

# Variable Selection via a Bayesian Ensemble

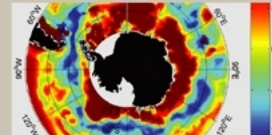
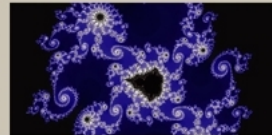
Hugh Chipman, Acadia University  
Edward George, University of Pennsylvania and  
Robert McCulloch, University of Chicago

R package BayesTree available on CRAN



*Mathematics and Statistics at Acadia*

FACULTY OF PURE AND APPLIED SCIENCES



## A General Model-Free Regression Setup

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

### **BART is a Bayesian ensemble method that can:**

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

(making no assumptions about  $f$ ).

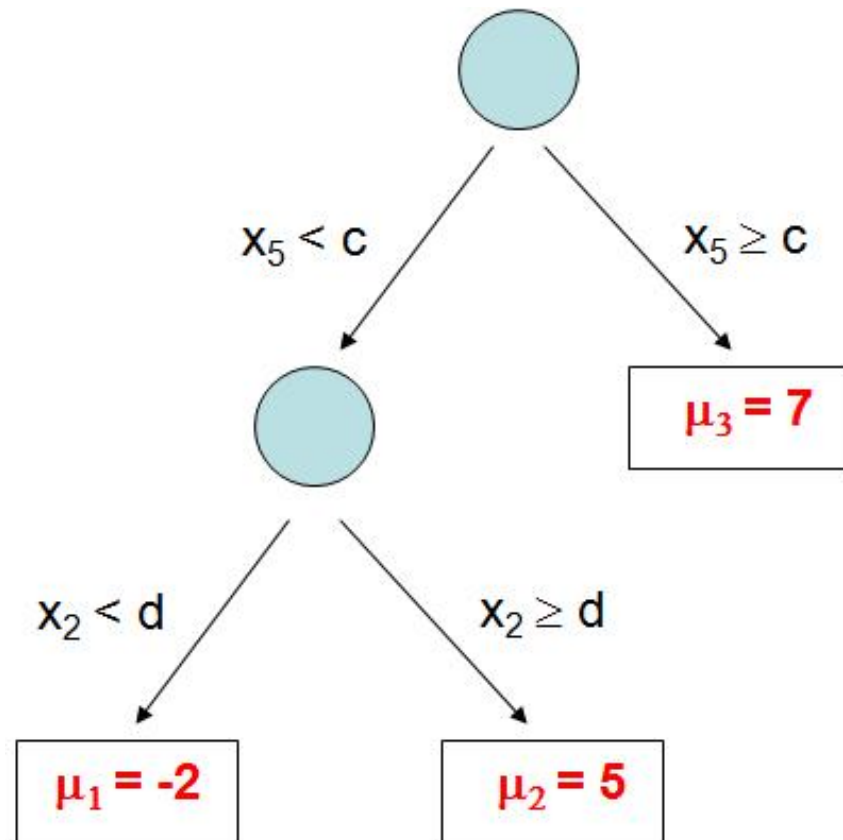
Remark: In what follows we will assume  $\varepsilon \sim N(0, \sigma^2)$  for simplicity, but extension to a more general DP process normal mixture model for  $\varepsilon$  works just fine.

## 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  $\mu$  value to  $x$  where:

- $T$  denotes the tree structure including the decision rules
- $M = (\mu_1, \mu_2, \dots, \mu_b)$  denotes the set of terminal node  $\mu$ 's.



**A single tree model:**  $y = g(x; T, M) + \sigma Z, \quad Z \sim N(0, 1)$

## A “Sum of Trees” Model

Let  $(T_1, M_1), (T_2, M_2), \dots, (T_m, M_m)$  identify a set of  $m$  trees and their  $\mu$ 's.

Our data model is

$$Y = g(x; T_1, M_1) + g(x; T_2, M_2) + \dots + g(x; T_m, M_m) + \sigma Z,$$

$$Z \sim N(0, \sigma^2).$$

- For a given input value  $x$ , each  $g(x; T_i, M_i)$  will output a corresponding  $\mu$ ; the prediction is the sum of the  $\mu$ 's
- Additive and interaction effects can be modelled.

## Completing the model with a regularization prior:

$$Y = g(x; T_1, M_1) + g(x; T_2, M_2) + \dots + g(x; T_m, M_m) + \sigma Z,$$

For  $m$  large,

- many parameters  $(T_1, \dots, T_m, M_1, \dots, M_m, \sigma)$
- $g(x; T_1, M_1), g(x; T_2, M_2), \dots, g(x; T_m, M_m)$  is a highly redundant “over-complete basis”

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

$$\pi((T_1, M_1), (T_2, M_2), \dots, (T_m, M_m), \sigma)$$

Strongly influential prior  $\pi$  is used to keep each  $g()$  small  
(**weak learners**)

## BART Implementation

BART's fully Bayesian specification implies that information about all unknowns, namely  $\Theta = ((T_1, M_1), (T_2, M_2), \dots, (T_m, M_m), \sigma)$ , is captured by the posterior

$$\pi(\Theta|Y) \propto p(Y|\Theta)\pi(\Theta)$$

Thus to implement BART we need to simply:

1. Construct the prior  $\pi(\Theta)$ 
  - Small trees are most likely
  - Outputs ( $\mu$ 's) are near zero (scaled using observed  $Y$ )
  - Extremely effective default choices available
2. Calculate the posterior  $\pi(\Theta|Y)$ 
  - Bayesian backfitting MCMC (variation on Hastie & Tibshirani 2000)
  - Some analytic simplification possible

$$Y = g(x; T_1, M_1) + g(x; T_2, M_2) + \dots + g(x; T_m, M_m) + \sigma Z,$$

### 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  $m$ .
- Choose  $m$  large (say 100 or 200) for best estimation of  $E(y|x)$  and prediction  
(more trees yields more approximation flexibility)
- Choose  $m$  small for variable selection  
(fewer trees forces the  $x$ 's to compete for entry).

# Does it work?: A large empirical study

6 learners  $\times$  42 datasets

- **Learners:**

- Random Forests
- Boosting (Friedman's gradient boosting machine)
- Linear regression with lasso
- Neural networks (single hidden layer)
- BART (Bayesian Additive Regression Trees)
- BART-cv (choose prior parameters via cross-validation)

- **Datasets:**

- From Kim, Loh, Shih and Chaudhuri (2006)
- Up to 65 predictors and 6806 observations

- **Details:**

- Train on 5/6 of data, test on 1/6
- Learners tuned via 5-fold CV within training set.
- 20 Train/Test replications per dataset

## Results: Root Mean Square Errors

Average test set RMSE across 42 datasets (after standardizing  $Y$ ):

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

BART-CV:	0.5042
Boosting:	0.5089
BART:	0.5093
Random Forest:	0.5097
Neural Net:	0.5160
Lasso:	0.5896

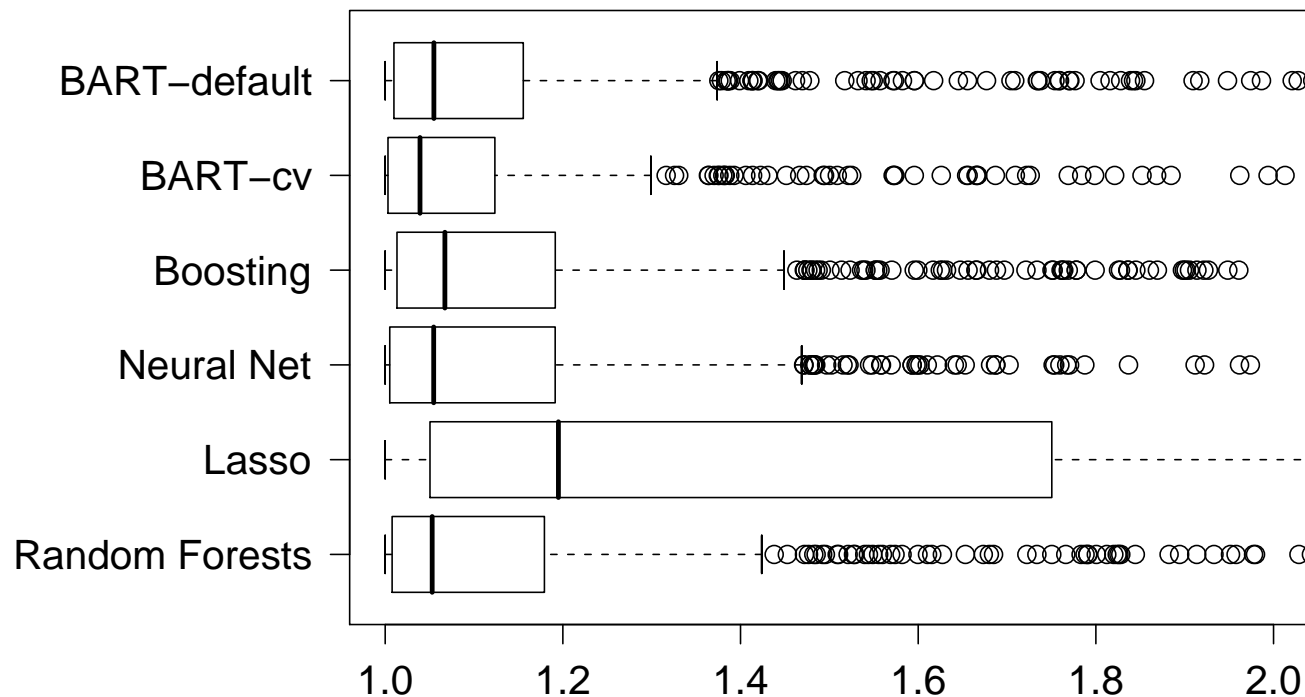
### Some comments:

- BART does quite well
- Treating BART like a machine learner only gives a modest improvement.
- It's actually pretty surprising how close the different models are.

