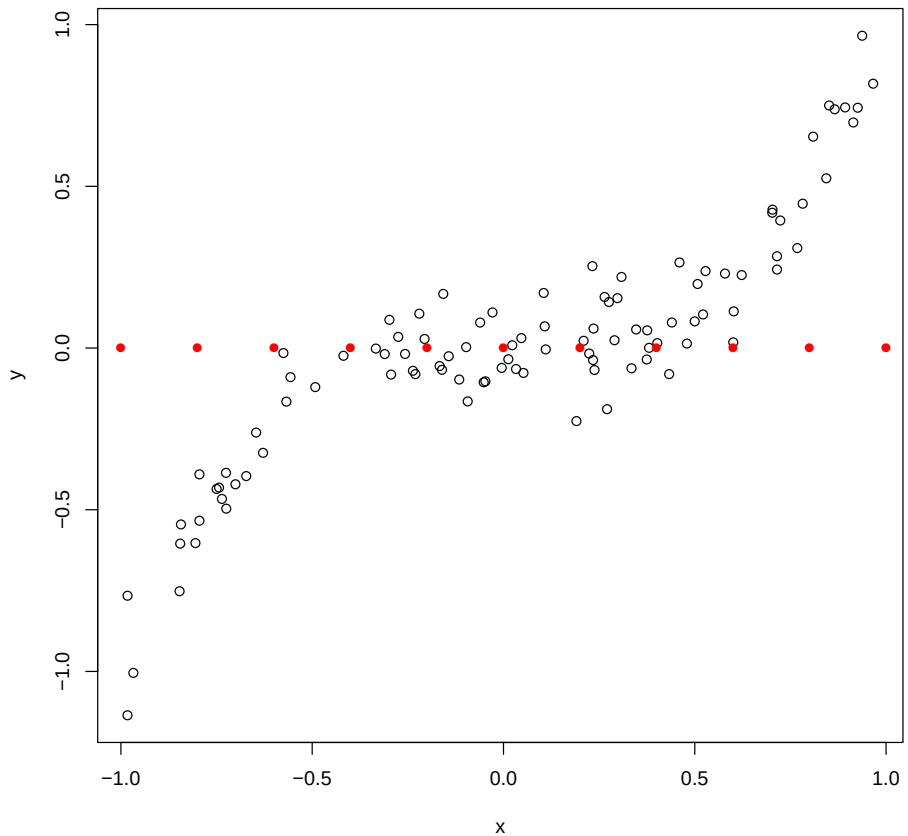
Simulate data from the model:

$$y_i = x_i^3 + \epsilon_i \quad \epsilon_i \sim N(0, \sigma^2) \ \ iid$$

```
-----------------------------------------
n = 100
sigma = 0.1
g = function(x) {x^3}
set.seed(14)
x = sort(2*runif(n)-1)
y = g(x) + sigma*rnorm(n)
xtest = seq(-1,1,by=0.2)
-----------------------------------------
```

`xtest` are *out of sample* x values at which we wish to infer g or make predictions.

```
plot(x,y)
points(xtest,rep(0,length(xtest)),col="red",pch=16)
```



Red is `xtest`

R package BayesTree

```
library(BayesTree)
rb = bart(x,y,xtest)
length(xtest)
[1] 11
dim(rb$yhat.test)
[1] 1000   11
```

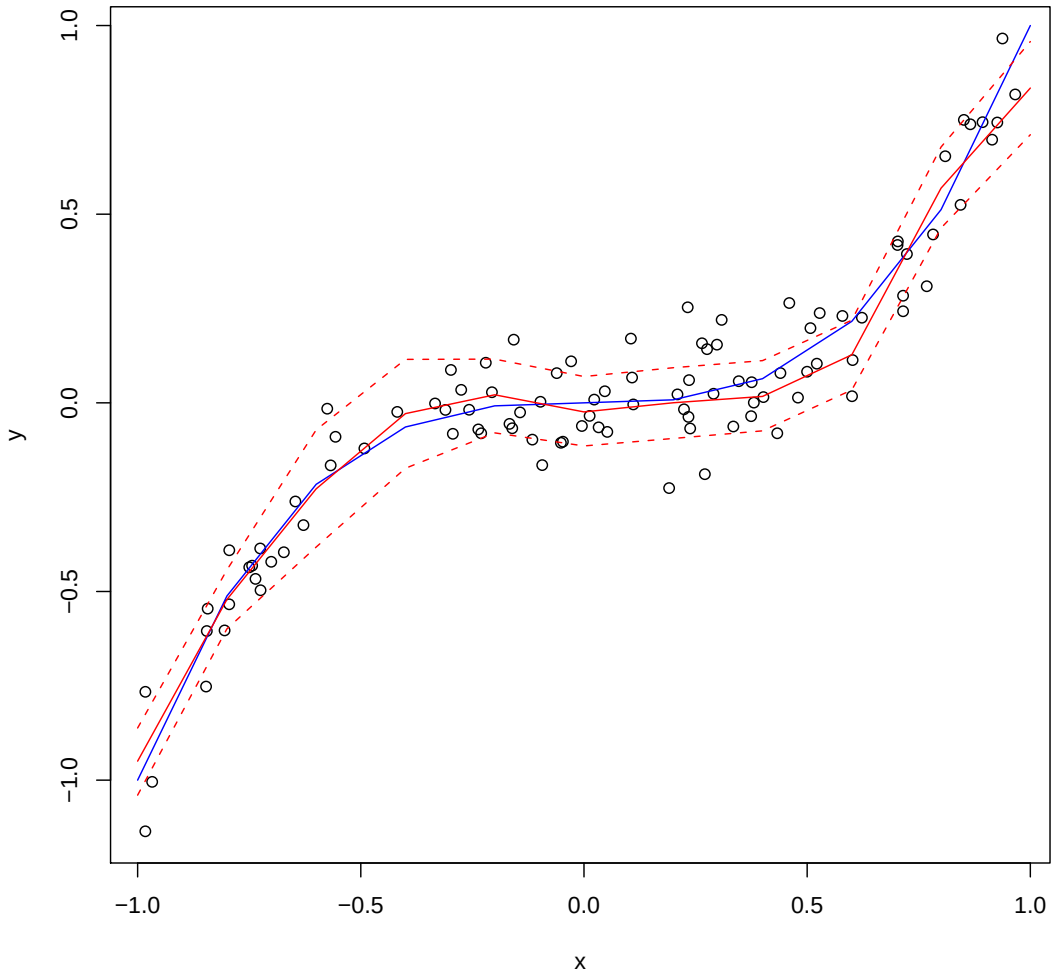
The (i, j) element of `yhat.test` is

the i^{th} draw of g evaluated at the j^{th} value of `xtest`.

1,000 draws of g, each of which is evaluated at 11 `xtest` values.

Fitted model

```
plot(x,y)
lines(xtest,xtest^3,col='blue')
lines(xtest,apply(rb$yhat.test,2,mean),col='red')
qm = apply(rb$yhat.test,2,quantile,probs=c(.05,.95))
lines(xtest,qm[1,],col='red',lty=2)
lines(xtest,qm[2,],col='red',lty=2)
```



Let us get serious: out of sample prediction

- Out of sample predictive comparisons on 42 data sets (thanks to Wei-Yin Loh!!)
- $p = 3 - 65$, $n = 100 - 7,000$.
- for each data set 20 random splits into 5/6 train and 1/6 test
- use 5-fold cross-validation on train to pick hyperparameters (except BART-default!)
- gives $20 \times 42 = 840$ out-of-sample predictions, for each prediction, divide rmse of different methods by the smallest

Competitors

- Linear regression with L1 regularization - Efron et al. (2004)
- Gradient boosting - Friedman (2001)
  Implemented as `gbm` in R by Ridgeway (2004)
- Random forests - Breiman (2001)
  Implemented as `randomforest` in R
- Neural networks with one layer of hidden units
  Implemented as `nnet` in R by Venables and Ripley (2002)

These competitors, like BART, are black box predictors.

