Pre-Modeling Via BART

Ed George, University of Pennsylvania
(joint work with H. Chipman and R. McCulloch)

Tenth Annual Winter Workshop
Bayesian Model Selection and Objective Methods

Department of Statistics
University of Florida
January 11-12, 2008

A General Nonparametric Regression Setup

- Data: n observations on y and $x = (x_1,...,x_p)$
- Suppose: $y = f(x) + \varepsilon$, ε symmetric around 0
- Unknowns: f and the distribution of ε

For this model free setup, BART can help us to:

- estimate $f(x) = E(y|x)$
- obtain prediction intervals for future y
- estimate the effect of a particular x_j
- select an informative subset of $x_1,...,x_p$

  (making no assumptions about f)

Remark: In what follows we will assume $\varepsilon \sim N(0, \sigma^2)$ for simplicity, but extension to a general DP process normal mixture model for ε works just fine.

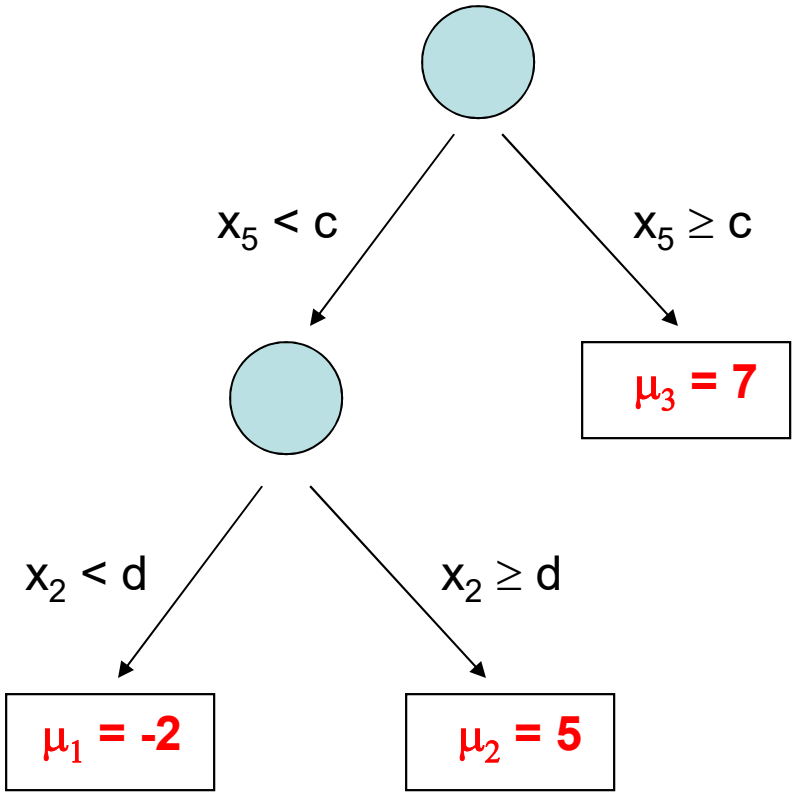
How Does BART Work?

BART (= Bayesian Additive Regression Trees) is composed of many single tree models

Let g(x;T,M) be a function which assigns a μ value to x where:

T denotes the tree structure including the decision rules

M = $\{\mu_1, \mu_2, \ldots \mu_b\}$ denotes the set of terminal node μ's.



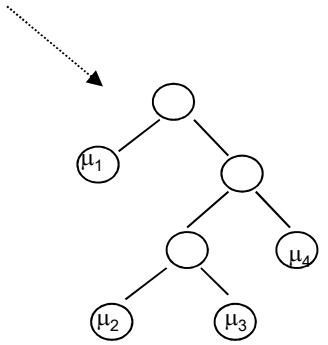
A Single Tree Model: $y = g(x;T,M) + \sigma z, \quad z \sim N(0,1)$

An Additive Multiple Tree Model

Let (T_1,M_1), (T_2,M_2), …, (T_m,M_m) identify a set of m trees and their μ's.

An Additive Multiple Tree Model:

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



E(y | x) is the sum of all the corresponding μ's at each tree bottom node.

Such a model combines additive and interaction effects.

Completing the BART Model

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

is determined by

$$(T_1,M_1),\ldots,(T_m,M_m),\sigma$$

For m large:

Many, many parameters

$g(x;T_1,M_1)$, $g(x;T_2,M_2)$, ... , $g(x;T_m,M_m)$ is a highly redundant "over-complete basis"

To unleash the potential of this formulation, BART is completed by **adding a regularization prior**

$$\pi((T_1,M_1),\ldots,(T_m,M_m),\sigma)$$

Strongly influential π is used to keep each (T_i, M_i) small

BART Implementation

Because BART is a fully Bayesian specification, information about all the unknowns, namely $\theta = ((T_1,M_1),....(T_m,M_m),\sigma)$, is captured by the posterior

$$\pi(\theta \mid y) \propto p(y \mid \theta)\, \pi(\theta)$$

Thus, to implement BART we need to:

1. Construct the prior $\pi(\theta)$
   - Independent tree generating process on $T_1,..,T_m$
   - Use observed y to properly scale $\pi(\theta \mid T)$

2. Calculate the posterior $\pi(\theta \mid y)$
   - Bayesian backfitting MCMC
   - Interweaving marginalization and regeneration of θ

R package BayesTree available on CRAN

$$y = g(x;T_1,M_1) + ... + g(x;T_m,M_m) + \sigma z, \quad z \sim N(0,1)$$

plus

$$\pi((T_1,M_1),....(T_m,M_m),\sigma)$$

Connections to Other Modeling Ideas

Bayesian Nonparametrics:

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

Dynamic Random Basis:

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

Gradient Boosting:

- Overall fit becomes the cumulative effort of many “weak learners”

$$y = g(x;T_1,M_1) + ... + g(x;T_m,M_m) + \sigma z, \quad z \sim N(0,1)$$

plus

$$\pi((T_1,M_1),....(T_m,M_m),\sigma)$$

Some Distinguishing Features of BART

BART is NOT obtained by Bayesian model averaging of a single tree model !

Unlike boosting, BART uses a FIXED number of trees m!!

The identification of subsets for variable selection via BART is obtained by observing what happens as m is varied!!

Experimental Comparison on 37 datasets

Out-sample-performance compared for 6 methods

- Neural networks (single layer)
- Random Forests
- Boosting (Friedman's gradient boosting machine)
- Linear regression with lasso
- BART (Bayesian Additive Regression Trees)
- BART/default - *NO* tuning of parameters

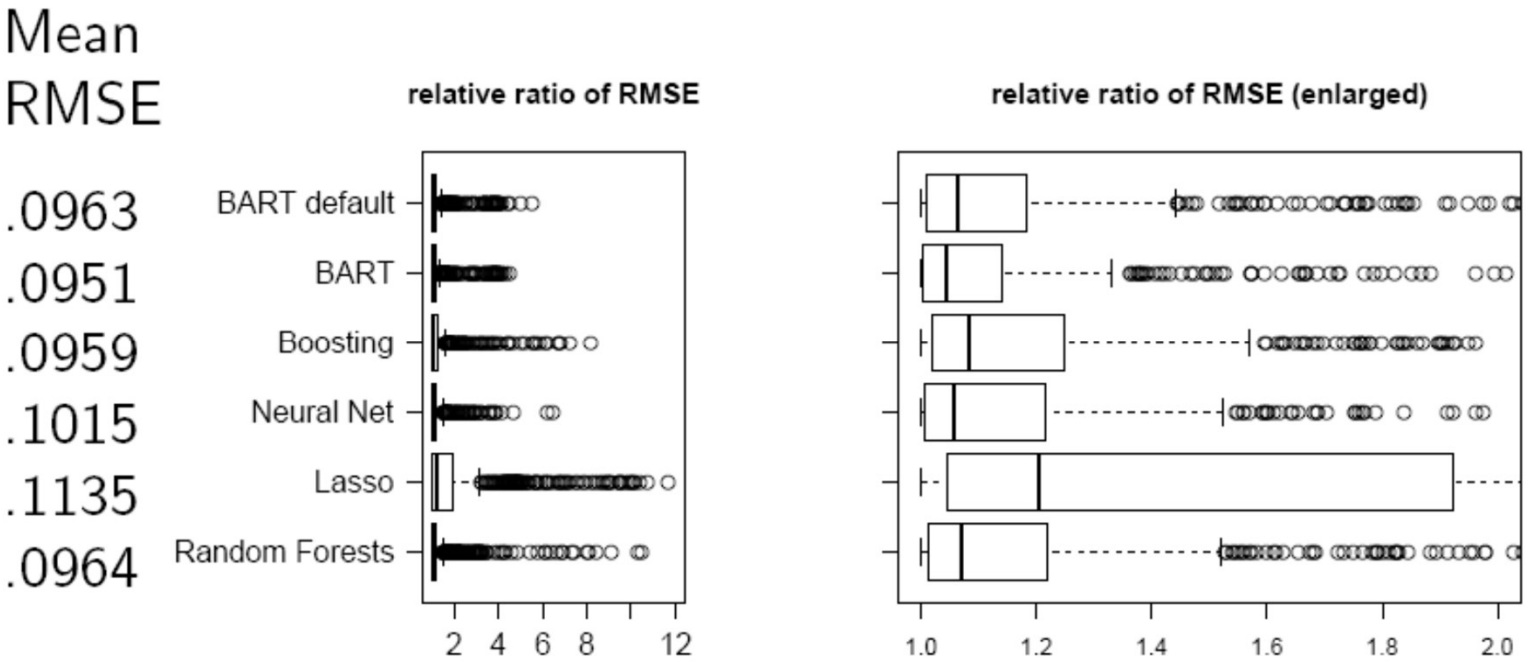
Data from Kim, Loh, Shih and Chaudhuri (2006)