## Results: Relative Root Mean Square Errors

“Relative”  $\Rightarrow$  for each replicate on each data set, we identify best model, and all RMSEs are divided by the error of the best model.

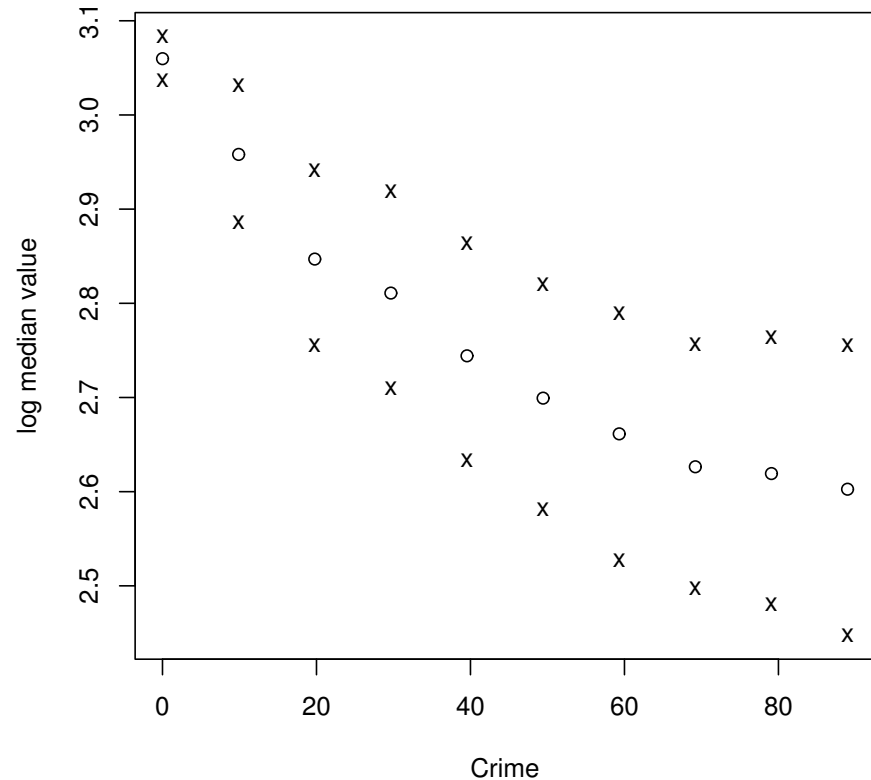
$\Rightarrow$  “1.0” is best, “2.0” is a RMSE twice as large as best model.



# BART offers estimates of predictor effects

Partial dependence plot of crime effect in Boston Housing Data

- Estimate of  $f_3(x_3) = \frac{1}{n} \sum_i f(x_3, x_{i,c})$  where  $x_c = x \setminus x_3$
- 'o' = posterior mean
- 'x' = 90% posterior intervals



- Almost all crime rates are in the 0-5 range.
- Bounds widen as we have less data (high crime rate).

## Other good things about BART...

... that I could mention in detail but won't:

- Robustness of prediction to prior specification
- Quick burn-in and convergence of MCMC
- Ability to identify low-dimensional structure in high-dimensional data.

## Onwards to variable selection...

But first note that this BART overview wasn't only a sales pitch.

Variable selection isn't terribly useful if you have the wrong functional form of model....

... so an adaptive, flexible model is a good place to start for variable selection.

## Variable selection

- We could modify BART to look for the most important variables (e.g. like Bayesian variable selection in regression)...
- ... But it's already doing this.
  - Every MCMC step involves a stochastic search for good splits in the tree nodes.
  - We've discovered that the posterior frequencies of variable inclusions contain highly relevant information for variable selection.
  - ... You just need to process and explore this information appropriately.

## Variable selection

Simple example:

- Suppose we have  $p = 3$  variables and use  $m = 5$  trees.
- Below is one realization of trees  $T_1, T_2, \dots, T_5$ .

Tree	# splits	usage frequencies		
		$X_1$	$X_2$	$X_3$
1	2	0	1	1
2	1	0	1	0
3	3	0	2	1
4	0	0	0	0
5	2	0	2	0
total	8	0	6	2
rel.freq		0	0.75	0.25

Every MCMC draw gives us a relative frequencies of usage.

- We report the posterior average of these as an index of variable importance (percent.used in plots).
- Note that percent.used sums to 1.

## Variable selection:Friedman example

Friedman (1990) used the following example to demonstrate his MARS algorithm:

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

where

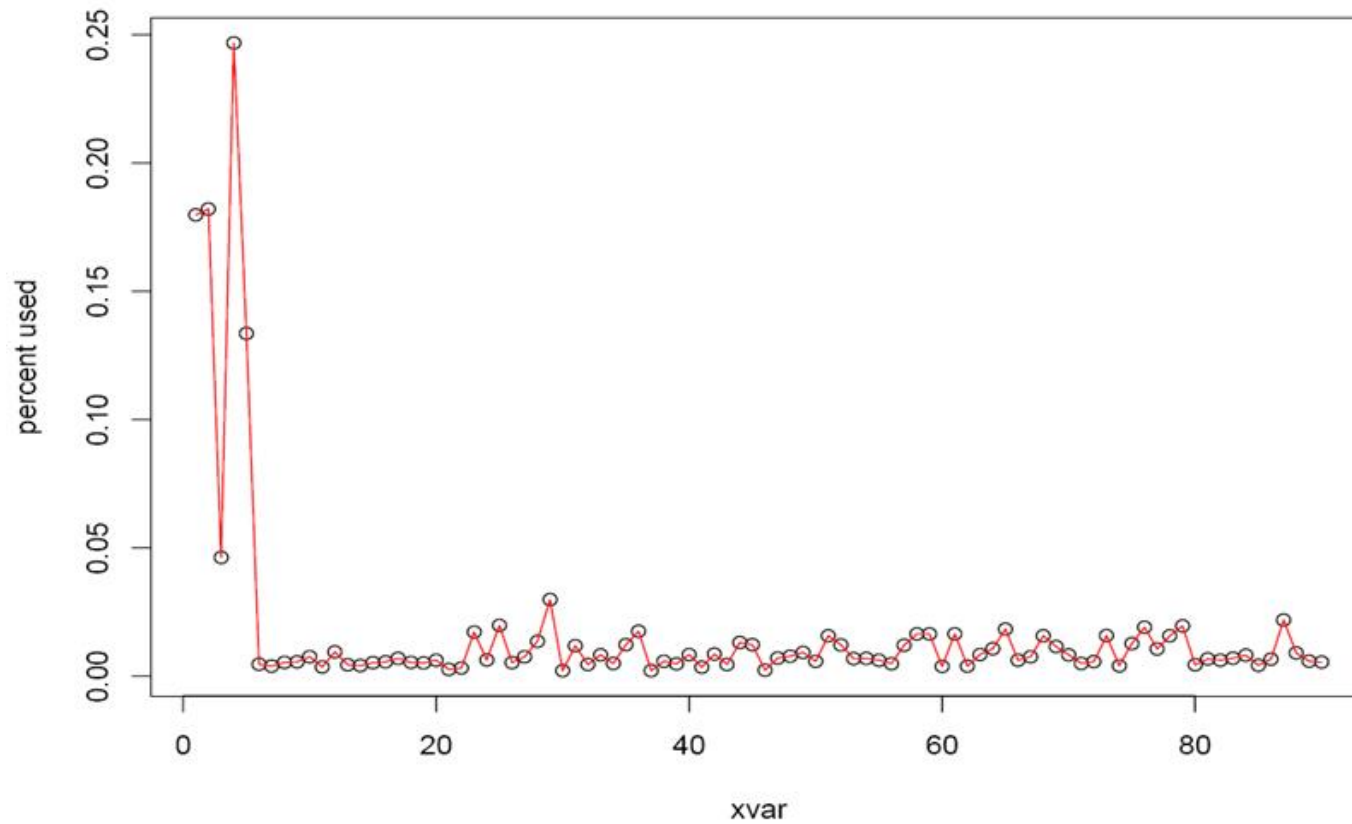
$$f(x) = 10 \sin(\pi x_1 x_2) + 20(x_3 - .5)^2 + 10x_4 + 5x_5 + 0x_6 + \dots + 0x_{10}$$

- Note that there are 10  $X$ 's but only the first 5 matter.
- Each  $X$  is  $U(0, 1)$ .
- Next slide reports percent .used for 5 different BART models with  $m = 10, 20, 50, 100, 200$ .