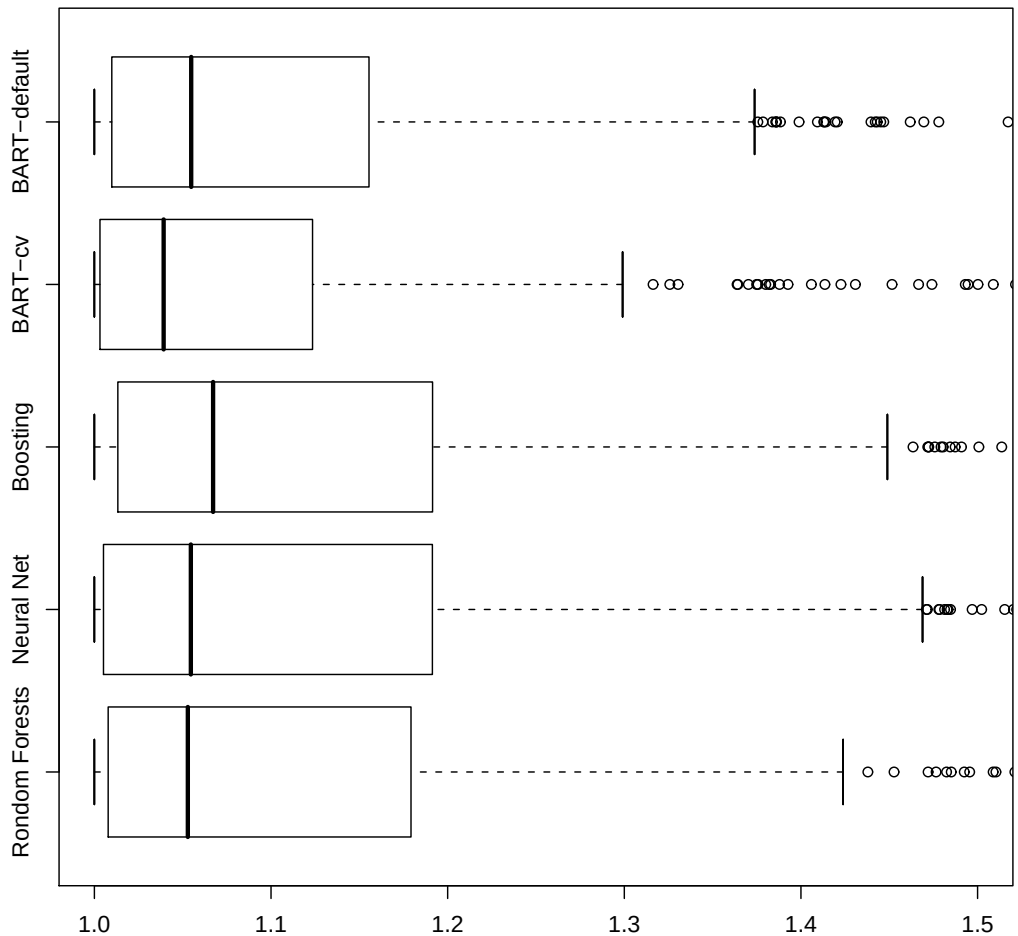
Trees, Bayesian CART[2] and Bayesian treed regression[3] models were not considered, since they tend to sacrifice predictive performance for interpretability.

With the exception of BART-default (which requires no tuning), the operational parameters of every method were chosen via 5-fold cross-validation within each training set.

[2] Chipman, George and McCulloch (1998)
[3] Chipman, George and McCulloch (2002)

Comparison

- Each boxplots represents 840 predictions for a method
- 1.2 means you are 20% worse than the best
- BART-cv best
- BART-default (use default prior) does amazingly well!!



Relative RMSE

TABLE 3
(50%, 75%) quantiles of relative RMSE values for each method across the 840 test/train splits

Method	**(50%, 75%)**
Lasso	(1.196, 1.762)
Boosting	(1.068, 1.189)
Neural net	(1.055, 1.195)
Random forest	(1.053, 1.181)
BART-default	(1.055, 1.164)
BART-cv	(1.037, 1.117)

Relative RMSE > 1.5

- Lasso: 29.5%
- Random forests: 16.2%
- Neural net: 9.0%
- Boosting: 13.6%
- BART-cv: 9.0%
- BART-default: 11.8%

Ensemble methods

Various methods which combine a set of tree models, so called ensemble methods, have attracted much attention, each of which use different techniques to fit a linear combination of trees.

- Bagging (Breiman, 1996)
- Random forests (Breiman, 2001)
- Boosting (Friedman, 2001)
- Bayesian model averaging (Chipman, George and McCulloch, 1998)

Bagging and random forests use randomization to create a large number of independent trees, and then reduce prediction variance by averaging predictions across the trees.

Boosting fits a sequence of single trees, using each tree to fit data variation not explained by earlier trees in the sequence.

Bayesian model averaging (BMA) applied to the posterior arising from a Bayesian single-tree model.

Key references

Breiman (1996) Bagging predictors
Machine Learning, 26, 123-140.

Hastie and Tibshirani (2000) Bayesian Backfitting
Statistical Science, 15(3), 196-223.

Friedman (2001) Greedy function approximation: A gradient boosting machine
Annals of Statistics, 29, 1189-1232.

Breiman (2001) Random forests
Machine Learning, 45, 5-32.

Chipman, George and McCulloch (1998) Bayesian CART model search
Journal of the American Statistical Association, 93, 935-960.

Efron, Hastie, Johnstone and Tibshirani (2004) Least angle regression
Annals of Statistics, 32, 407-499.

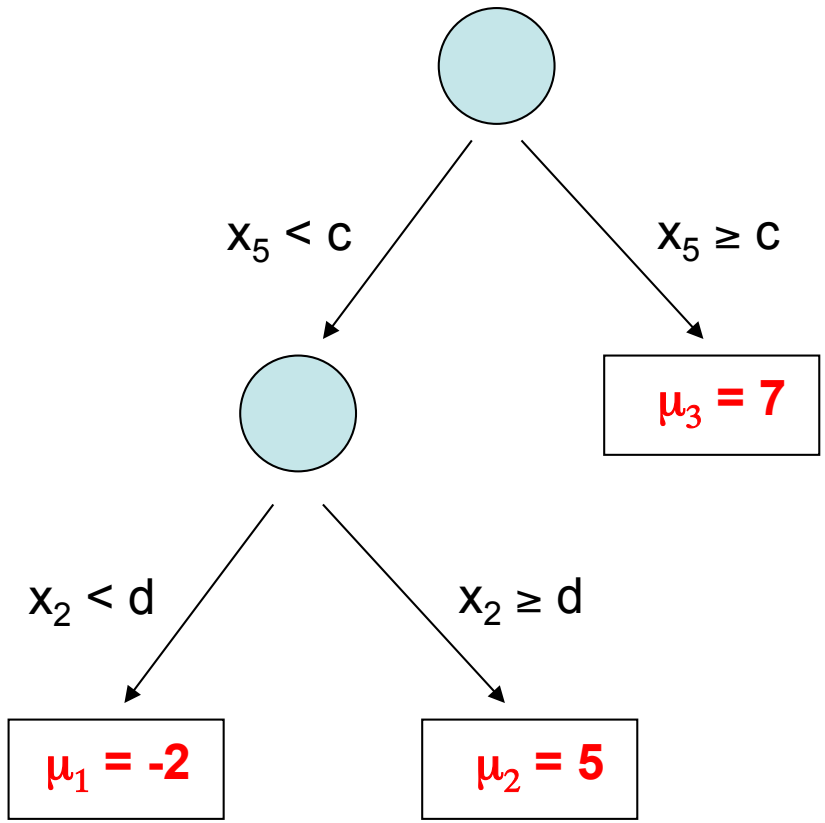
Outline

A regression tree model

Let T denote the tree structure including the decision rules.

Let $M = \{\mu_1, \mu_2, \ldots, \mu_b\}$ denote the set of bottom node μ's.

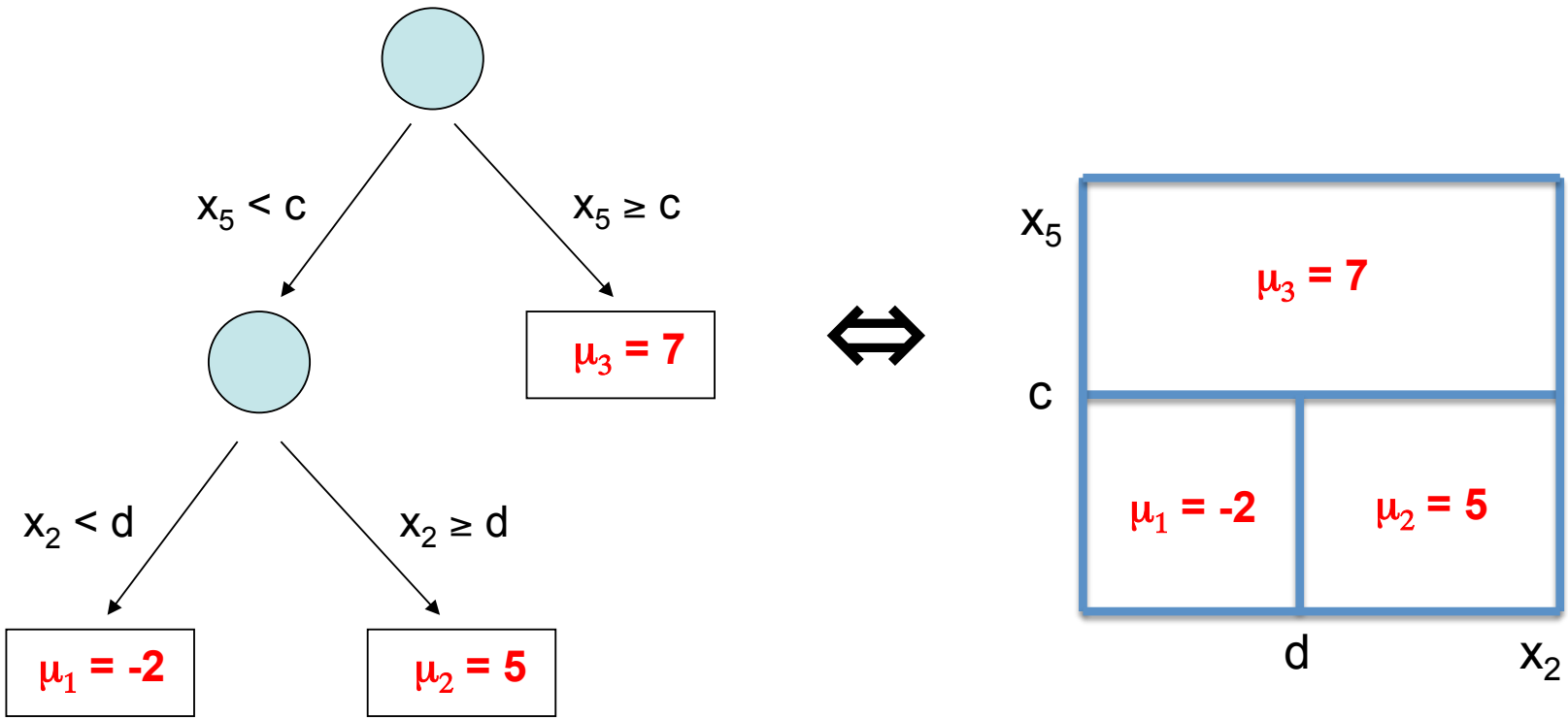
Let $g(x; \theta)$, $\theta = (T, M)$ be a regression tree function that assigns a μ value to x.