- Up to 65 predictors and 2953 observations

Train on 5/6 of data, test on 1/6

- Tuning via 5-fold CV within training set
- 20 Train/Test replications per dataset

Results: Root Mean Squared Errors



Left: RMSE averaged over datasets and replications

Box Plots: RMSE relative to best

BART is a very strong performer!

One of the 37 Datasets is the well-known Boston Housing Data

Each observation corresponds to a geographic district

y = log(median house value)

13 x variables, stuff about the district

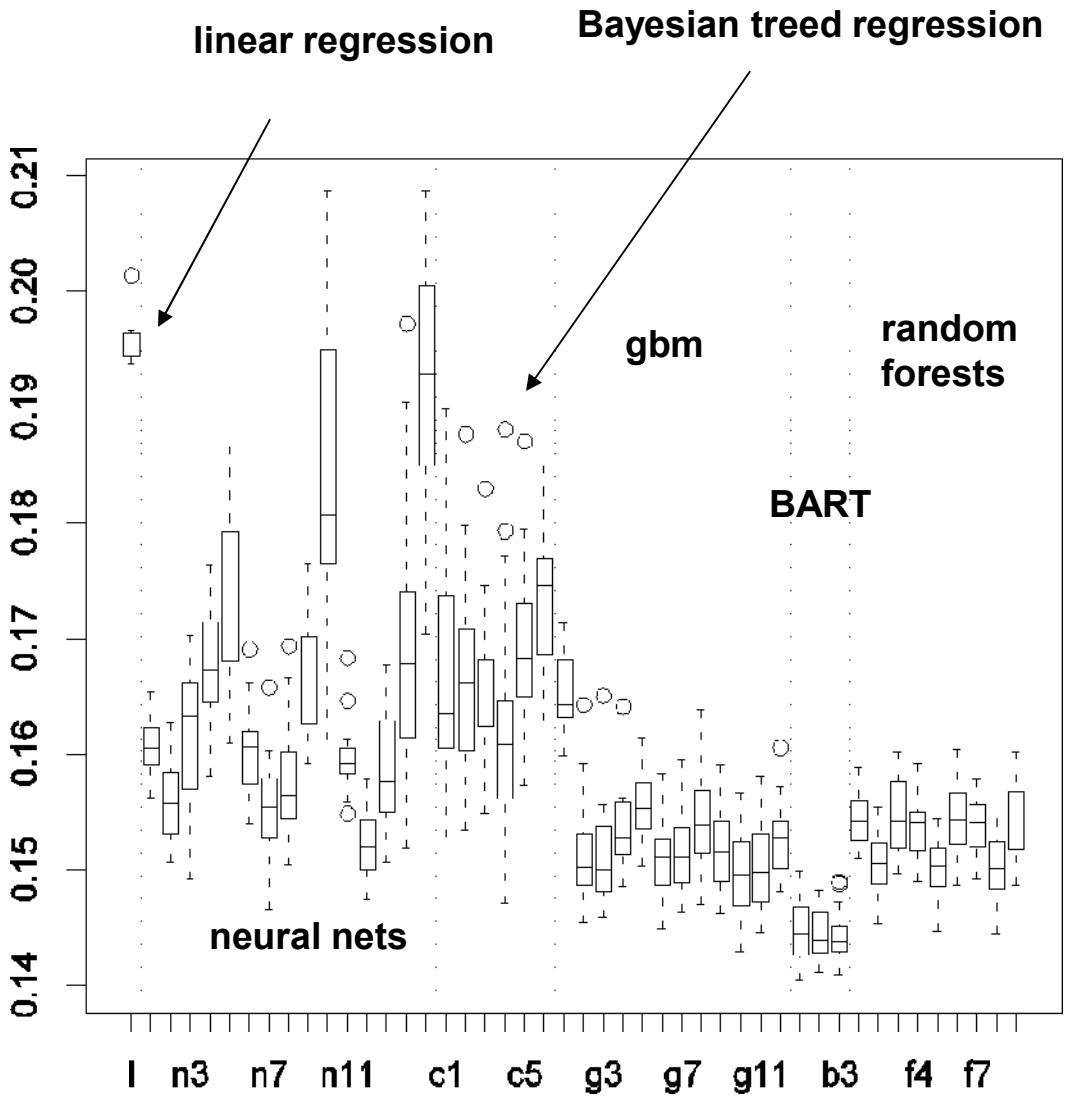
eg. crime rate, % poor, riverfront, size, air quality, etc.

n = 507 observations

Each boxplot depicts 20 rmse's **out-of-sample** for a version of a method.

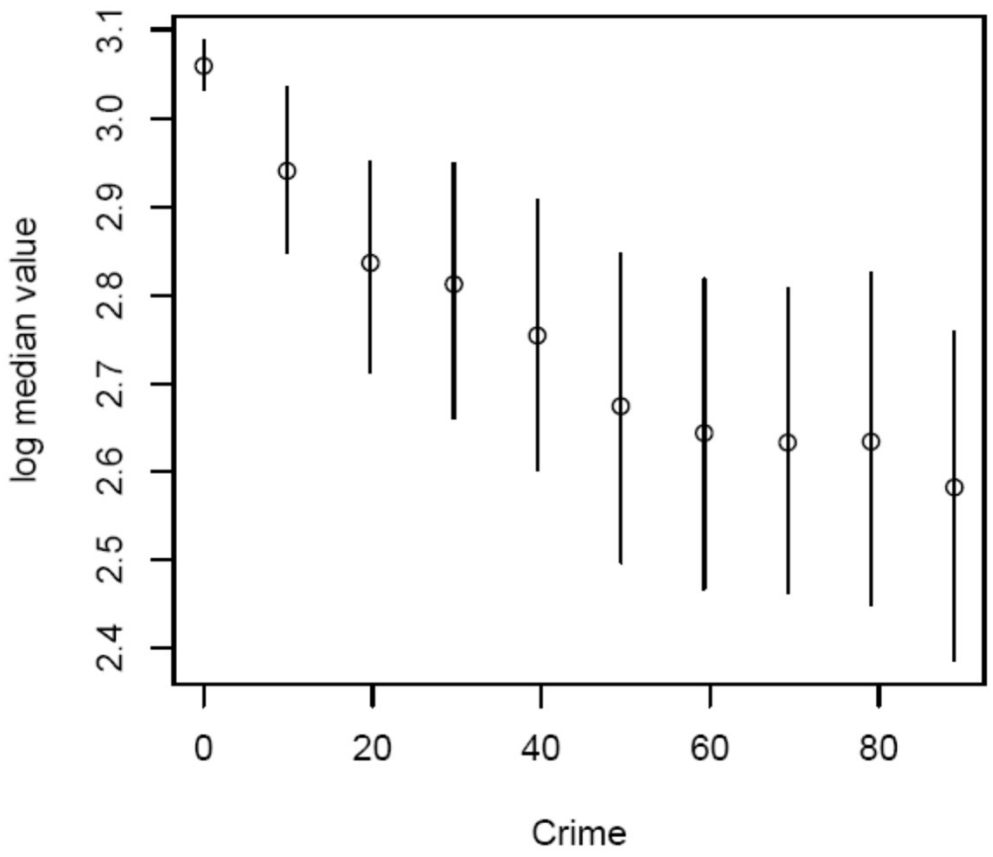
eg.
the method neural nets with a given number of nodes and decay value.

Smaller is better.
BART wins!