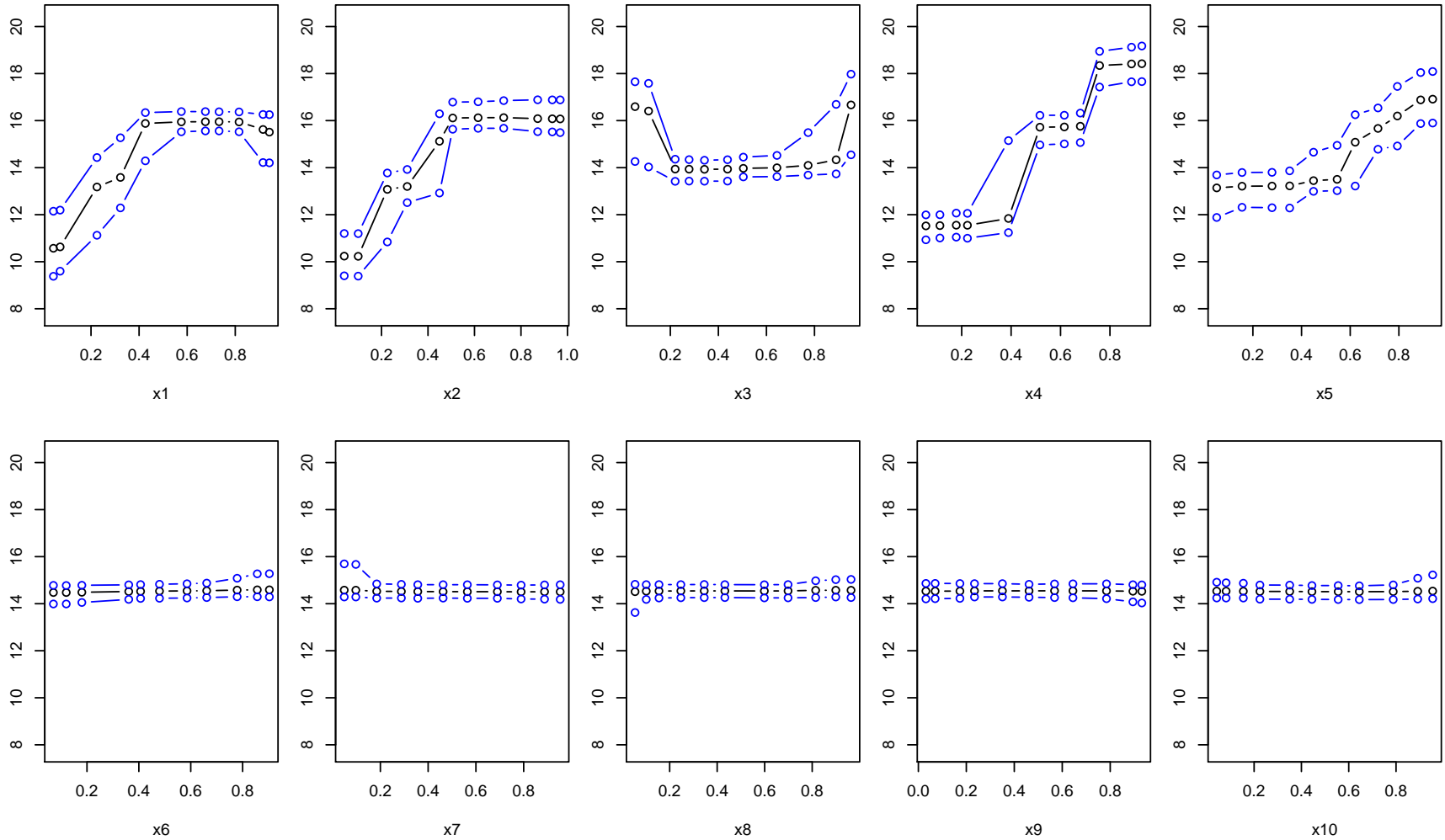


# Variable selection via BART

With  $m = 10$  trees, it remains effective with  $n = 100$  and  $p = 90$   $X$ 's.



# Partial dependence plots - Friedman example



## Model selection with no model

Consider Friedman example with

- $n = 100$  observations
- $p = 90$  predictors (85 junk variables)

Simulate two different datasets

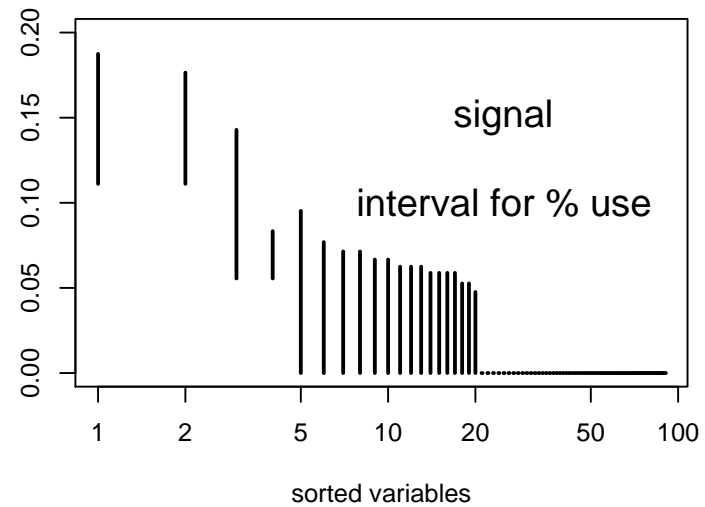
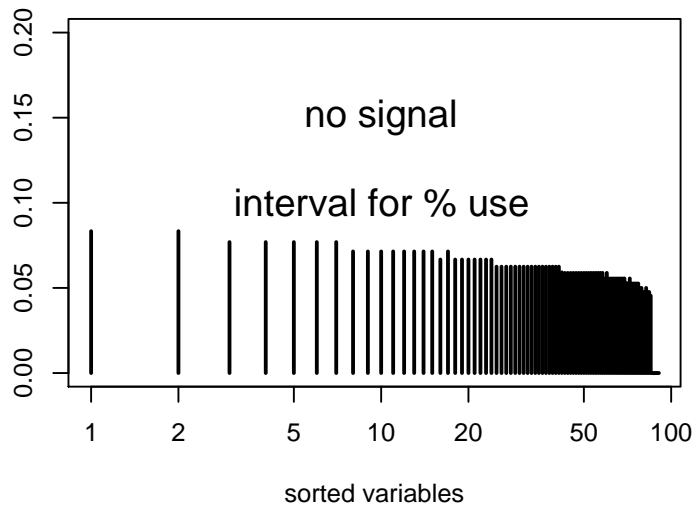
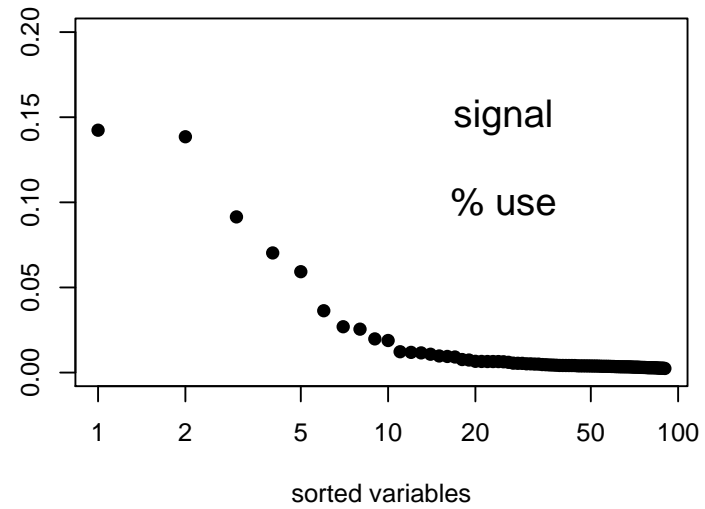
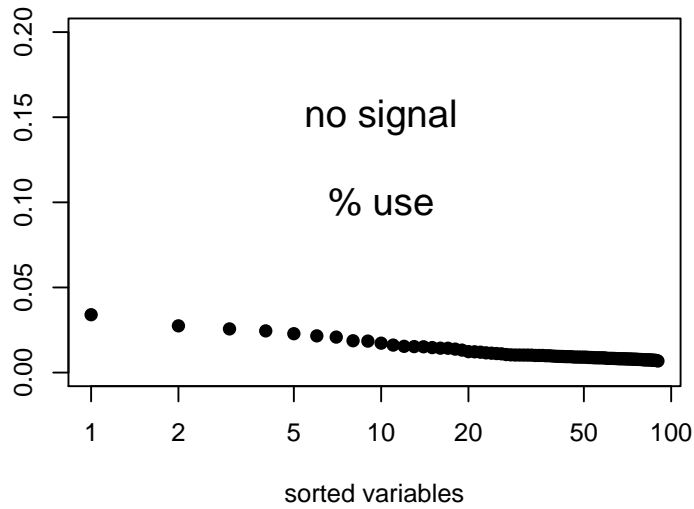
- Dataset 1 (“no signal”): permute  $y$  after simulation  $\Rightarrow$  no signal.
- Dataset 2 (“signal”): as before,  $X_1, \dots, X_5$  important

Can BART detect that nothing’s going on in the “no signal” case? Are the two cases clearly different?