A single tree model:

$$y_i = g(x_i; \theta) + \epsilon_i.$$

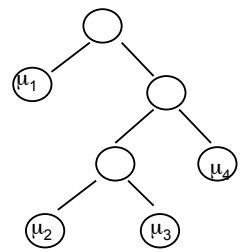
A coordinate view of $g(x;\theta)$



Easy to see that $g(x;\theta)$ is just a step function.

The BART model

$$Y = g(x;T_1,M_1) + g(x;T_2,M_2) + \ldots + g(x;T_m,M_m) + \sigma z, \quad z \sim N(0,1)$$



$m = 200, 1000, \ldots,$ big, $\ldots$

$f(x \mid \cdot)$ is the sum of all the corresponding μ's at each bottom node.

Such a model combines additive and interaction effects.

Complete the model with a regularization prior

The prior of the BART model can be written as

$$\pi(\theta) = \pi((T_1, M_1), (T_2, M_2), \ldots, (T_m, M_m), \sigma).$$

π wants:

- Each T small.
- Each μ small.
- "nice" σ (smaller than least squares estimate).

We refer to π as a regularization prior because it keeps the overall fit small.

In addition, it keeps the contribution of each $g(x; T_i, M_i)$ model component small.

Consider the prior on μ.
Let θ denote all the parameters.

$$f(x \mid \theta) = \mu_1 + \mu_2 + \cdots \mu_m.$$

Let $\mu_i \sim N(0, \sigma_\mu^2)$, iid.

$$f(x \mid \theta) \sim N(0, m\,\sigma_\mu^2).$$

In practice we often, unabashadly, use the data by first centering and then choosing σ_μ so that

$$f(x \mid \theta) \in (y_{min}, y_{max})$$

with high probability:

$$\sigma_\mu^2 \propto \frac{1}{m}.$$

BART MCMC

The model/prior is described by

$$Y = g(x;T_1,M_1) + ... + g(x;T_m,M_m) + \sigma z$$
plus
$$\pi((T_1,M_1),....(T_m,M_m),\sigma)$$

First, it is a "simple" Gibbs sampler:

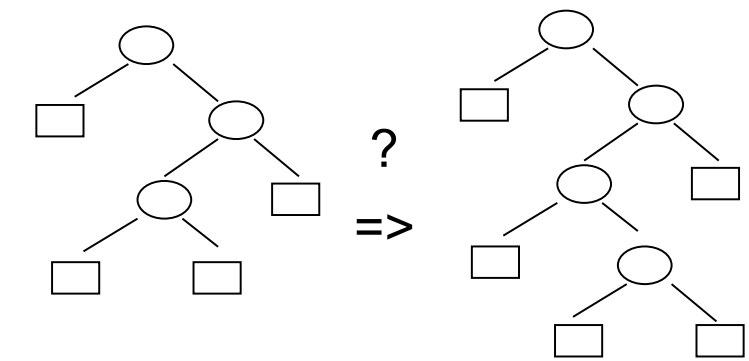
$$\begin{aligned} (T_i, M_i) \quad &| \quad (T_1, M_1, \ldots, T_{i-1}, M_{i-1}, T_{i+1}, M_{i+1}, \ldots, T_m, M_m, \sigma) \\ \sigma \quad &| \quad (T_1, M_1, \ldots, \ldots, T_m, M_m) \end{aligned}$$

To draw $(T_i, M_i) \,|\, \cdot$ we subract the contributions of the other trees from both sides to get a simple one-tree model.

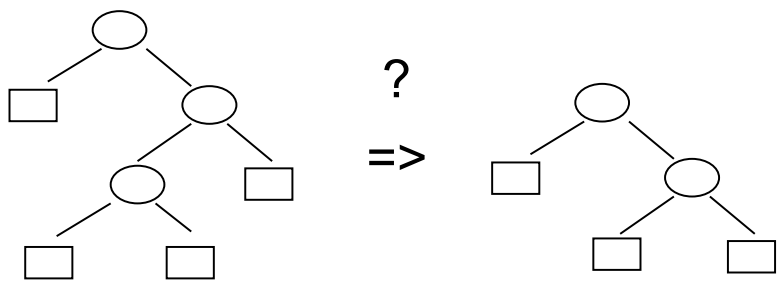
We integrate out M to draw T and then draw $M \,|\, T$.

Birth-death moves

To draw T we use a Metropolis-Hastings with Gibbs step.
We use various moves, but the key is a "birth-death" step.

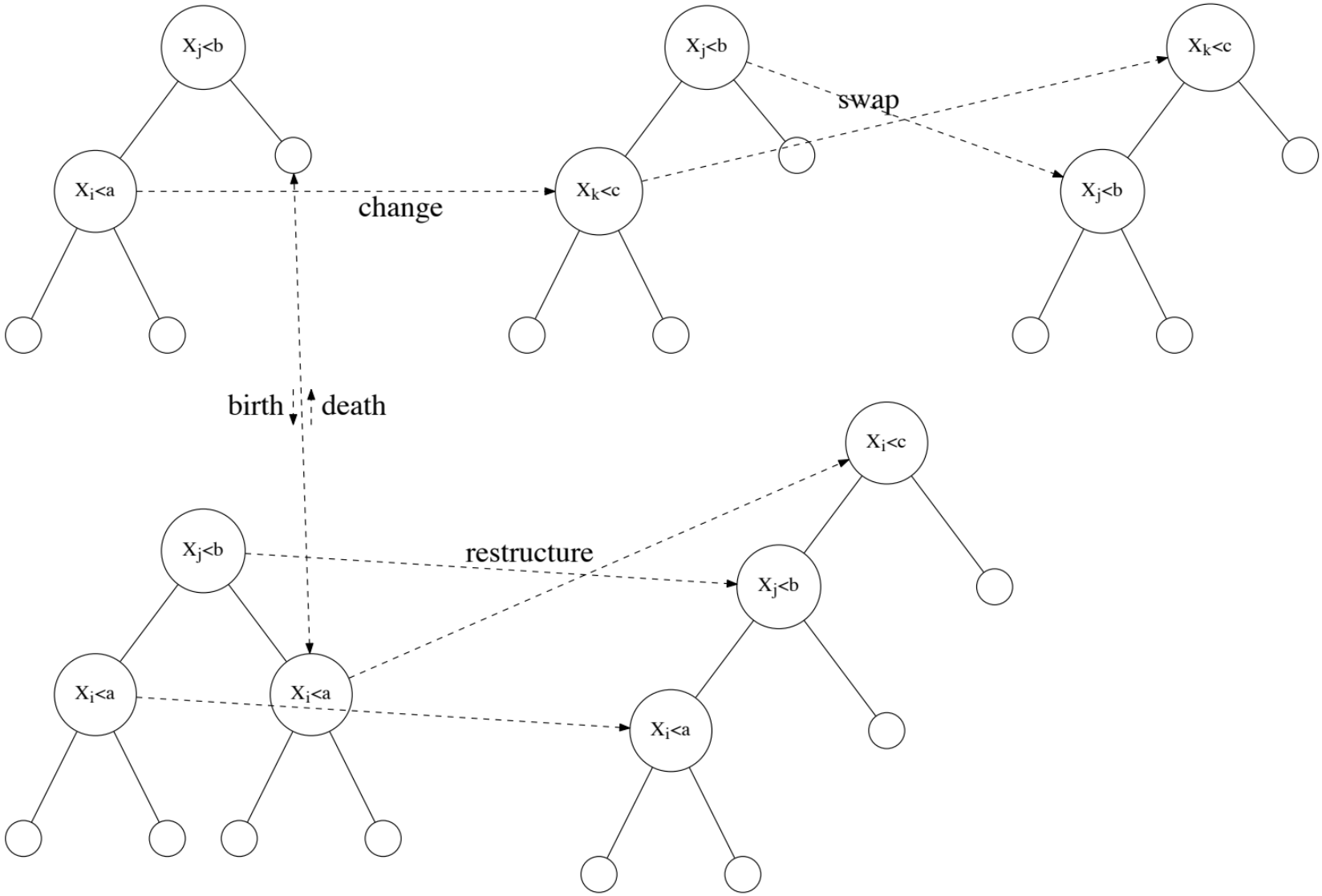


propose a more complex tree



propose a simpler tree

Tree moves[4]