BART Offers Estimates of Predictor Effects

Partial Dependence Plot of Crime Effect in Boston Housing



These are estimates of $f_3(x_3) = (1/n) \sum_i f(x_3, x_{ic})$ where $x_c = x \setminus x_3$

Friedman's Simulated Example

$$y = f(x) + \sigma z, \qquad z \sim N(0,1)$$

where

$$f(x) = 10 \sin(\pi x_1 x_2) + 20(x_3 - .5)^2 + 10x_4 + 5x_5 + 0x_6 + \ldots + 0x_{10}$$

10 x's, but only the first 5 matter!

Friedman (1991) used n = 100 observations from this model with $\sigma = 1$ to illustrate the potential of MARS

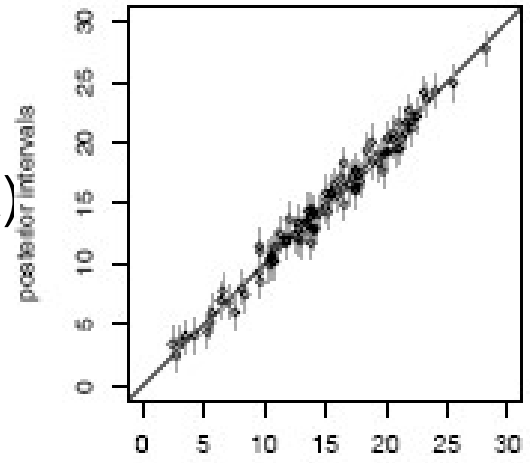
Applying BART to the Friedman Example

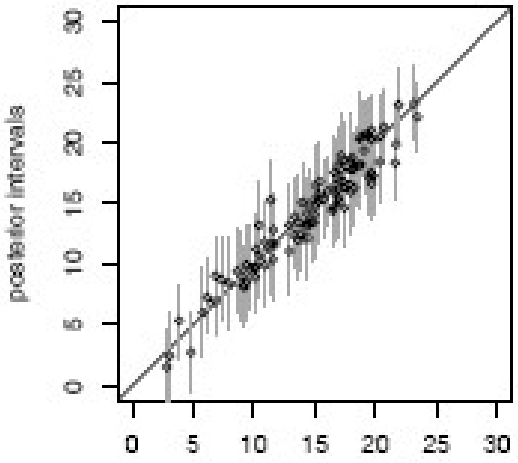
We applied BART with m = 100 trees to n = 100 observations of the Friedman example.

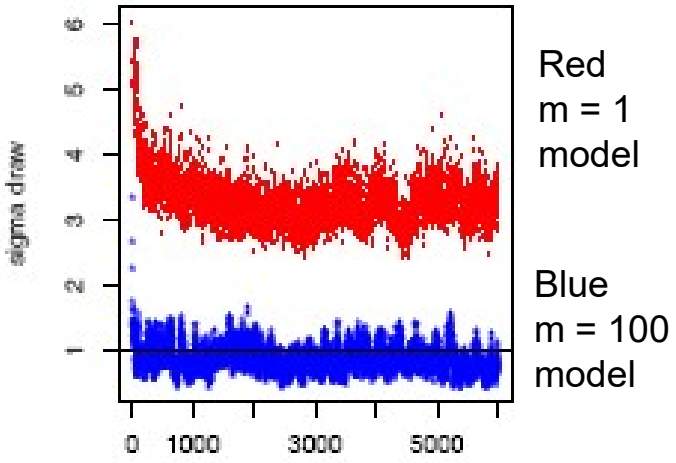
95% posterior intervals vs true f(x)

σ draws

$\hat{f}(x)$







in-sample f(x)

out-of-sample f(x)

MCMC iteration

Comparison of BART with Other Methods

50 simulations of 100 observations of Friedman example

The cross validation domain used to tune each method

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.

BART Wins Again!

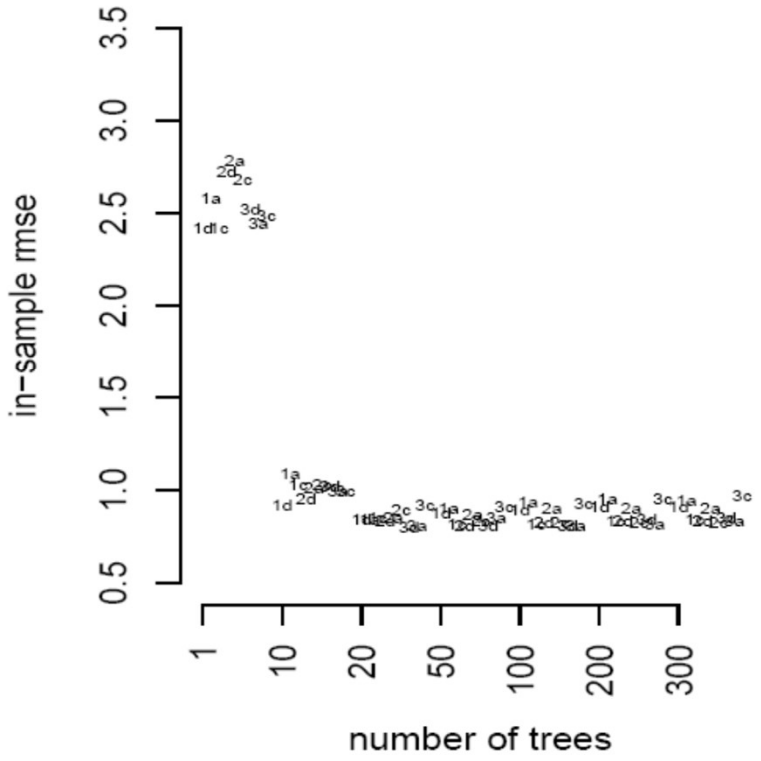
Performance measured on 1000 out-of-sample x's by

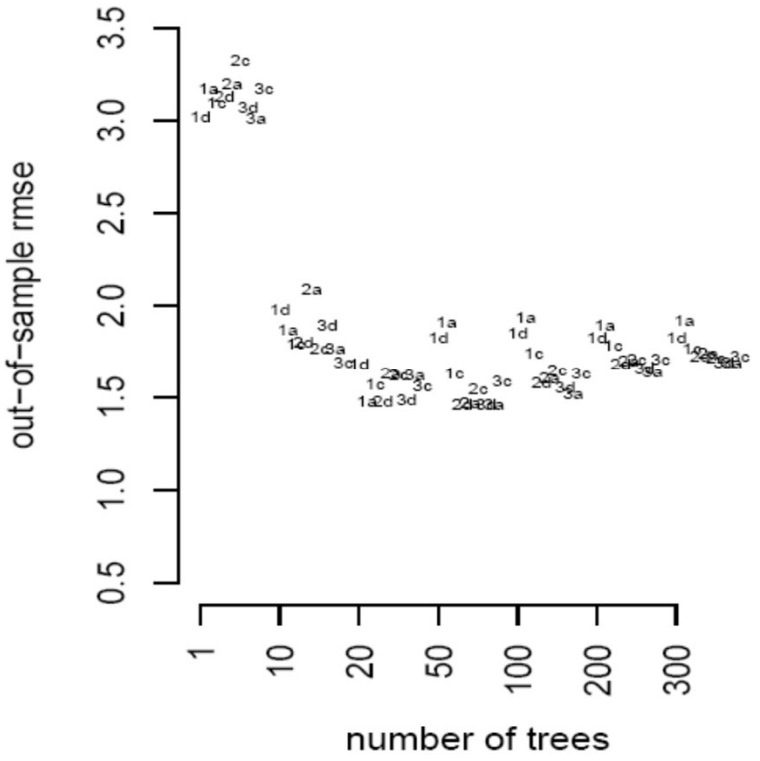
$$\text{RMSE} = \sqrt{\frac{1}{1000}\sum_{i=1}^{1000}(\hat{f}(x_i) - f(x_i))^2}$$

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

BART is Robust to Prior Settings

On the Friedman (1991) example, BART's robust RMSE performance
Is illustrated below where the (ν,q,k,m) choice is varied