# Model selection with no model

Top row: posterior means for variable usage.

Bottom row: posterior 80% intervals for variable usage.



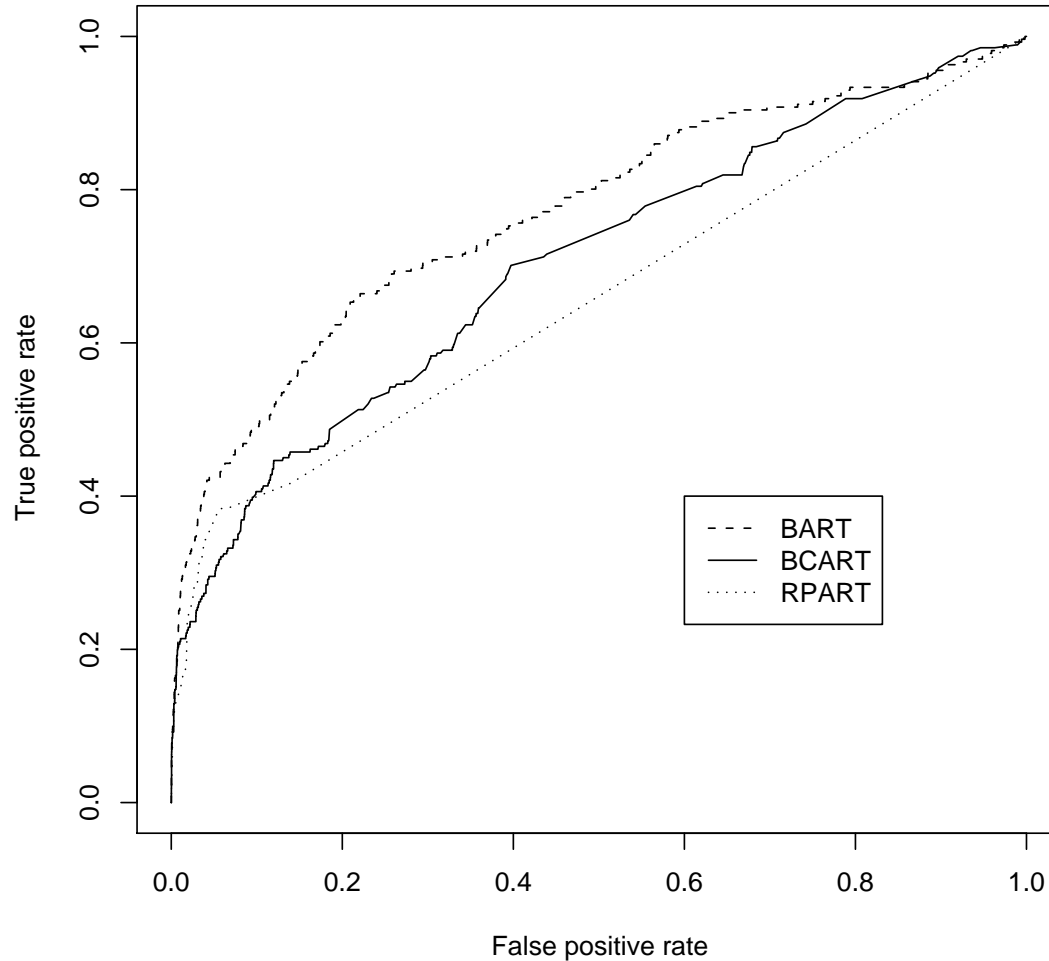
## Drug discovery application

- Each observation is a lab test of a potential drug against a biological target (e.g. AIDS virus)
- Binary response: 0= “inactive” , 1= “active”
- Predictors: 266 molecular characteristics of drug.
- Goal: find the “actives” ( $Y = 1$ )
- 29,374 drugs
- Only 542/29,374 drugs are active.

**BART modification for binary classification: probit regression via latent response (Albert & Chib 1993)**

# Drug discovery application

Out of sample ROC curves (true positives vs. false positives)