[4] http://www.matthewpratola.com/wp-content/uploads/2017/11/stat8810-slides13.pdf

Connections to other modeling ideas

$$Y = g(x;T_1,M_1) + \ldots + g(x;T_m,M_m) + \sigma z$$
plus
$$\pi((T_1,M_1),\ldots.(T_m,M_m),\sigma)$$

Bayesian nonparametrics:

- Lots of parameters to make model flexible.
- A strong prior to shrink towards a simple structure.
- BART shrinks towards additive models with some interaction.

Dynamic random basis:

- $g(x; T_1, M_1), g(x; T_2, M_2), \ldots, g(x; T_m, M_m)$ are dimensionally adaptive.

Gradient boosting:

- Overall fit becomes the cumulative effort of many weak learners.

Some Distinguishing Feastures of BART

$$Y = g(x;T_1,M_1) + ... + g(x;T_m,M_m) + \sigma z$$
plus
$$\pi((T_1,M_1),....(T_m,M_m),\sigma)$$

- BART is NOT Bayesian model averaging of single tree model.
- Unlike boosting and random forests, BART updates a set of m trees over and over, *stochastic search*.
- Choose m large for flexible estimation and prediction.
- Choose m smaller for variable selection
  - fewer trees forces the x's to compete for entry.

Example: Friedman simulated exercise

For $i = 1, \ldots, n = 100$,

$$y_i = g(x_i) + \epsilon_i, \quad \epsilon_i \sim N(0, 1),$$

where

$$g(x_i) = 10 \sin(\pi x_{i1} x_{i2}) + 20(x_{i3} - 0.5)^2 + 10 x_{i4} + 5 x_{i5}$$

Add 5 irrelevant $x_{i6}, \ldots, x_{i,10}$ $(p = 10)$.

$x_{ij} \sim \text{uniform}(0, 1)$.

$\hat{g}(x)$ is the posterior mean.

Root MSE

Compute out of sample RMSE using 1,000 simulated $x \in R^{10}$.

$$\text{RMSE} = \sqrt{\frac{1}{1000}\sum_{i=1}^{1000}(g(x_i) - \hat{g}(x_i))^2}$$

Method	average RMSE	se(RMSE)
Random Forests	2.655	0.025
Linear Regression	2.618	0.016
Neural Nets	2.156	0.025
Boosting	2.013	0.024
MARS	2.003	0.060
BART-cv	1.787	0.021
BART-default	1.759	0.019

Details about competing schemes

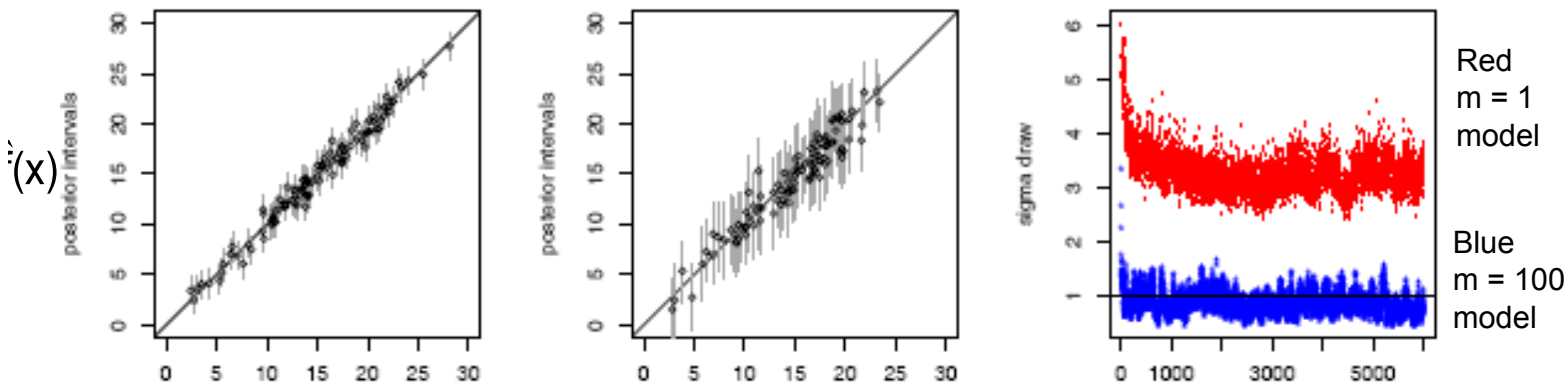
Method	Parameter	Values considered
Boosting	# boosting iterations	`n.trees`= $1, 2, \ldots, 2000$
	Shrinkage (multiplier of each tree added)	`shrinkage`= 0.01, 0.05, 0.10, 0.25
	Max depth permitted for each tree	`interaction.depth`= 1,2,3,4
Neural Nets	# hidden units	`size`= 10, 15, 20, 25, 30
	Decay (penalty coef on sum-squared weights)	`decay`= 0.50, 1, 1.5, 2, 2.5
	(Max # optimizer iterations, # restarts)	fixed at `maxit`= 1000 and 5
Random Forests	# of trees	`ntree`= 200, 500, 1000
	# variables sampled to grow each node	`mtry`= 3, 5, 7, 10
MARS	GCV penalty coefficient	`gcv`= 1, 2, ..., 8
BART -cv	Sigma prior: (ν, q) combinations	(3,0.90), (3,0.99), (10,0.75)
	μ Prior: k value for σ_μ	1, 1.5, 2, 2.5, 3
	(# trees m, iterations used, burn-in iterations)	fixed at (200, 1000,500)
BART -default	Sigma prior: (ν, q) combinations	fixed at (3,0.90)
	μ Prior: k value for σ_μ	fixed at 2
	(# trees m, iterations used, burn-in iterations)	fixed at (200, 1000,500)

Table 1: Operational parameters for the various competing models. `Names` in last column indicate parameter names in R.

Results for one draw

95% posterior intervals vs true f(x)

σ draws



in-sample f(x)

out-of-sample f(x)