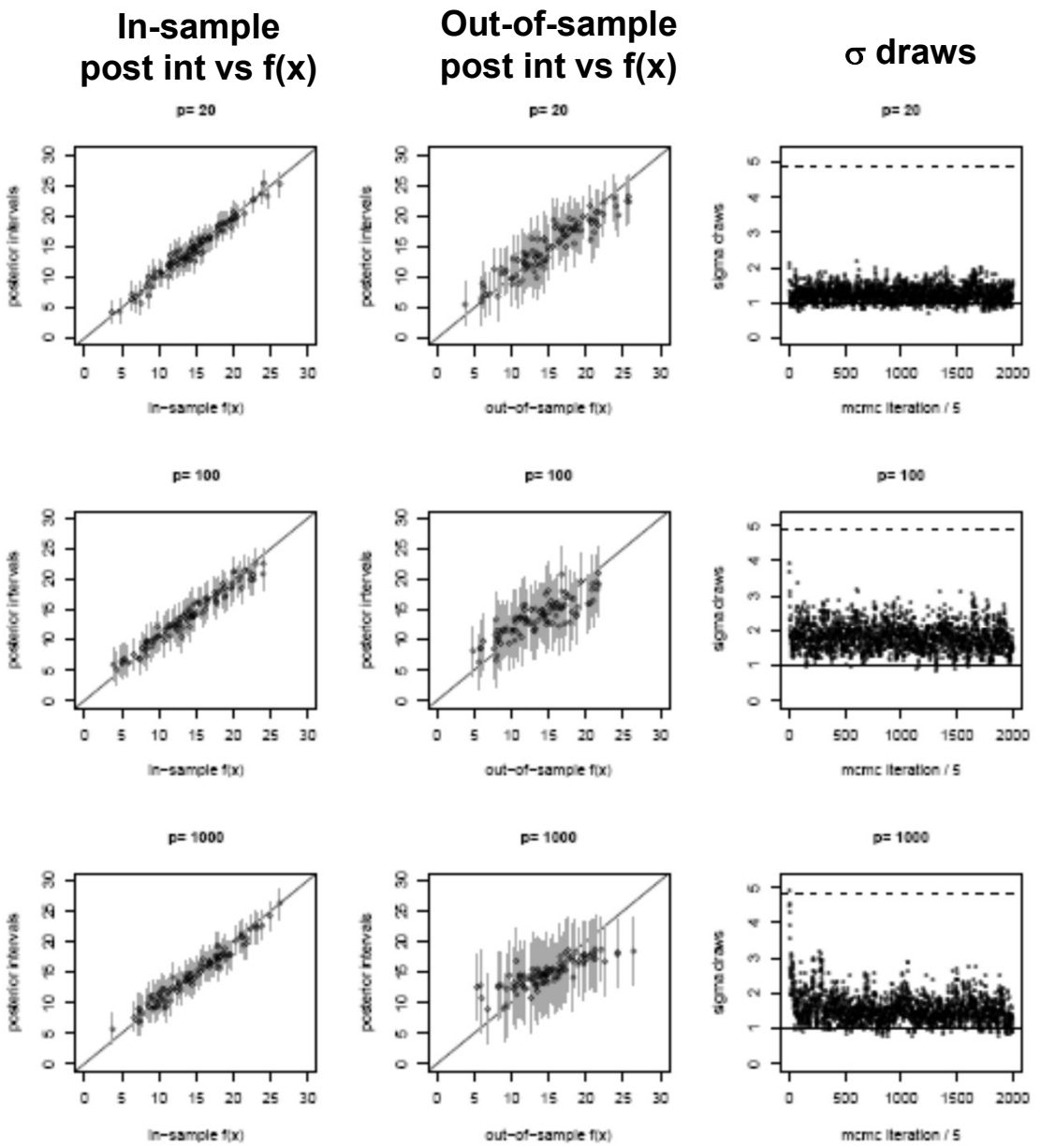




Detecting Low Dimensional Structure in High Dimensional Data

Added many useless x's to Friedman's example

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

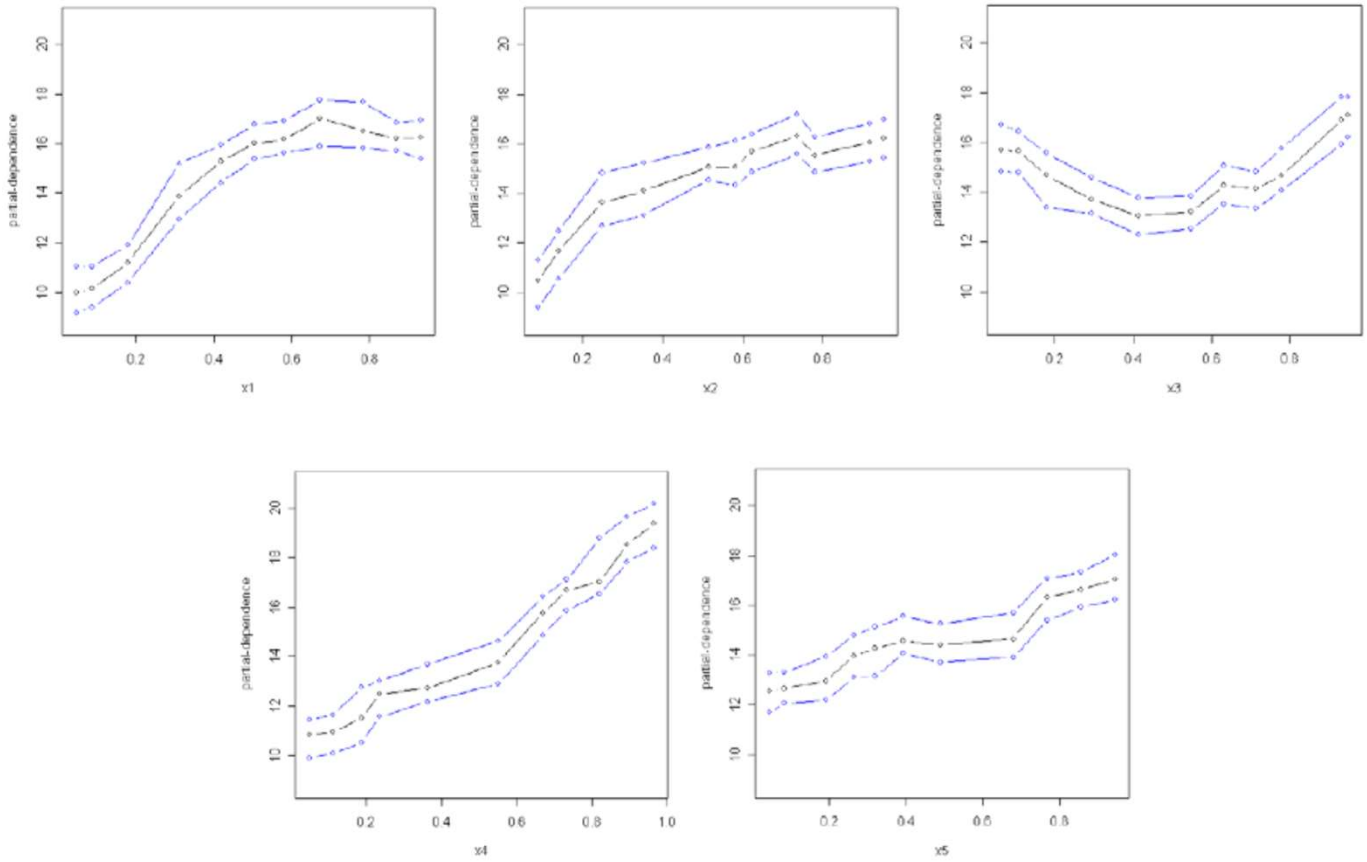


20 x's

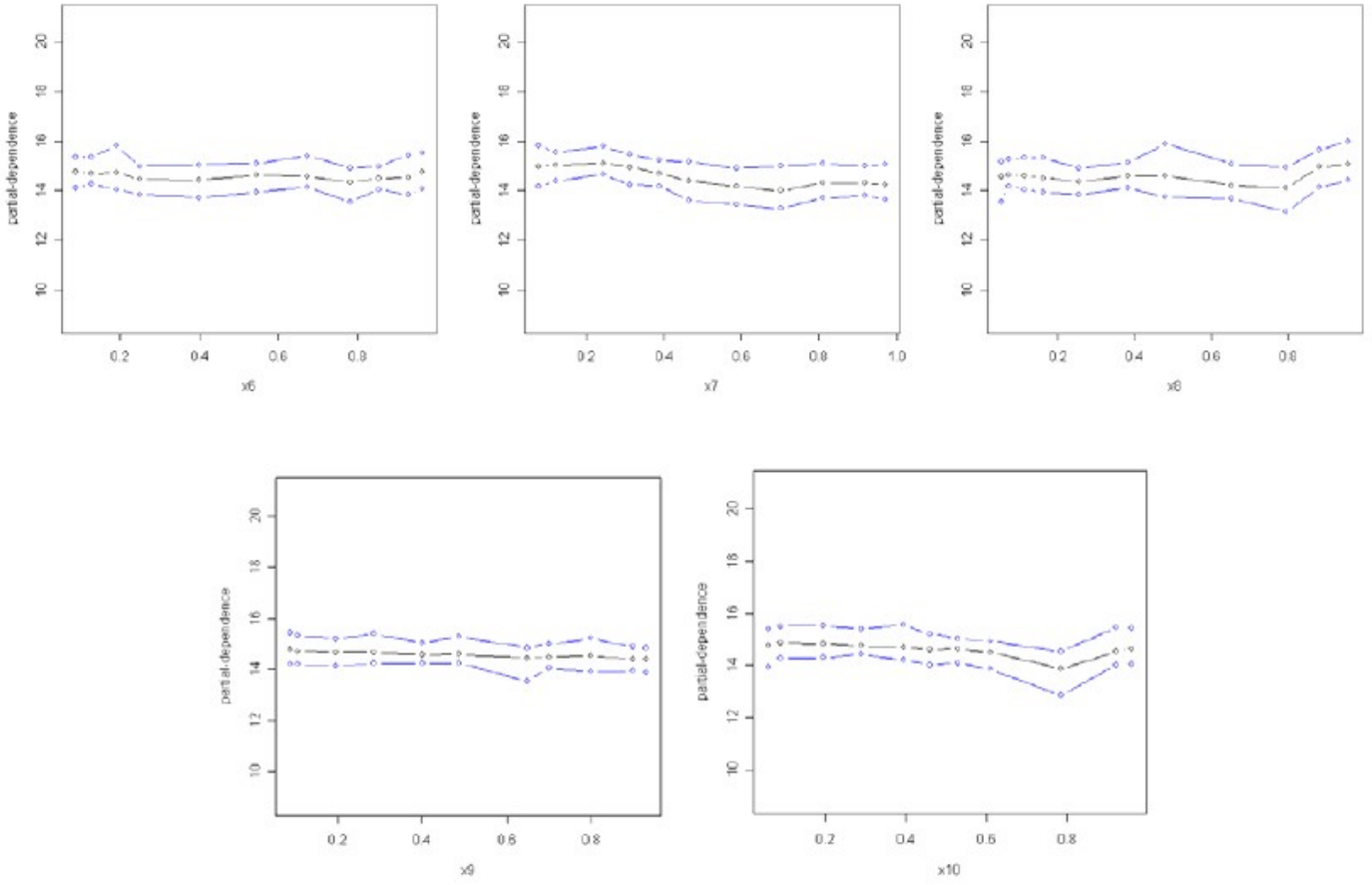
100 x's

1000 x's

Partial Dependence Plots for the Friedman Example
The Marginal Effects of $x_1 - x_5$



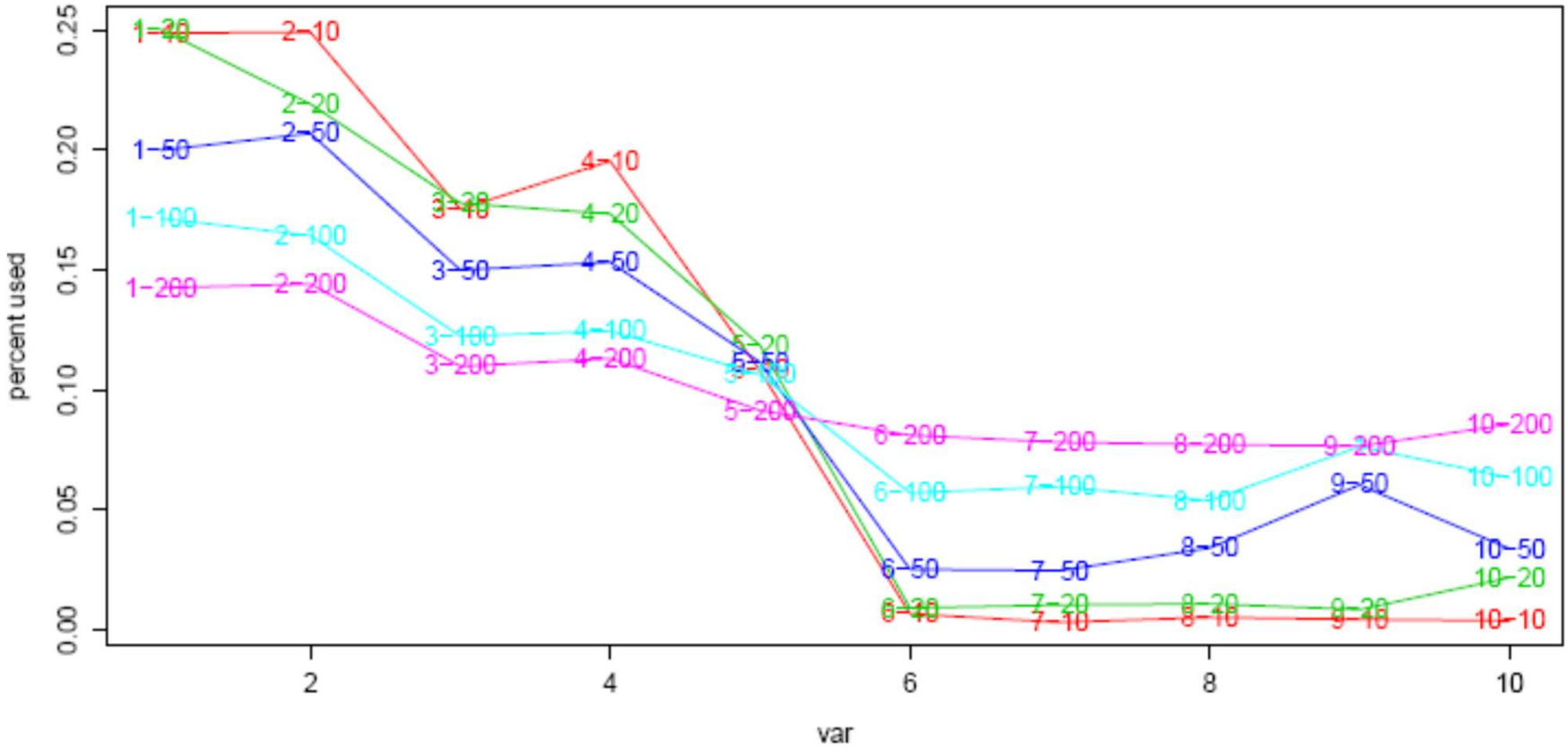
Partial Dependence Plots for the Friedman Example