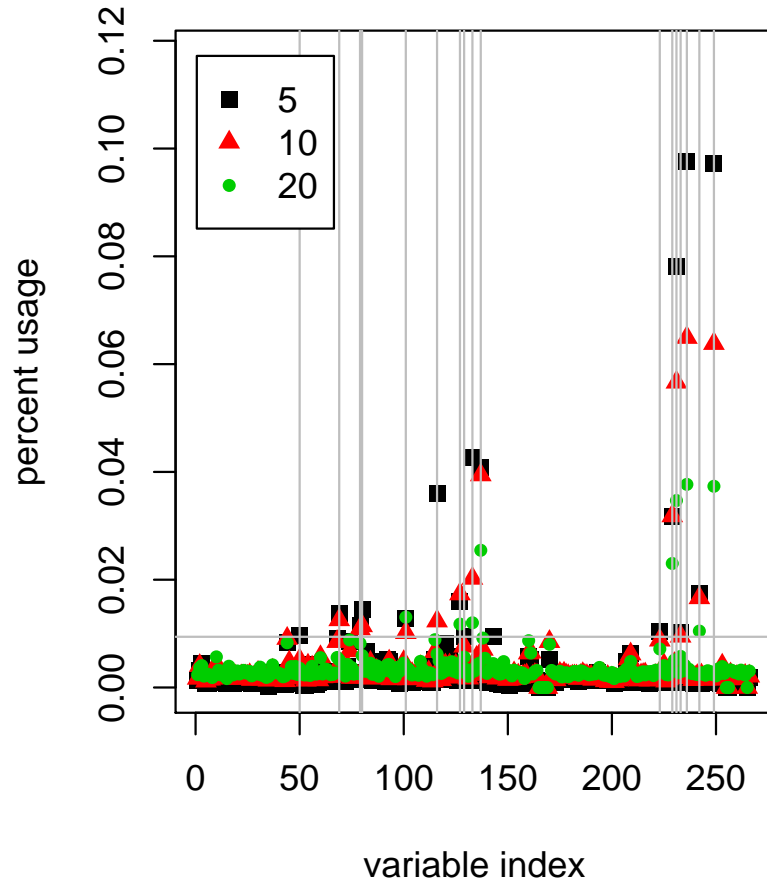
Area under curve

- BART = 0.77
- Bayesian CART = 0.70
- RPART = 0.66

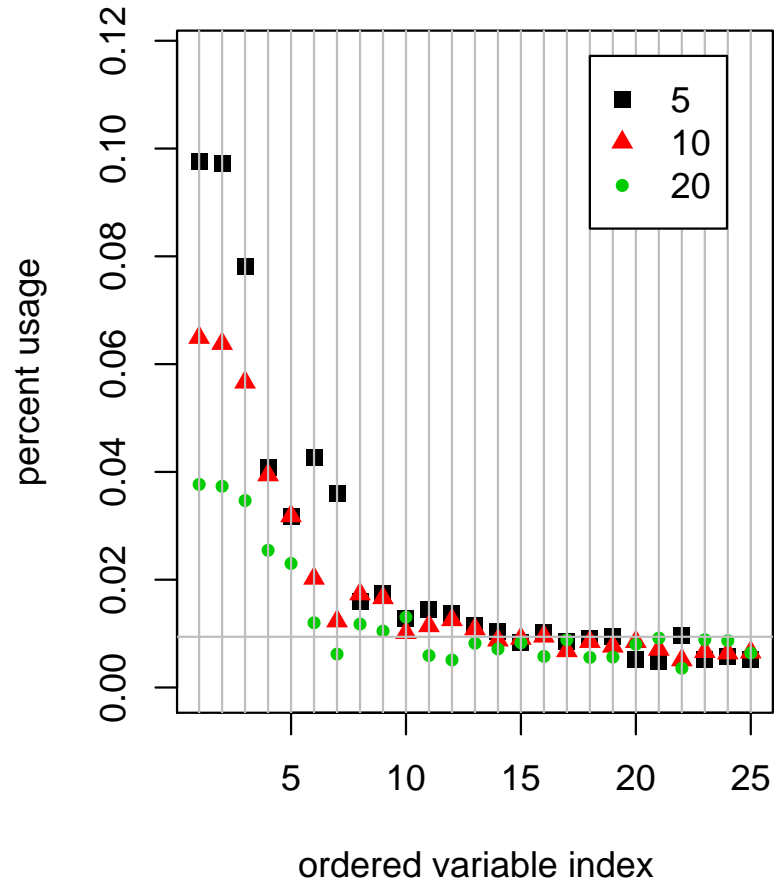
# Drug discovery application

## Variable selection via BART

all 266 variables



top 25 variables



## Another simulated example: aliased variables

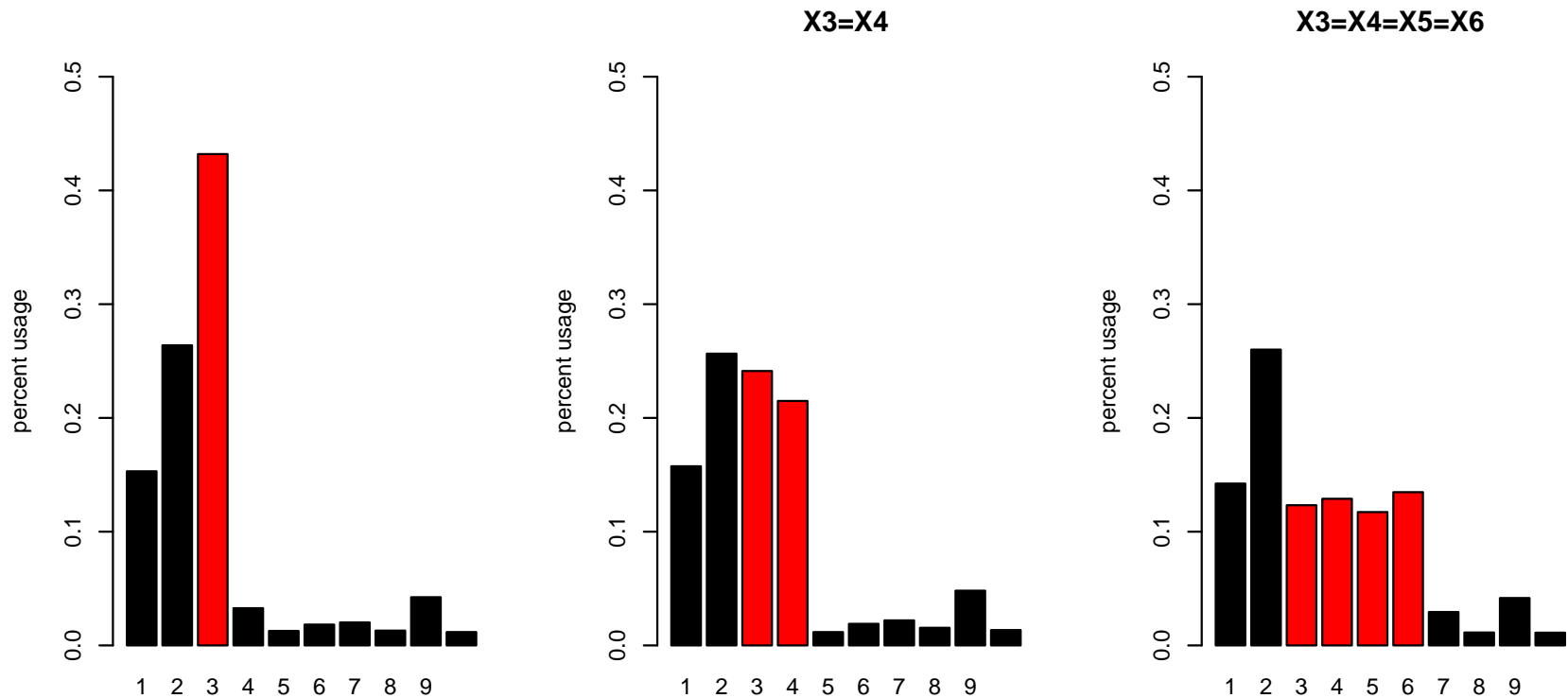
$$Y = X_1 + 2X_2 + 3X_3 + 0X_4 + \dots + 0X_{10} + \varepsilon, \quad \varepsilon \sim N(0, 1)$$

- $n = 100$  observations
- $p = 10$  predictors ( $X_4, \dots, X_{10}$  are junk variables)
- BART works well for variable selection.

But what if we replace one or more of  $X_4, \dots, X_{10}$  with exact copies of  $X_3$ ?

## Another simulated example: aliased variables

Left: no aliasing, Centre: aliased pair, Right: four identical variables.

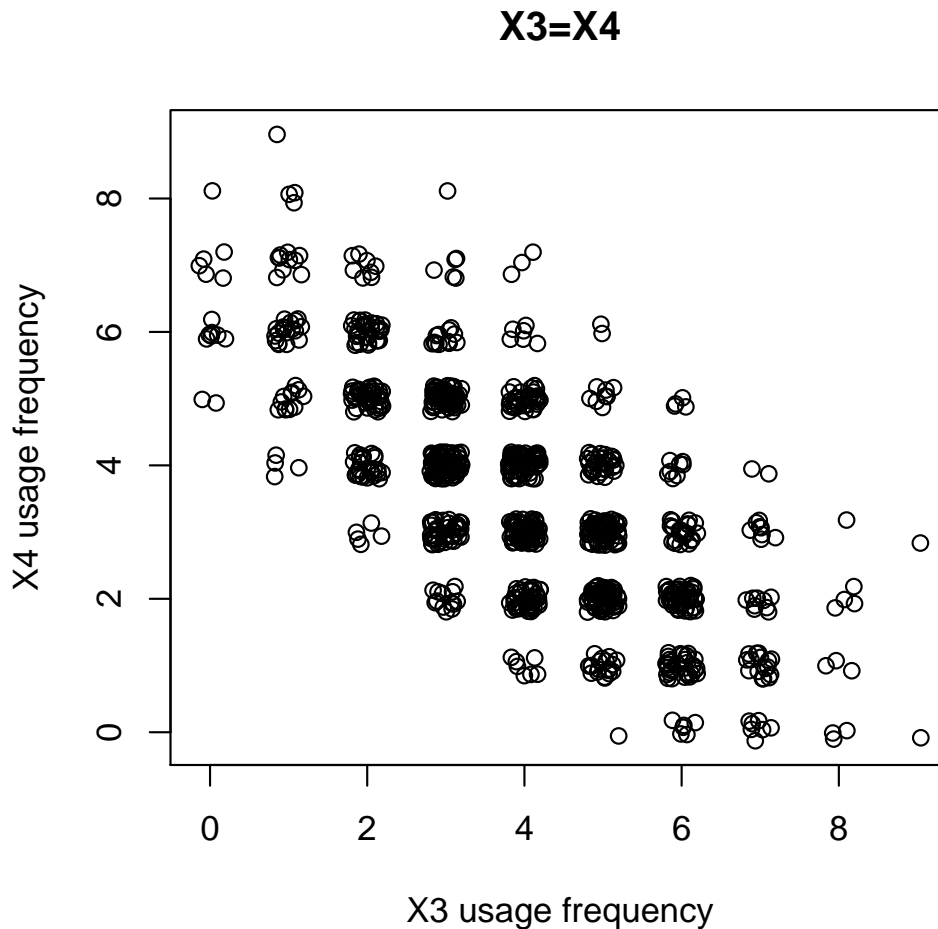


**“Dilution” Problem:** aliasing of  $X$ 's reduces “percent use”.

- Collinearity could produce a similar problem.

## Another simulated example: aliased variables

However, we can discover what's happening using the posterior.



- Aliased pair example ( $X_3 = X_4$ )
- 1000 posterior draws of “usage frequencies”
- Joint distribution of usage frequencies informative.
- Strong negative correlation (-0.80) between usage frequencies of  $X_3, X_4$

## Another simulated example: aliased variables

What about 4 aliased variables (i.e.  $X_3 = X_4 = X_5 = X_6$ )?

- Scatterplots and correlations don't save us.
- However, a PCA of the correlation matrix of posterior usage frequencies does:
  - The last PC has near-constant variance, and corresponds to the linear combination  $X_3 + X_4 + X_5 + X_6$ .

Side note: Convergence/mixing of variable selection indicators needs to be considered in many problems.

## Conclusions:

Take-home message: careful examination of the posterior distribution can yield useful information for variable selection.

Extensions are as easy as extending the original BART model.

- Binary response
- Nonparametric error distribution

And let's not forget: BART is...

- competitive for predictive accuracy,
- robust to prior specification,
- flexible,
- fast (no cv or bootstrap required),
- and can give valid statistical inference!