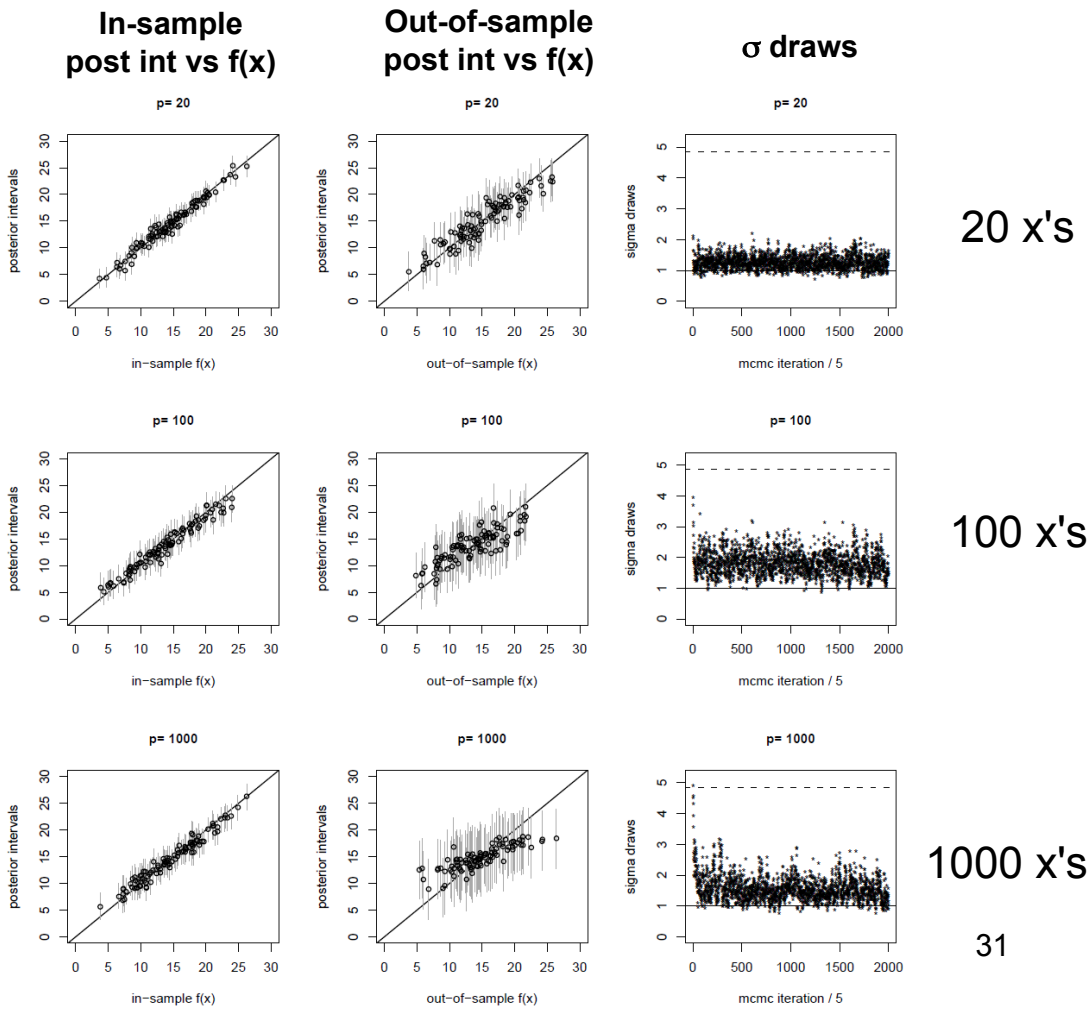
MCMC iteration

Frequentist coverage rates of 90% posterior intervals:
in sample: 87%
out of sample: 93 %.

Adding many useless predictors

Added many useless x's to Friedman's example

With only 100 observations on y and *1000 x's, BART yielded "reasonable" results !!!!*

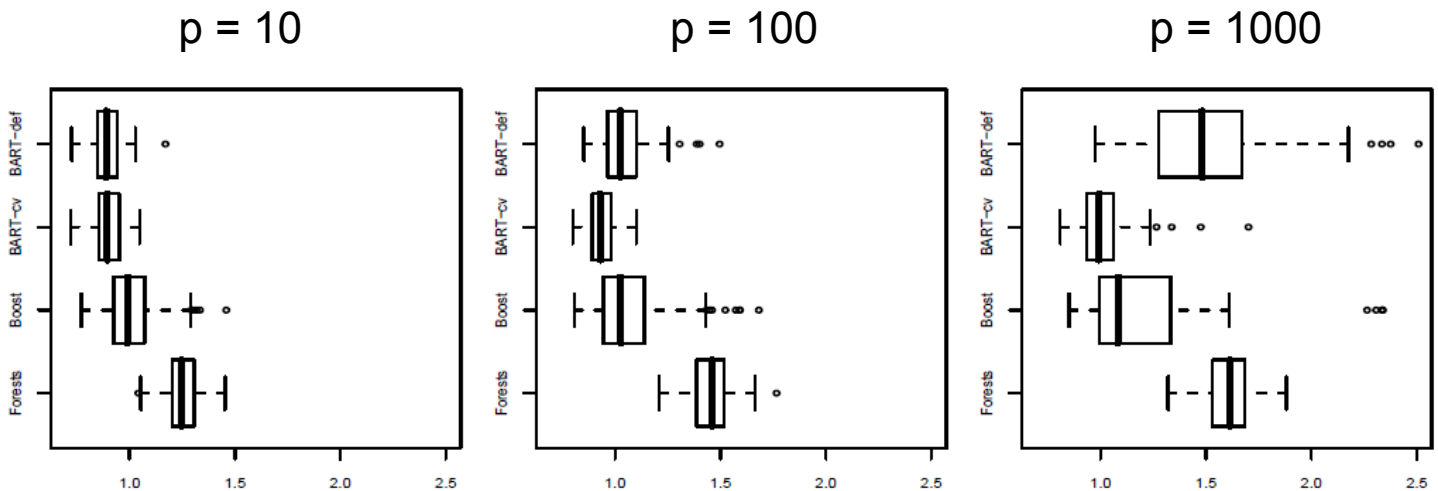


Big p, small n

$n = 100$.

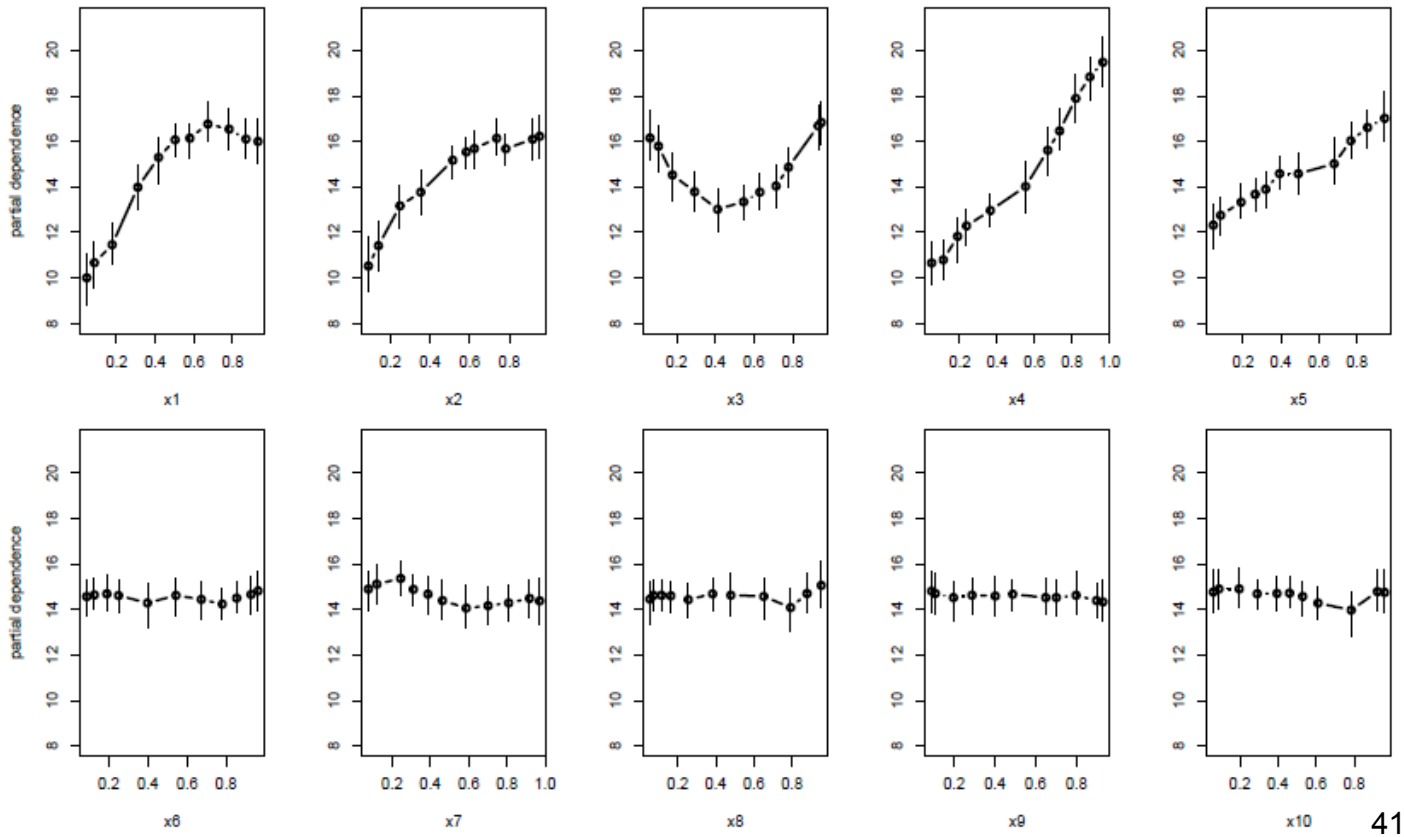
Compare BART-default,BART-cv,boosting, random forests.

Out of sample RMSE.