The Marginal Effects of $x_6 - x_{10}$



Variable Selection via BART

Variable usage frequencies as the number of trees m is reduced

Notation: 2-20 means x_2 with m = 20 trees, etc.



The Football Data

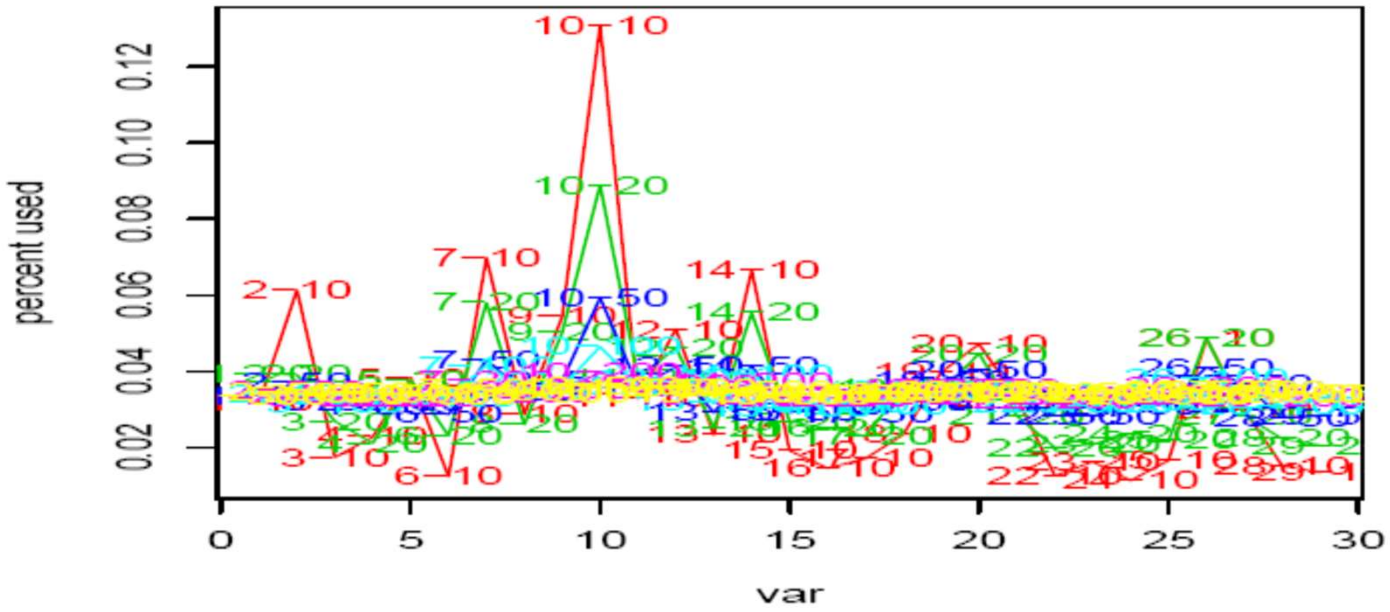
Each observation (n=245) corresponds to an NCAA football game.

y = Team A points - Team B points

29 x's. Each is the difference between the two teams on some measure. eg x_{10} is average points against defense per game for Team A for team B.