**THE END**

## Details:

1. Prior specification
2. MCMC sampling of the posterior

One important point. The model

$$Y = g(x, T_1, M_1) + g(x, T_2, M_2) + \dots + g(x, T_m, M_m) + \epsilon.$$

is different from Bayesian model averaging of a single tree model.

We are obtaining a posterior for a “sum of  $m$  trees” model, with a joint posterior on  $m$  trees and  $m$  terminal node parameter vectors.

## Prior specification:

Need to specify a prior on  $T$ 's,  $M$ 's, and  $\sigma$ .

- Assume prior structure:

$$\begin{aligned} p((T_1, M_1), (T_2, M_2), \dots, (T_m, M_m), \sigma) \\ = p(T_1, T_2, \dots, T_m) p(M_1, M_2, \dots, M_m | T_1, T_2, \dots, T_m) p(\sigma). \end{aligned}$$

- Since the dimension of the  $M$  depends on the  $T$ , this conditional structure is essential.
- We simplify even further by imposing independence whenever possible.

$$p(T_1, T_2, \dots, T_m) = \prod p(T_j)$$

$$p(M_1, M_2, \dots, M_m | T_1, T_2, \dots, T_m) = \prod p(M_j | T_j)$$

$$p(M_j | T_j) = \prod p(\mu_{i,j} | T_j)$$

## Prior Specification:

Semi-automatic choices motivated by Empirical Bayes methods.

Basic ideas:

- $T$ 's: how big a tree is probable?
- $\sigma$ : how much noise in the response?
- $M$ 's: How much can each individual tree contribute?  
Clever trick: Make this depend on the number of trees.
- Number of trees could be a parameter, but we fix it instead.

## Prior Specification:

### 1. Residual variance $\sigma^2$ :

Prior for  $\sigma$  is simplest and most important.

We use the standard conjugate prior:

$$\sigma^2 \sim \frac{\nu \lambda}{\chi^2_\nu}.$$

- $\nu$  determines spread of the prior
- $\lambda$  determines location of the prior

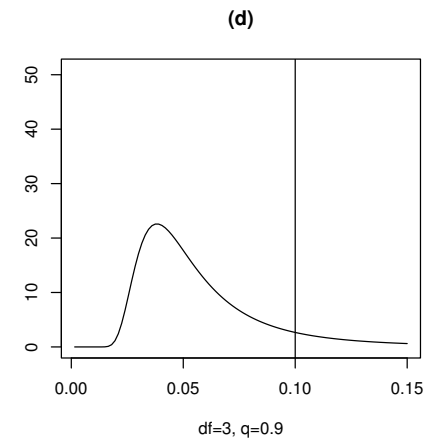
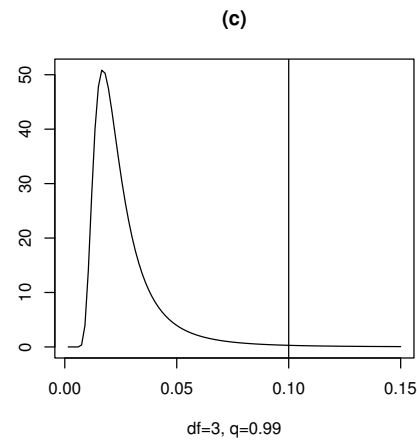
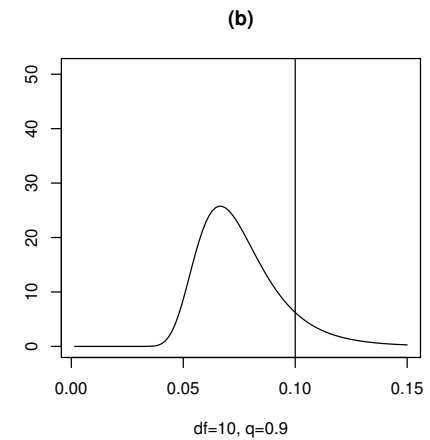
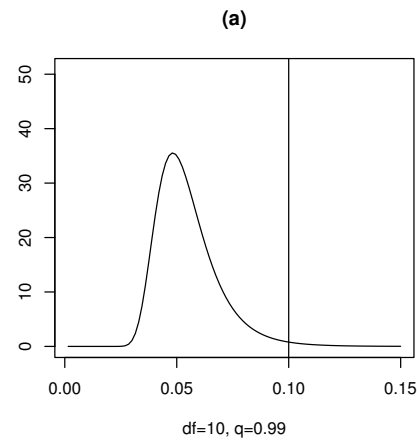
Instead of eliciting  $\nu, \lambda$  directly we:

- (a) Guess at an upper quantile of  $\sigma$ , say 90% or 99%. Set this equal to least squares linear regression estimate of  $\sigma$
- (b) Choose  $\nu$  to give good spread of  $\sigma$  prior.

# Prior Specification:

Vertical line indicates  $\hat{\sigma}$ , the rough estimate of  $\sigma$  from linear least squares model.

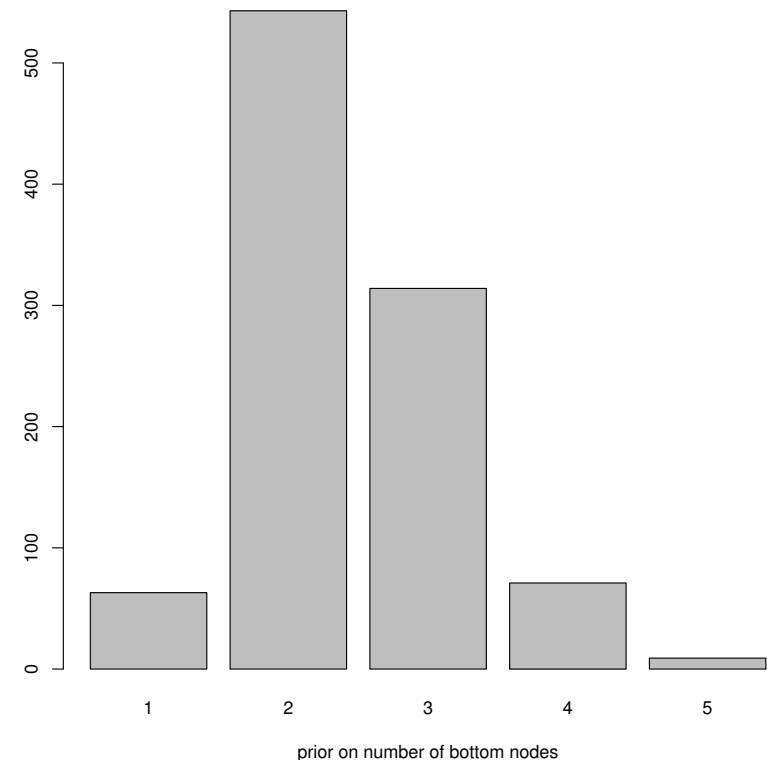
- Top:  $\nu = 10$
- Bottom:  $\nu = 3$
- Left:  $\hat{\sigma}$  at 99th quantile.
- Right:  $\hat{\sigma}$  at 90th quantile.



## Prior Specification:

### 2. Prior on tree structure $T$ :

- Basically a prior on tree size\*.
- Actual prior used in examples later gives tree size prior in plot.
- **NOTE** that unlike Boosting, we don't fix tree depth. We put a prior on it and let the data determine tree depth.
- Tree size determines how many variables are used in each "weak learner"  $g(x; T, M)$ .



\* Actually a distribution on whether you split, which variable you split on, and the splitting rule, for each node.

## Prior Specification:

3. Terminal node parameters  $\mu_i$

- Suppose we have  $m = 200$  trees.
- For any  $x$ , the prediction  $f(x)$  will be a sum of 200  $\mu$ 's, one from each tree.

$$\theta = E(Y|x) = f(x) = \sum_{i=1}^{200} \mu_i$$

$$\text{Var}(\theta) = \sum_{i=1}^{200} \text{Var}(\mu_i) = 200\text{Var}(\mu_i)$$

(assuming  $\mu_i$ 's independent given the trees)

- So, we can specify how much we expect the mean of  $y$  given  $x$  (i.e.,  $\theta$ ) to vary, and take

$$\text{Var}(\mu_i) = \text{Var}(\theta)/200$$

$$\text{sd}(\mu_i) = \text{sd}(\theta)/\sqrt{200}$$

## Prior Specification:

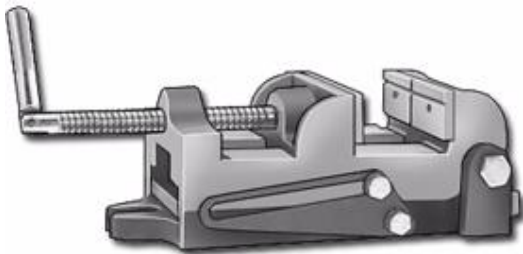
### 3. Terminal node parameters $\mu_i$

- Assume  $\mu_i$  is normally distributed, with mean 0, and standard deviation

$$\text{sd}(\mu_i) = \frac{\text{range}(Y)}{4\sqrt{200}}, \quad \text{for 200 trees in sum}$$

NOTE: The amount of shrinkage of the  $\mu$ 's *depends on the number of trees* (here  $m = 200$ ).