Partial Dependence plot

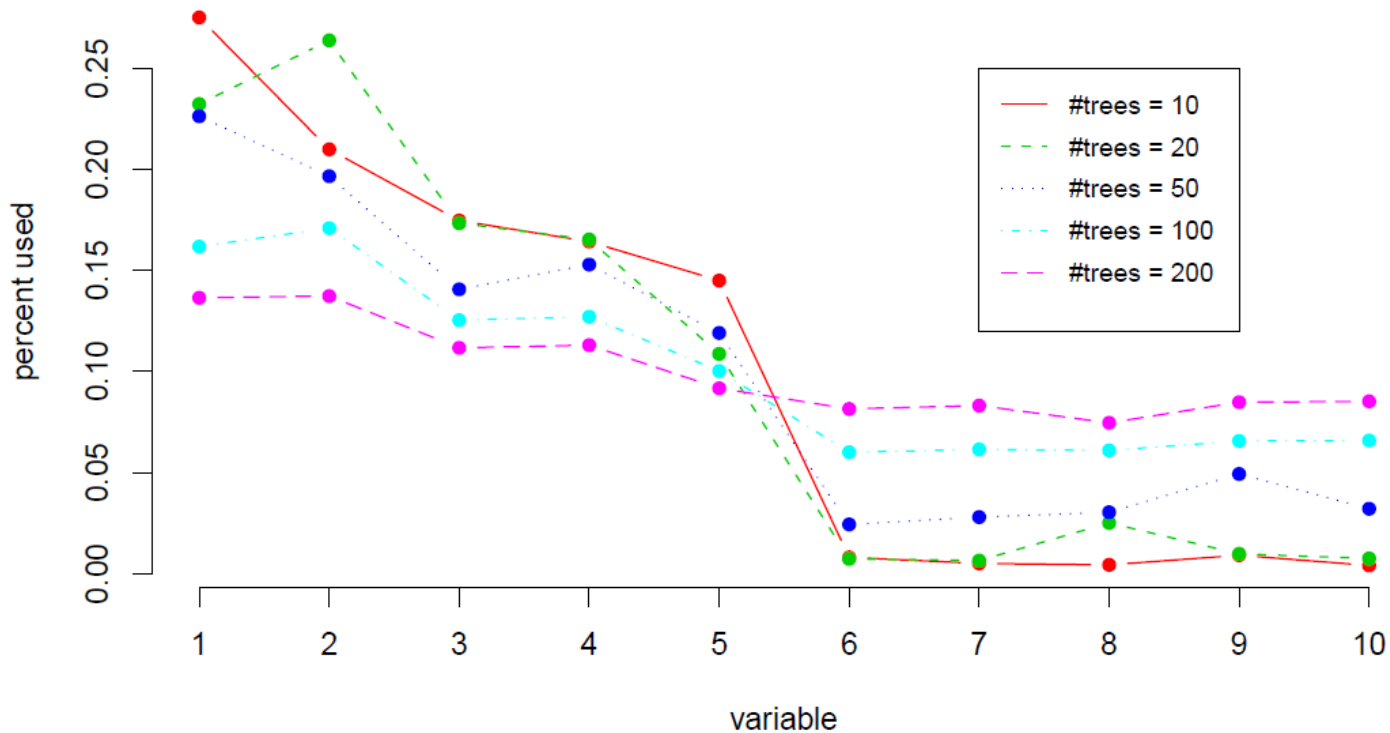
Vary one x and average out the others.



41

Variable selection

Frequency with which a variable is used.



Example: Drug Discovery

Goal: To predict the "activity" of a compound against a biological target.

That is: $y = 1$ means drug worked (compound active), 0 means it does not.

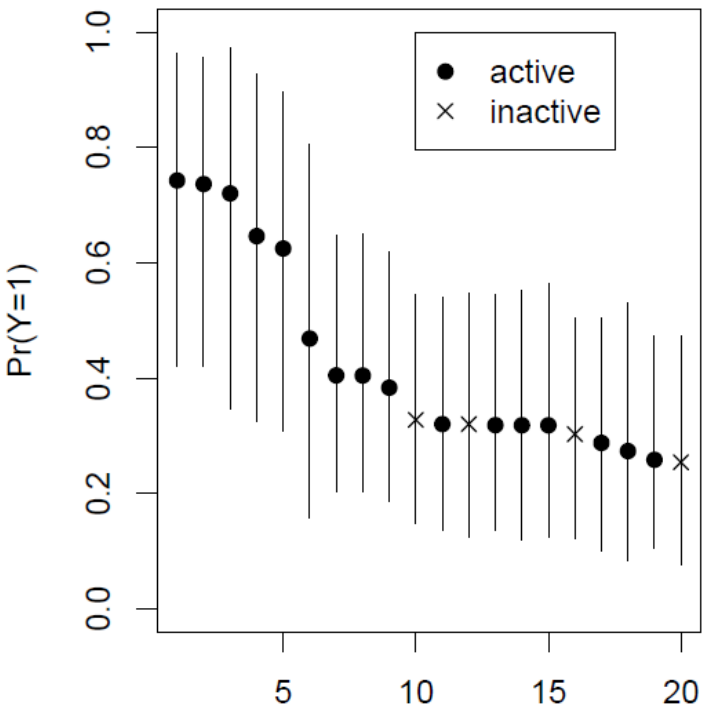
Easy to extend BART to binary y using Albert & Chib.

$n = 29,3744 \rightarrow$ 14,687 train, 14,687 test.
$p = 266$ characterizations of the compound's molecular structure.

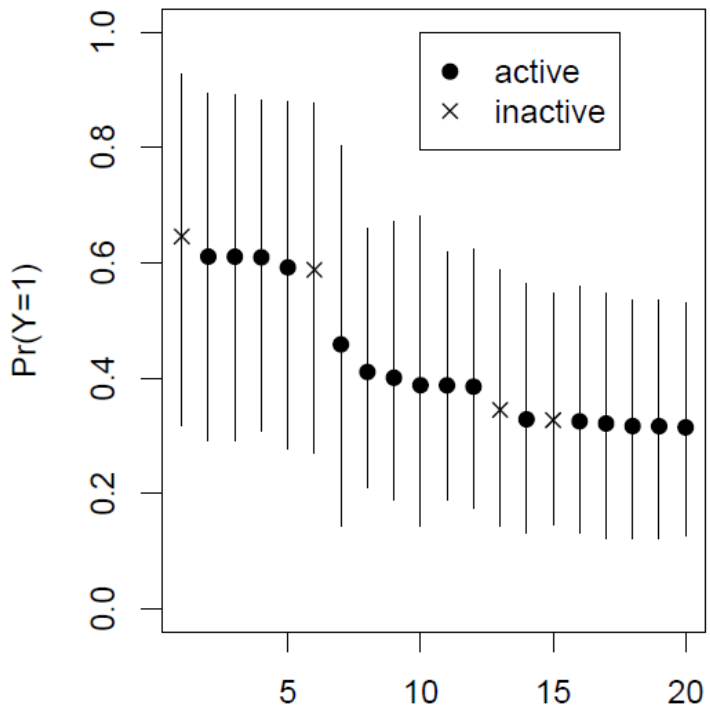
Again, out-of-sample prediction competitive with other methods, compared to neural-nets, boosting, random forests, support vector machines.

20 compounds with highest $Pr(Y = 1 \mid x)$ estimate.
90% posterior intervals for $Pr(Y = 1 \mid x)$.