Variable Selection for the Football Data

For each draw, for each variable calculate the percentage of time that variable is used in a tree. Then average over trees.

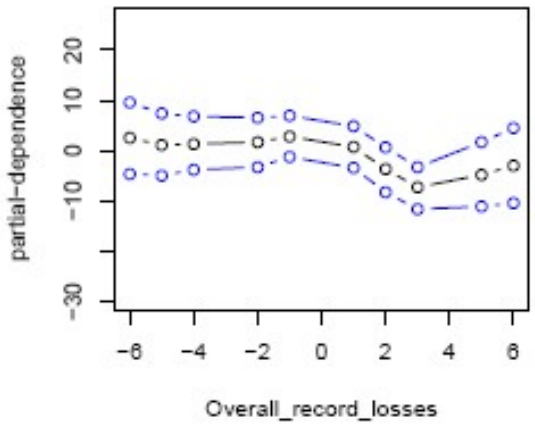


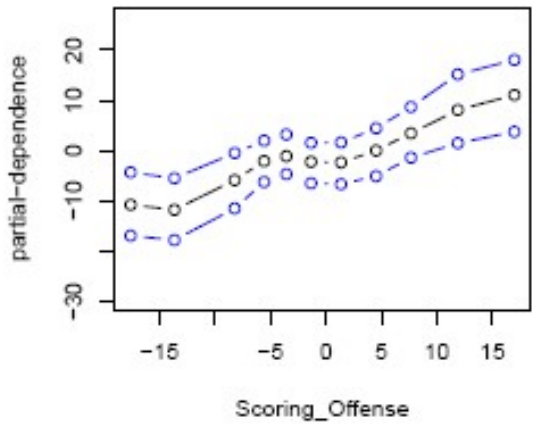
Subtle point: Can't have too many trees. Variables come in without really doing anything.

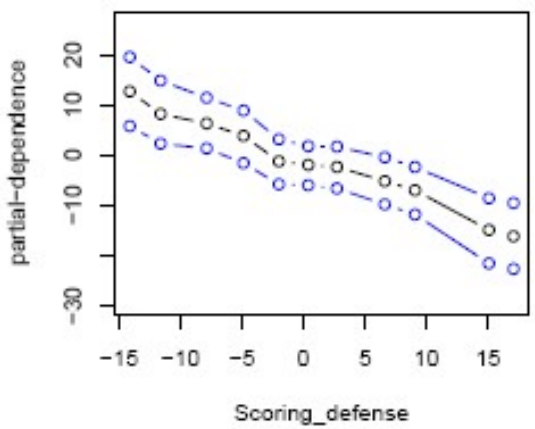
Marginal Effects of the Variables

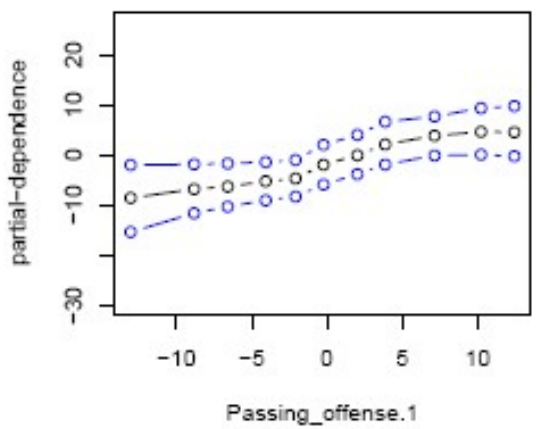
Just used variables 2,7,10, and 14.

Here are the four univariate partial-dependence plots.

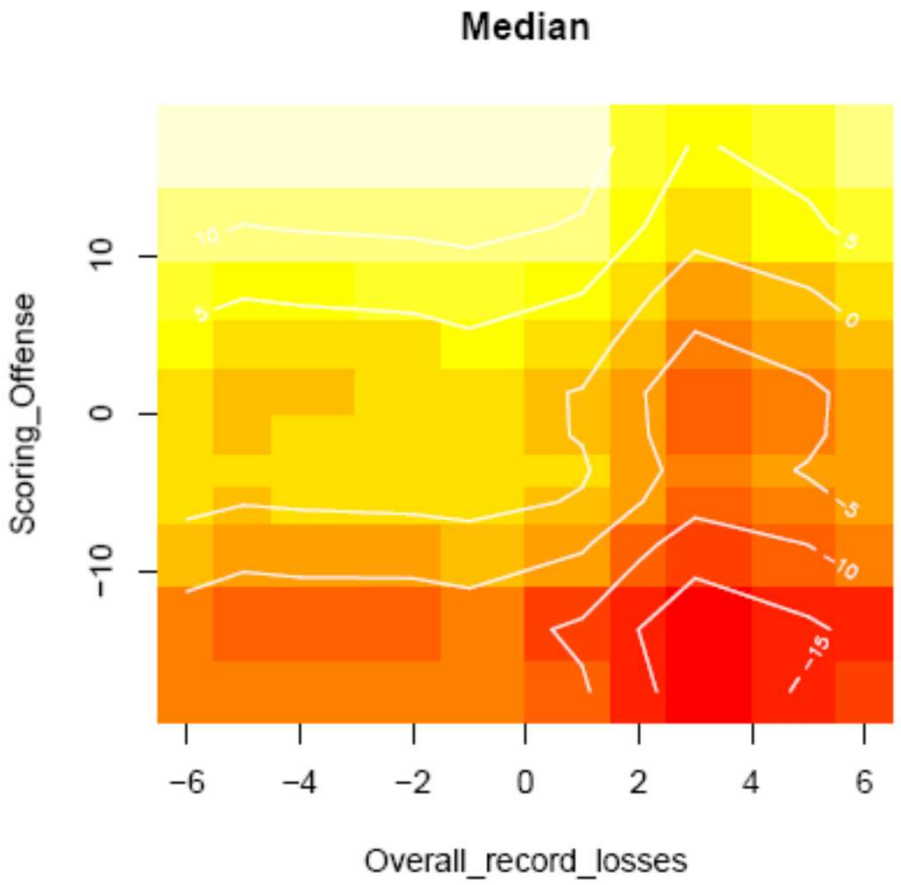








A Bivariate Partial Dependence Plot
The joint effect of two of the x's



Illustrative Application to HIV Data Analysis

Y = LDHL (log of hdl level)

X's = CD4, Age, Sex, Race, Study,
PI1,PI2,NNRTI2, NRTI1, NRTI2,
ABI_349, CRC_71, CRC_72, CRC_55, CRC_73, CRC_10,
ABI_383, ABI_387, ABI_391, ABI_395, ABI_400, ABI_401,
CRC_66, CRC_67, CRC_68, CRC_69