- Each term  $g(x;T,M)$  will be “regularized” so it contributes only a small part of the overall fit.



So the model is adaptively regularized in several ways: Tree prior and terminal node prior.

## MCMC Estimation:

Instead of explicitly maximizing the posterior, we simulate from it, via Markov chain Monte Carlo (MCMC).

In a nutshell:

Let  $T_{(-j)}$  be all trees *except*  $T_j$ , define  $M_{(-j)}$  similarly.

Repeat  $k = 1, \dots, 1000$  (say)

- Repeat  $j = 1, \dots, m$  times
  - Metropolis-Hastings step: Draw  $T_j$  conditional on  $Y, T_{(-j)}, \sigma$
  - Draw  $M_j$  given  $Y, T_1, \dots, T_m, M_{(-j)}, \sigma$
- Draw  $\sigma$  given  $Y$  and all other parameters.

Note that the sample of  $T_j$  at step  $k$  is actually a modification of the  $T_j$  sample at step  $k - 1$ .

## MCMC Estimation:

### Final prediction:

- Each sweep of algorithm yields a draw from the posterior of

$$f(x) = g(x, T_1, M_1) + g(x, T_2, M_2) + \dots + g(x, T_m, M_m)$$

- Average the draws - gives the posterior average of  $f(x)$ .
- Uncertainty in  $f(x)$  is also available, from the posterior distribution on  $f(x)$ .

### Connections to other learning algorithms:

1. Bayesian Backfitting (Hastie and Tibshirani) is a similar MCMC approach.
2. Like Boosting, each of our “weak learners”  $g(x; T_j, M_j)$  learns structure that the other weak learners do not capture.
3. Like Random Forests and Bagging, we model average over multiple draws of the sum of trees model.

## **Part 4: Examples with Data (Simulated, Boston, Active Learning)**

## Simulated example: Friedman (1991)

$$y = f(x) + \epsilon, \quad \epsilon \sim N(0, 1)$$

where

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

(10 x's but only the first 5 matter)

$n = 100$  observations

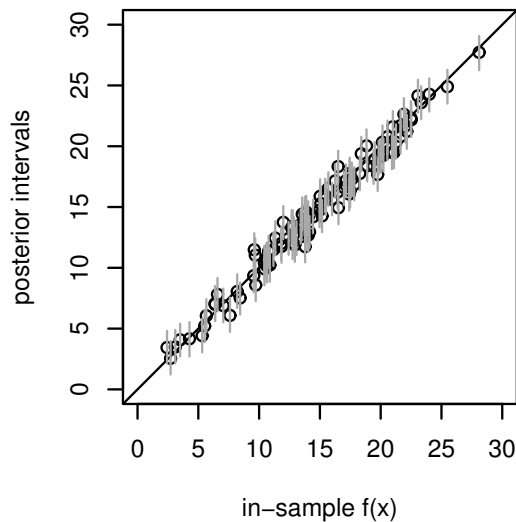
### BART settings:

- $m = 100$  trees
- $\sigma$  prior uses  $\hat{\sigma}$  from linear least squares regression as 90th quantile,  $\nu = 3$ .
- Tree prior puts most probability on 2, 3 terminal nodes.
- Automatic choice of  $M = \{\mu_j\}$  prior just discussed.

# Simulated example:

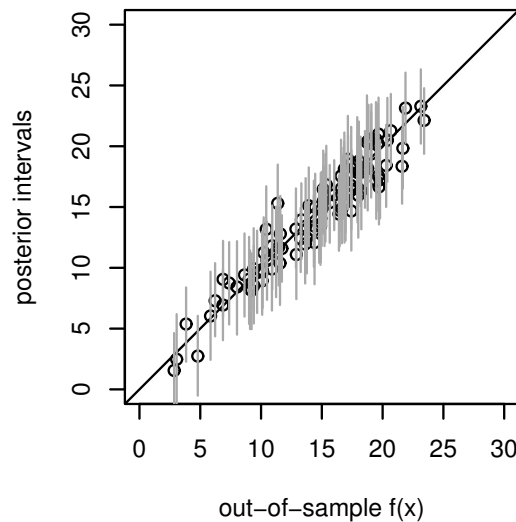
Training sample predictions

(a)



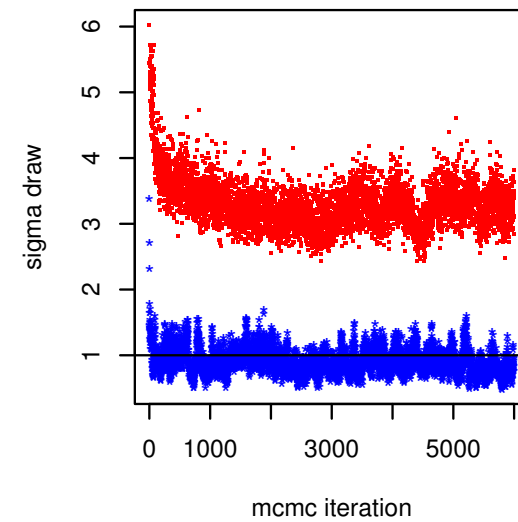
Test sample predictions

(b)



posterior for  $\sigma$   
(red= 1 tree,  
blue=ensemble)

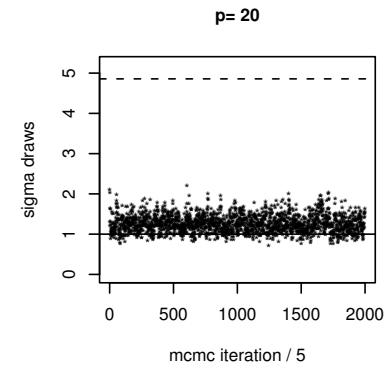
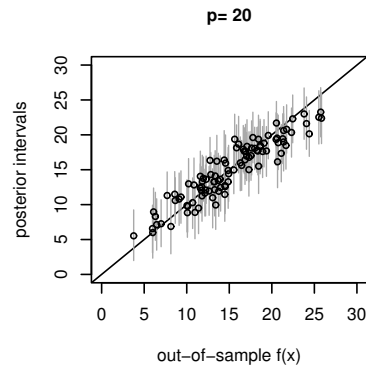
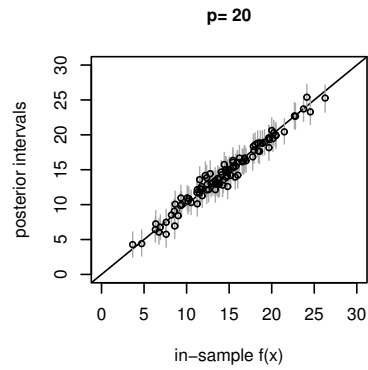
(c)



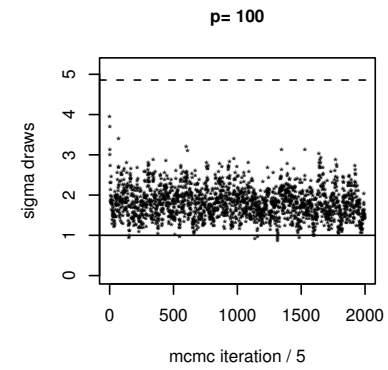
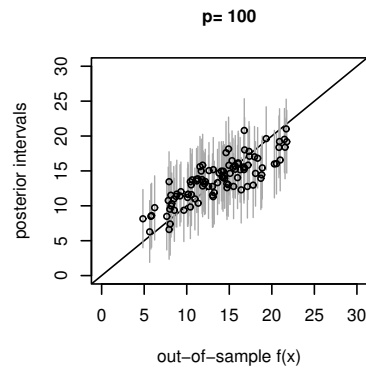
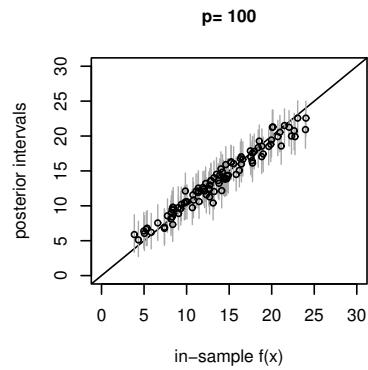
- Chain converges quickly and mixes well.
- Note that the model is not identifiable, but we are really only interested in identifiability of predictions.