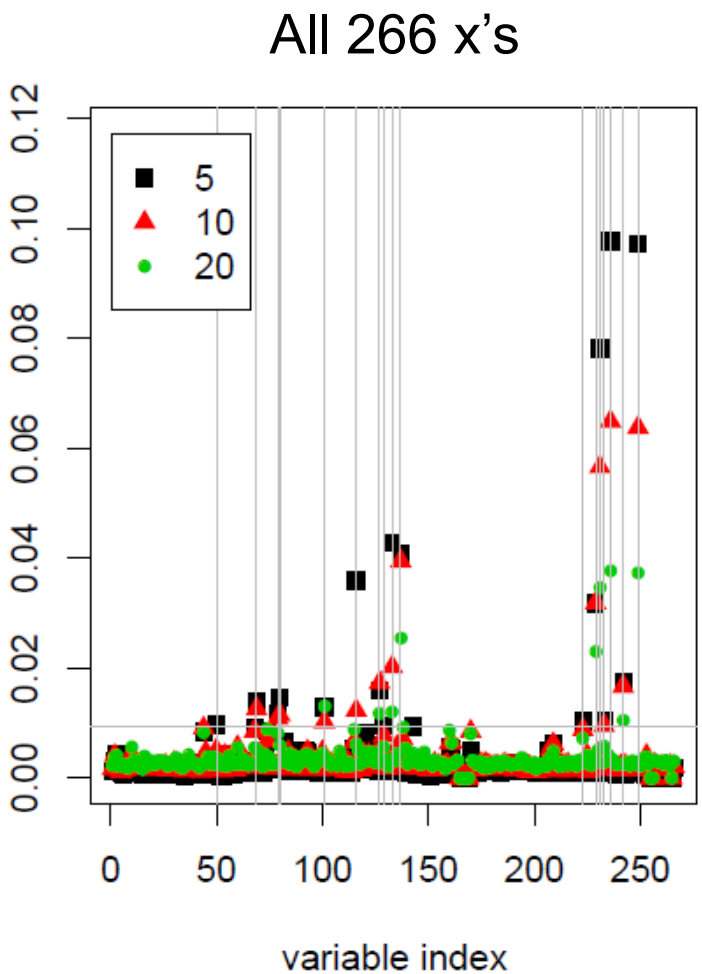
In-sample

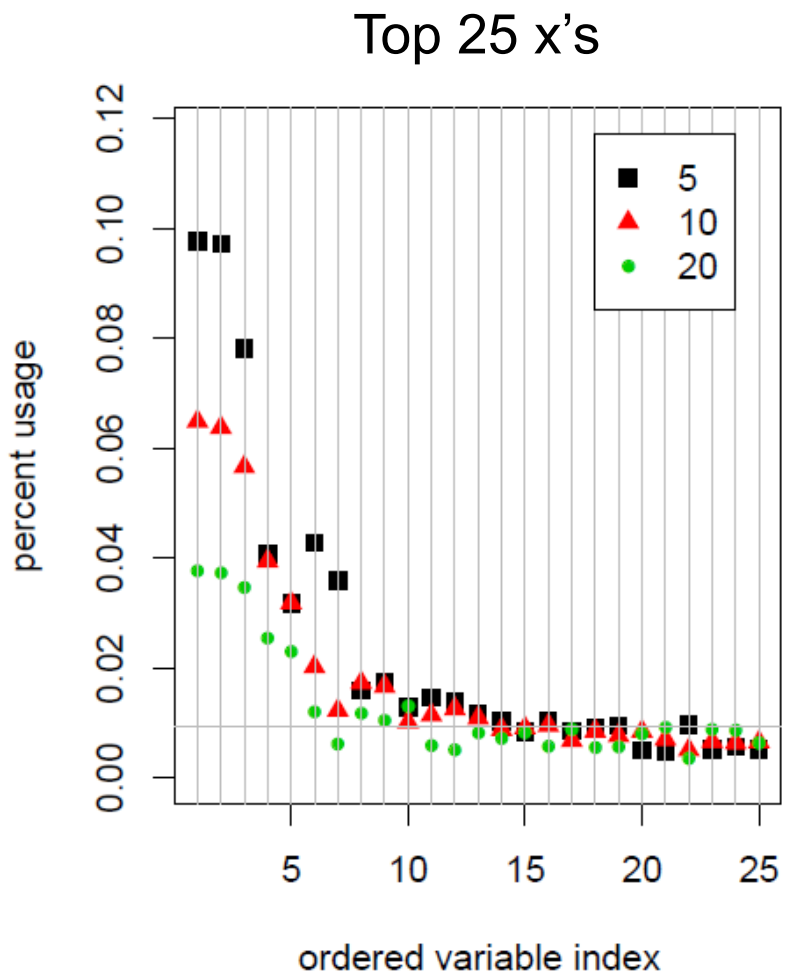


Out-of-Sample

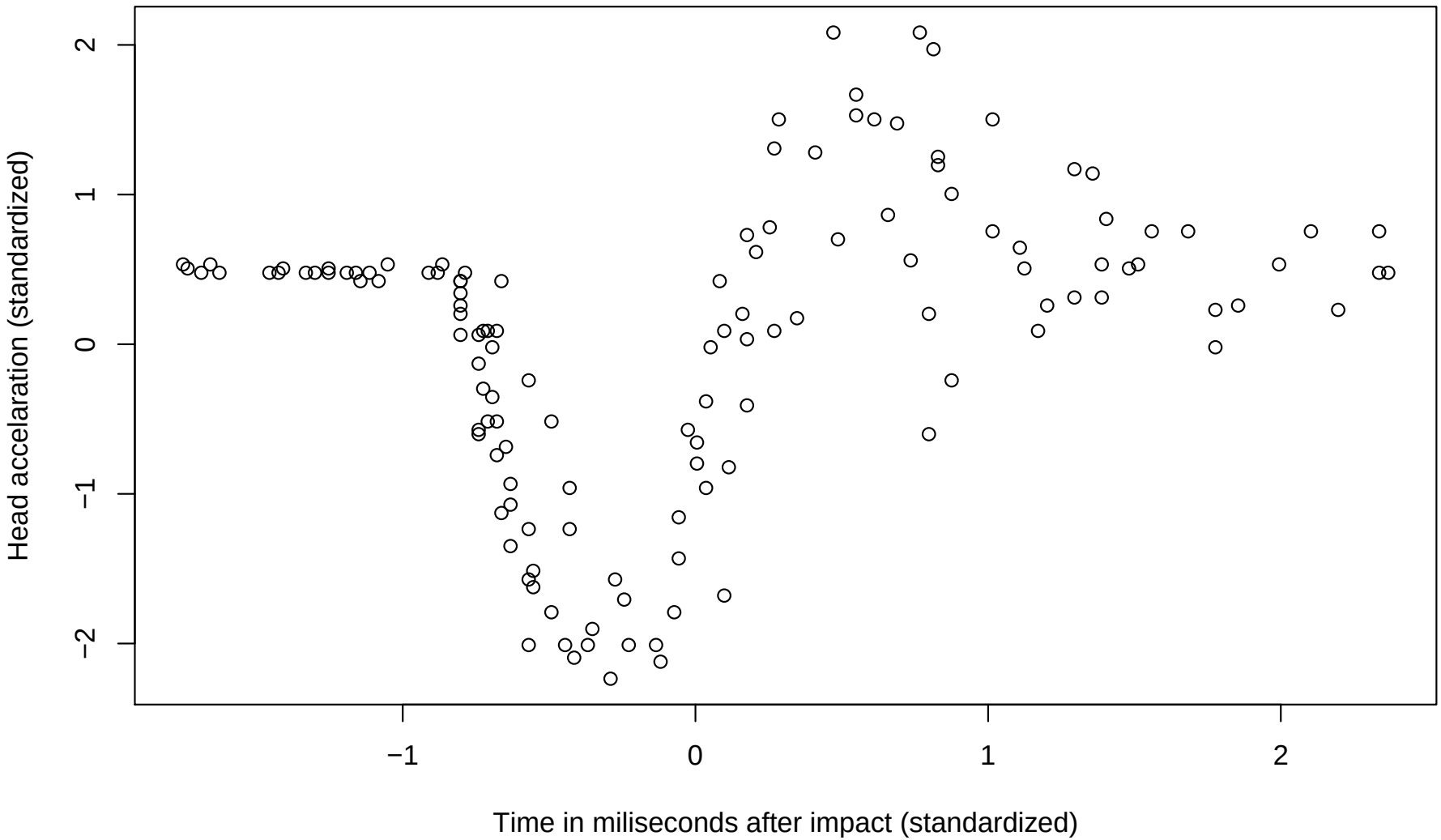


Variable selection





motorcycle dataset (revisited)



Smooth spline

The goal is to find $g(\cdot)$ that minimizes

$$\sum_{i=1}^{n}(y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt$$

for tuning parameter $\lambda > 0$.

The basis functions for a global cubic polynomial are $B_i(x) = x^{i-1}$ for $i = 1, 2, 3, 4$, so

$$g(x) = \sum_{j=1}^{4} \beta_j B_j(x)$$

Splines are piecewise cubic polynomials: $B_1(x) = 1$, $B_2(x) = x$ and

$$B_{2+i}(x) = \frac{(x - x_i)_+^3 - (x - x_n)_+^3}{x_n - x_i} - \frac{(x - x_{n-1})_+^3 - (x - x_n)_+^3}{x_n - x_{n-1}}$$