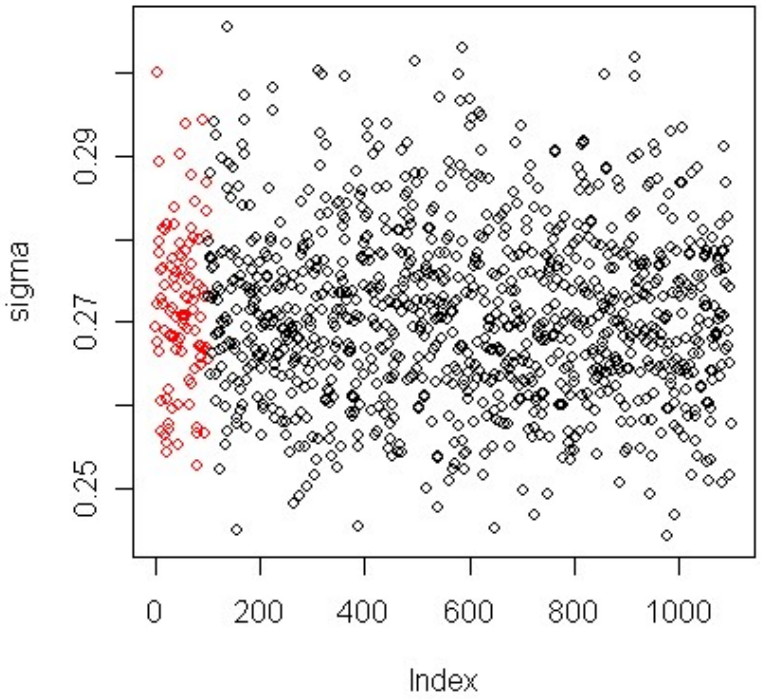
n = 458 patients

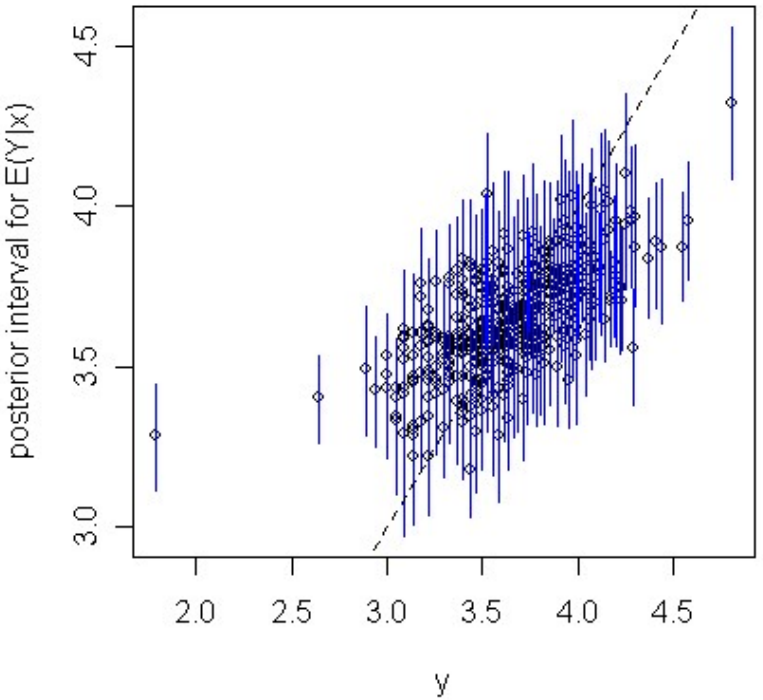
For this data

Least Squares yields $R^2 = 26\%$

BART yields $R^2 = 42\%$

The BART Fit for the HIV Data

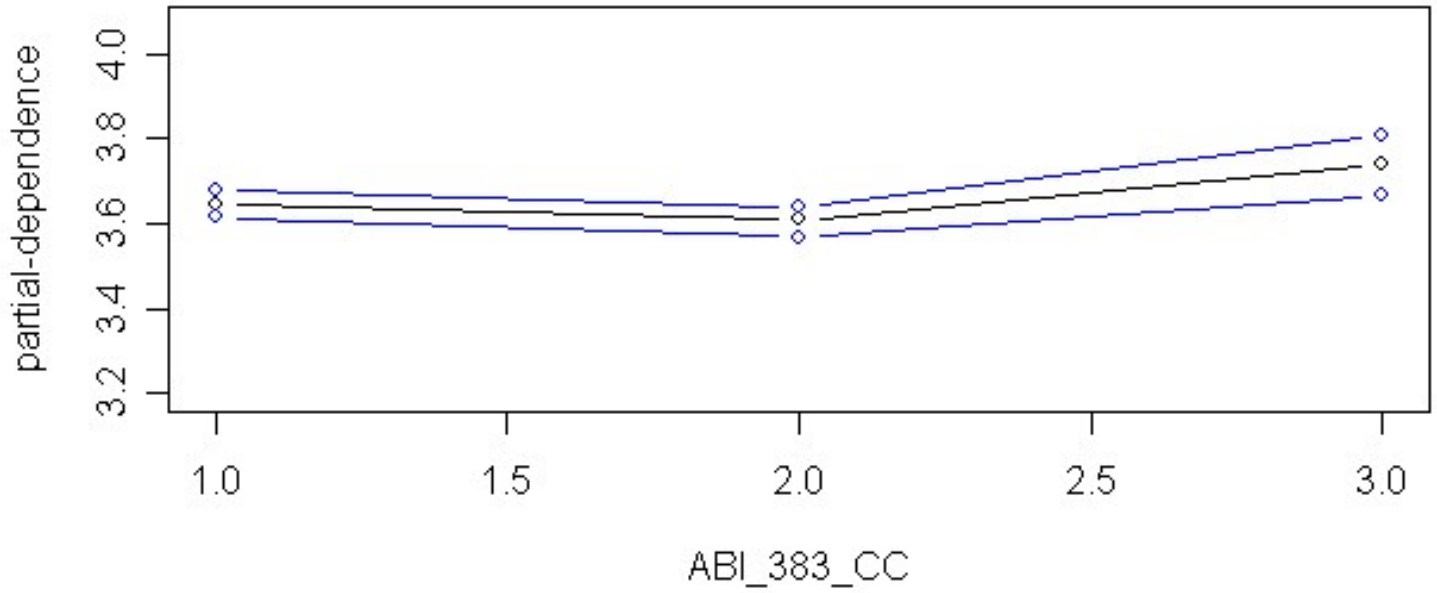




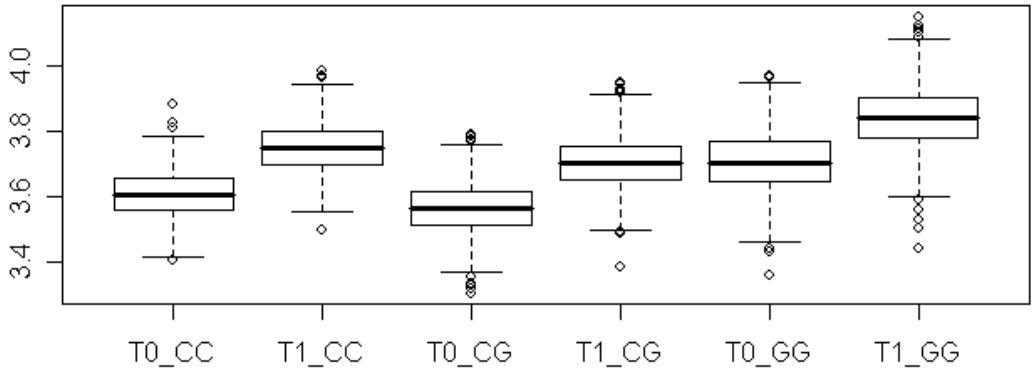
BART suggests there is not a strong signal in x for this y.

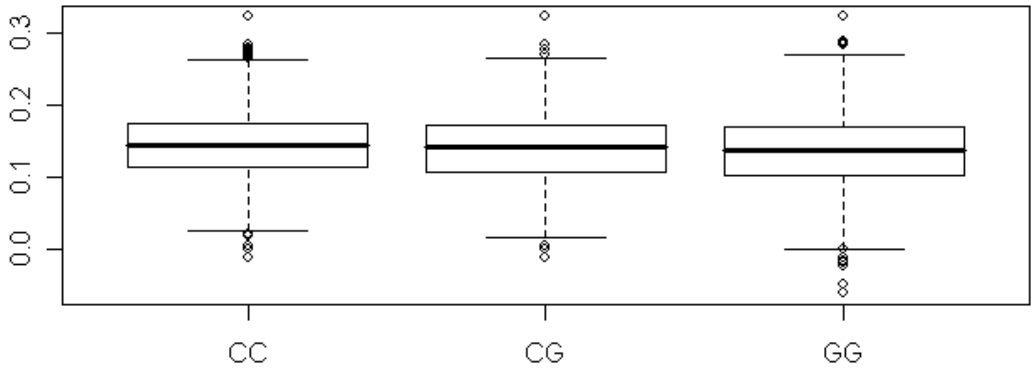
Partial Dependence Plots May Suggest Genotype Effects

For example, the average predictive effect of ABI_383



Predictive Inference about Interaction of NNRTI2 Treatment and ABI_383 Genotype





There appears to be no interaction effect

A Sketch of the Prior

First, introduce prior independence as follows

$$\pi((T_1,M_1),....,(T_m,M_m), \sigma) = [\ \Pi\ \pi(T_j,M_j)\ ]\ \pi(\sigma)$$

$$= [\ \Pi\ \pi(\mu_{ij} \mid T_j)\ \pi(T_j)\ ]\ \pi(\sigma)$$

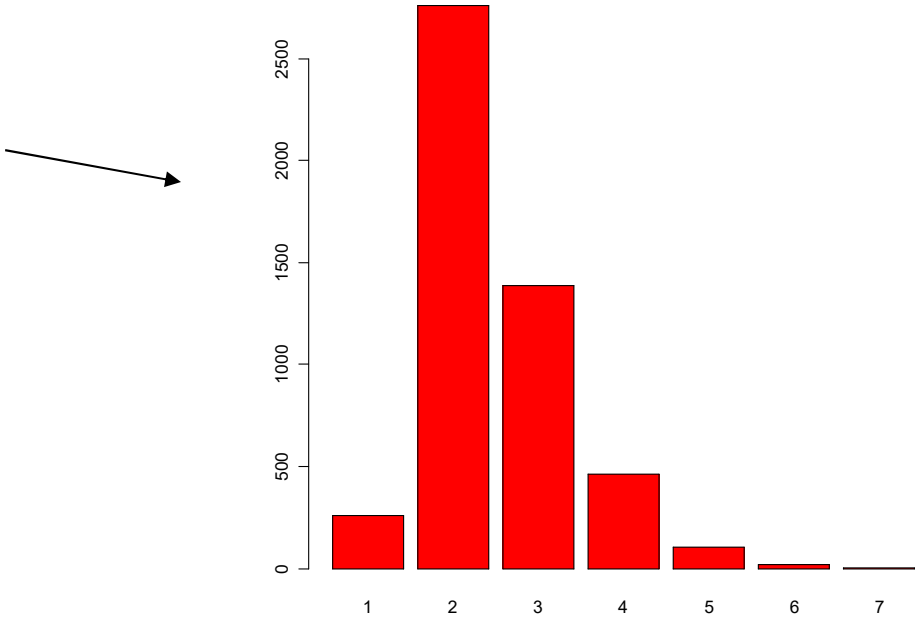
Thus we only need to choose $\pi(T)$, $\pi(\sigma)$, and $\pi(\mu \mid T) = \pi(\mu)$