# Simulated example:

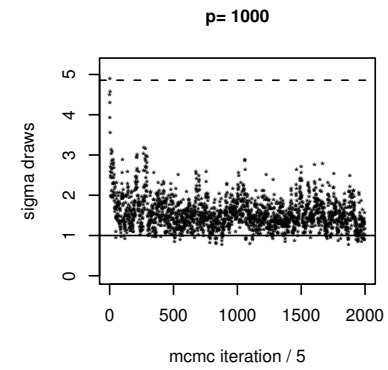
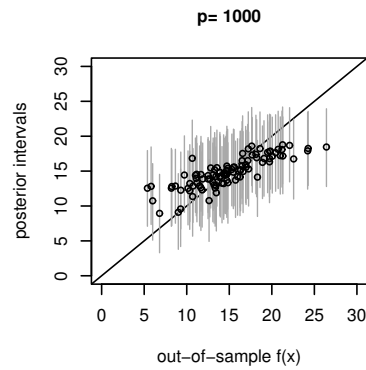
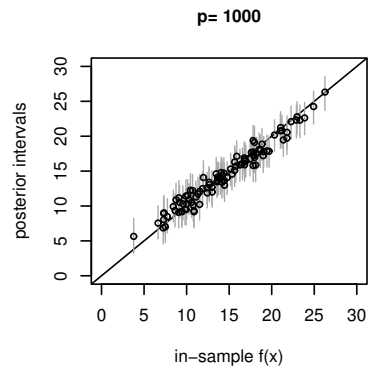
$p = 20$   
dimensions



$p = 100$   
dimensions



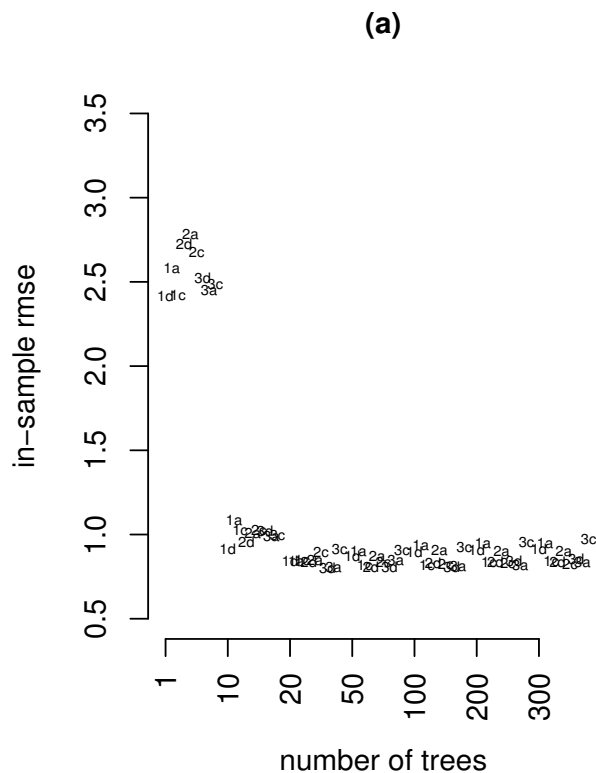
$p = 1000$   
dimensions



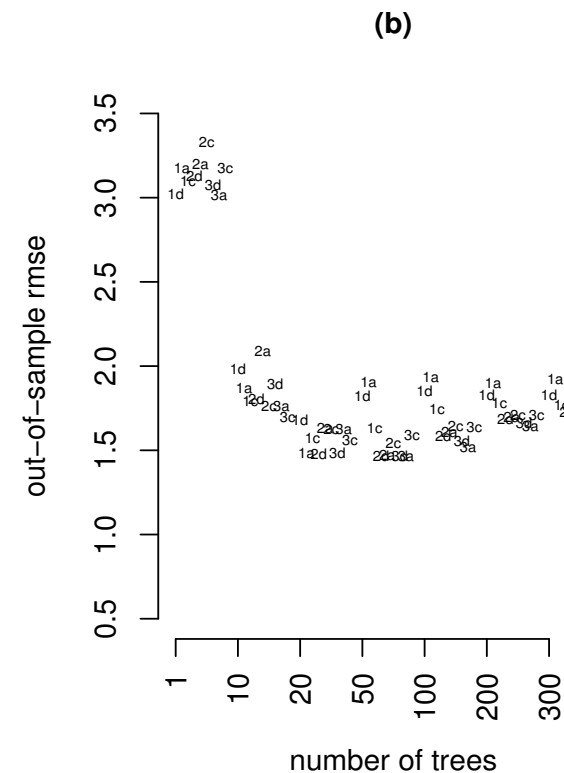
## Simulated example:

Previous page: BART capable of extracting low-dimensional signal with many  $x$ 's. (Even  $n \ll p$ , i.e.  $n = 100$  observations in  $p = 1000$  dimensions!)

Also reasonable robustness to prior settings:



Training Results



Test Results

## Additional Goodies: The Boston Housing Example

- Goal: Predict neighbourhood house price using demographic variables.
- Data:
  - $y = \log$  median house price in the region (the response)
  - $X$  is 13 predictors, measuring pollution, crime, house sizes, commute distance, racial diversity, tax rates, etc.
- Common “benchmark” problem.

## Additional Goodies: The Boston Housing Example

Posterior distribution on the number of terminal nodes of the 200 trees (actually a draw from the posterior).

Number nodes	1	2	3	4	5	6
Relative freq (%)	3.5	54.5	32.5	8.5	0.5	0.5

This can be interesting because

- a 1-node tree doesn't contribute to the model
- a 2 node tree is a main effect for one variable
- a 3 node tree is a two-way interaction
- ... etc.

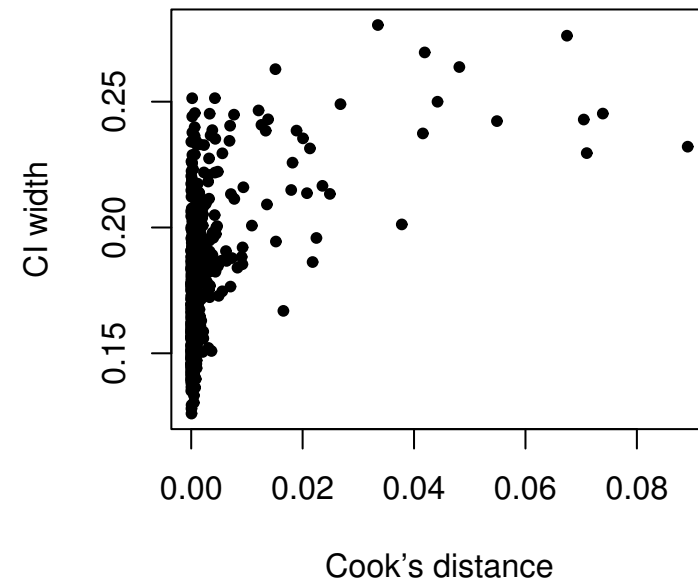
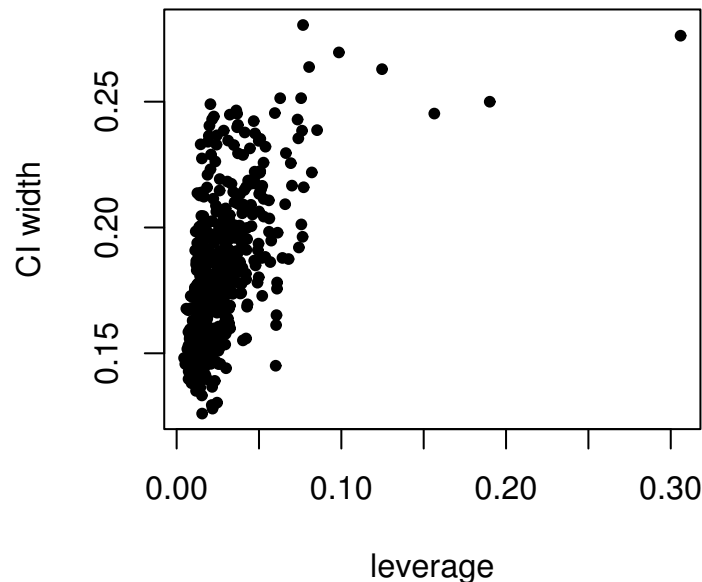
In this case, there seems to be mostly main effects and some two-way interactions.

(still a somewhat dodgy way to measure interaction order)

## Additional Goodies: The Boston Housing Example

### Relation to model diagnostics

- Consider predicting  $y$ . For each point, plot the posterior interval width against traditional regression diagnostics (left: leverage, right: Cook's distance).
- Influential points tend to have larger posterior intervals.
- Posterior gives information about influential points.



## Summary and future work:

1. It is possible to have a flexible predictive model, but still use it to make statistical inferences.
  - There is some computational cost.
  - Some derivations of models necessary.
  - But it's worth it: Cross-validation not necessary.
2. Extension to classification: 2-class problem is immediate: view binary outcome as corresponding to a latent continuous variable.
3. We plan to do extension to exponential family (similarities with Hastie and Tibshirani's Bayesian Backfitting).
4. Because we have a probability model, we can build in many interesting features. (e.g., different response data types, hierarchical models, outliers, modelling of  $\sigma$  as well as  $\mu, \dots$ )