R code

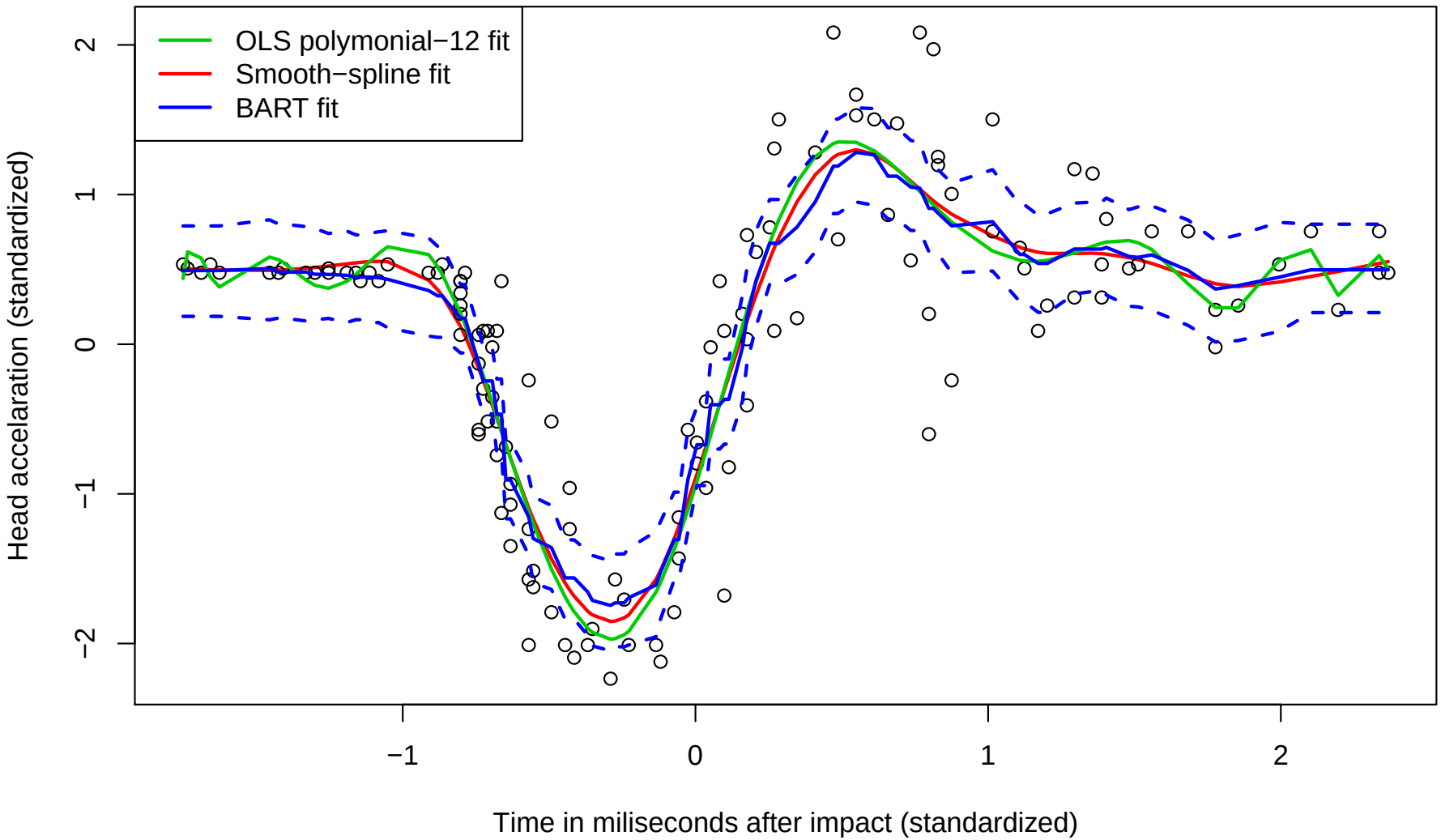
```
install.packages("BART")
library(MASS)
library(BART)
xt = mcycle$times[1:132]
yt = mcycle$accel[1:132]
xt = (xt-mean(xt))/sqrt(var(xt))
yt = (yt-mean(yt))/sqrt(var(yt))

d=12
xx = NULL
for (i in 1:d)
  xx = as.matrix(cbind(xx,xt^i))
xx = (xx - matrix(apply(xx,2,mean),n,d,byrow=TRUE))%*%diag(sqrt(1/apply(xx,2,var)))

# OLS, smooth spline and BART fits
linear.fit = lm(yt~xx-1)
fit = smooth.spline(xt,yt)
bart.fit   = wbart(xt,yt)
bart.q     = t(apply(bart.fit$yhat.train,2,quantile,c(0.05,0.5,0.95)))

plot(fit,xlab="Time in miliseconds after impact (standardized)",
     ylab="Head accelaration (standardized)",type="l",lwd=2,col=2,
     xlim=range(xt),ylim=range(yt))
points(xt,yt)
lines(xt,linear.fit$fit,col=3,lwd=2)
lines(xt,bart.q[,2],col=4,lwd=2)
lines(xt,bart.q[,1],col=4,lwd=2,lty=2)
lines(xt,bart.q[,3],col=4,lwd=2,lty=2)
legend("topleft",legend=c("OLS polymonial-12 fit","Smooth-spline fit","BART fit"),
      col=c(3,2,4),lwd=2,lty=1)
```

lm, smooth.spline and wbart in action

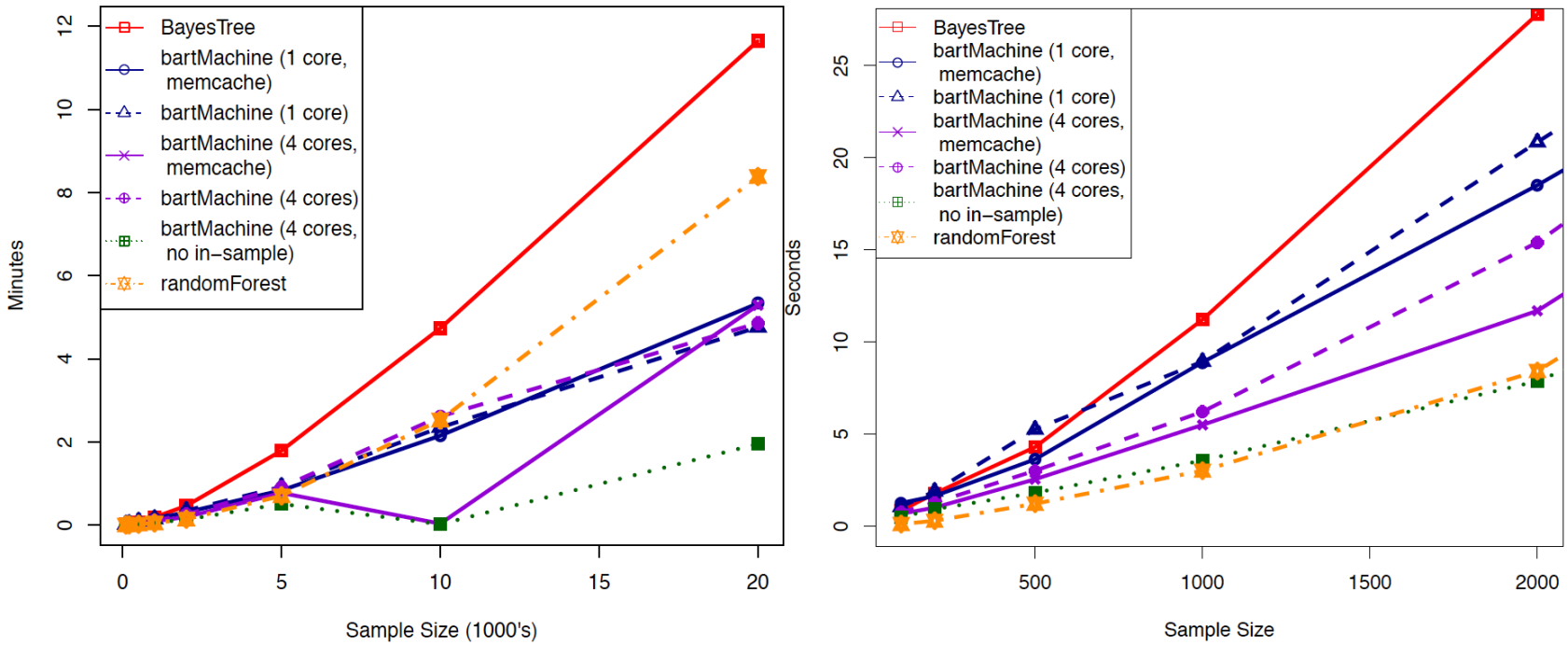


BayesTree versus bartMachine

Feature	**bartMachine**	**BayesTree**
Implementation language	Java	C++
External predict function	Yes	No
Model persistence across sessions	Yes	No
Parallelization	Yes	No
Native missing data mechanism	Yes	No
Built-in cross-validation	Yes	No
Variable importance	Statistical tests	Exploratory
Tree proposal types	3 types	4 types
Partial dependence plots	Yes	Yes
Convergence plots	Assess trees and σ^2	Assess σ^2
Model diagnostics	Yes	No
Incorporation into larger model	No	Through **dbarts**

Table 1: Comparison of features between **bartMachine** and **BayesTree**.

BayesTree versus bartMachine



(a) Large sample sizes

(b) Small sample sizes

Figure 1: Model creation times as a function of sample size for a number of settings of **bartMachine**, **BayesTree** and **randomForest**. Simulations were run on a quad-core 3.4GHz Intel i5 desktop with 24GB of RAM running the Windows 7 64bit operating system.

References

1. Chipman, George and McCulloch (2010) BART: Bayesian Additive Regression Trees.
   The Annals of Applied Statistics, 4(1), 266-298.

2. Taddy, Gramacy and Polson (2011) Dynamic Trees for Learning and Design.
   Journal of the American Statistical Association, 106(493), 109-123.

3. Pratola, Chipman, Higdon, McCulloch and Rust (2014) Parallel BART.
   Journal of Computational and Graphical Statistics, 23, 830-852.

4. Lakshminarayanan, Roy and Teh (2015) Particle Gibbs for BART.
   Proceedings of the 18th Conference on Artificial Intelligence and Statistics.

5. Kapelner and Bleich (2016) `bartMachine`: machine learning with BART.
   Journal of Statistical Software, 70(4).

6. Pratola (2016) Efficient Metropolis-Hastings Proposal Mechanisms for BART models.
   Bayesian Analysis, 11(3), 885-911.

7. Hernández, Raftery, Pennington and Parnell (2017) BART using BMA.
   Statistics and Computing.

8. Linero (2017) Bayesian Regression Trees for High Dimensional Prediction and Variable Selection.
   Journal of the American Statistical Association.

9. Pratola, Chipman, George and McCulloch (2017) Heteroscedastic BART Using Multiplicative Regression Trees.