$\pi(T)$

We specify a process that grows trees:

Step 1) Grow a tree structure with succesive biased coin flips
Step 2) Randomly assign variables to decision nodes
Step 3) Randomly splitting rules to decision nodes

Marginal prior on number of bottom nodes.



Hyperparameters chosen to put prior weight on small trees!!

$\pi(\mu \mid T)$

For each bottom node μ, let $\mu \sim N(0,\sigma_\mu^2)$

To set σ_μ, we proceed as follows:

First standardize y so that E(y | x) is in [-.5,.5] with high probability.

Note that in our model, E(y | x) is the sum of m independent μ's (a priori), so that the prior standard deviation of E(y | x) is $\sqrt{m}\sigma_\mu$

Thus, we choose σ_μ so that $k\sqrt{m}\sigma_\mu = .5 \Rightarrow \sigma_\mu = \frac{.5}{k\sqrt{m}}$ for a suitable value of k

Default choice is k = 2.

k is the number of standard deviations of E(y | x) from the mean of 0 to the interval boundary of .5

Note how the prior adapts to m: σ_μ gets smaller as m gets larger.

$\pi(\sigma)$

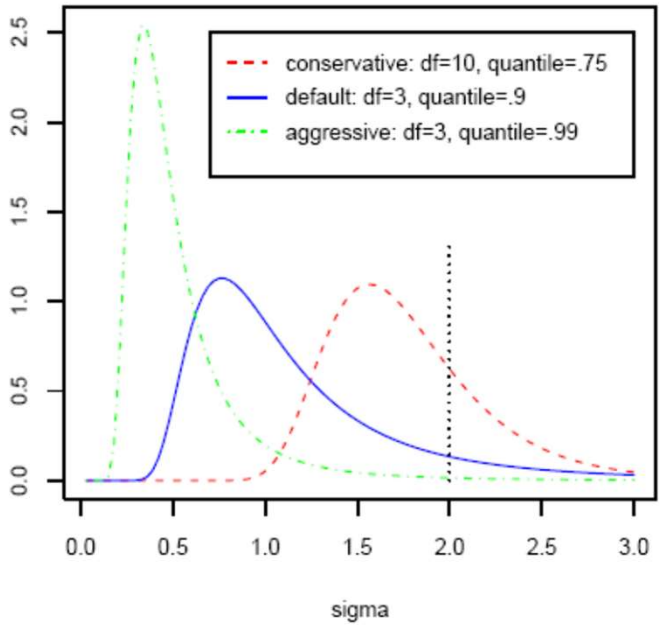
Let $\sigma^2 \sim \frac{\nu\lambda}{\chi^2_\nu}$ and consider ν = 3, 5 or 10.

To set λ, we use a rough overestimate of σ based on the data (such as sd(y) or the LS estimate for the saturated linear regression).

Determine λ by setting a quantile such as .75, .95 or .99 at this rough estimate.

$$\hat{\sigma} = 2$$

The three priors we have been using:



A Sketch of the MCMC algorithm

$$y = g(x;T_1,M_1) + g(x;T_2,M_2) + \ldots + g(x;T_m,M_m) + \sigma z$$

The "parameter" is: $\theta = ((T_1,M_1),....(T_m,M_m),\sigma)$

"Simple" Gibbs sampler:

(1) $\sigma \,|\, \{T_j\},\{M_j\},\text{data}$

(2) $(T_j,M_j) \,|\, \{T_i\}_{i\neq j},\{M_i\}_{i\neq j},\sigma,\text{data}$ (Bayesian backfitting)

(1) Subtract all the g's from y to update σ
(2) Subtract all but the j^{th} g from y to update (T_j,M_j)

Using the decomposition

$$p(T,M \mid \text{data}) = p(T \mid \text{data})\, p(M \mid T, \text{data})$$

and the fact that p(T | data) is available under our prior,
we sample

$$(T_j, M_j) \mid \{T_i\}_{i \neq j}, \{M_i\}_{i \neq j}, \sigma, \text{data}$$

by first drawing T from p(T | data), and then drawing M from p(M | T, data).

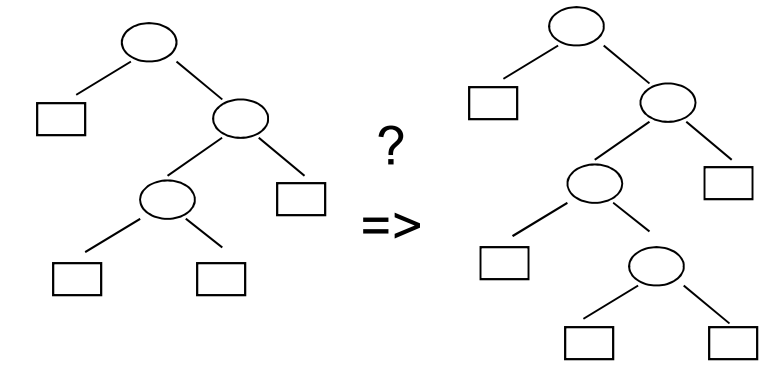
Drawing M from p(M | T,data) is routine

Just simulate μ's from the posterior under a conjugate prior

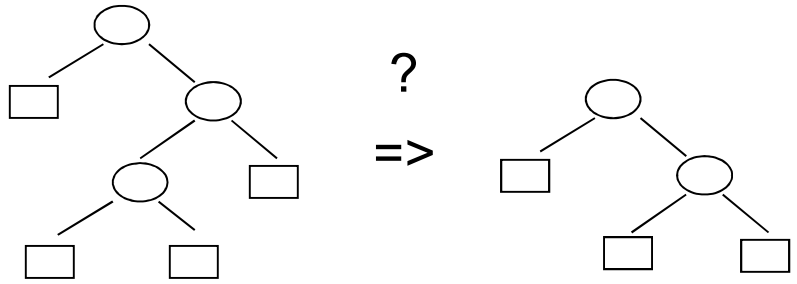
To draw T from p(T | data), we use a Metropolis-Hastings algorithm.

Given the current T, we propose a modification and
then either move to the proposal or repeat the old tree.

In particular we use proposals that change the size of the tree:



propose a more complex tree



propose a simpler tree

More complicated models will be accepted if the data's insistence
overcomes the reluctance of the prior.

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

Thus, at each iteration, T_i, M_i and σ are updated.

This is a Markov chain such that the stationary distribution is the posterior.

Each tree contributes a small part to the fit, and the fit is swapped around from tree to tree as the chain runs.

The Dynamic Random Basis in Action:

As we run the chain, we often observe that an individual tree grows quite large and then collapses back to a single node.

This illustrates how each tree is dimensionally adaptive.

Using the MCMC Output to Draw Inference

At iteration i we have a draw from the posterior of the function

$$\hat{f}_i(\cdot) = g(\cdot, T_{1i}, M_{1i}) + g(\cdot, T_{2i}, M_{2i}) + \cdots + g(\cdot, T_{mi}, M_{mi})$$

To get in-sample fits we average the $\hat{f}_i(\cdot)$ draws to obtain $\bar{f}_i(\cdot)$

Thus, $\bar{f}_i(x)$ estimates f(x).

Posterior uncertainty is captured by variation of the $\hat{f}_i(x)$

Where do we go from here?

BART (and probably other nonparametric methods) can give us a sense of

- E(y |x)
- the distribution of y around E(y|x)
- the individual effects of the $x_{j's}$
- a subset of $x_1,...,x_p$ related to y

This information would seem to be very valuable for model building. The next step is how?

To be continued…