

# Statistical Learning With Trees

Hugh Chipman, Acadia University

joint with

Edward George, University of Pennsylvania and  
Robert McCulloch, University of Texas

R package BayesTree available on CRAN



*Mathematics and Statistics at Acadia*

FACULTY OF PURE AND APPLIED SCIENCES



# Outline

1. Trees as statistical models
  - (a) One tree
  - (b) Many trees
2. What can you do with a forest of trees?
  - (a) Variable selection
  - (b) Sequential design
  - (c) Nonparametric error model

## A flexible 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  $\varepsilon$

We'll consider two models for  $f$ :

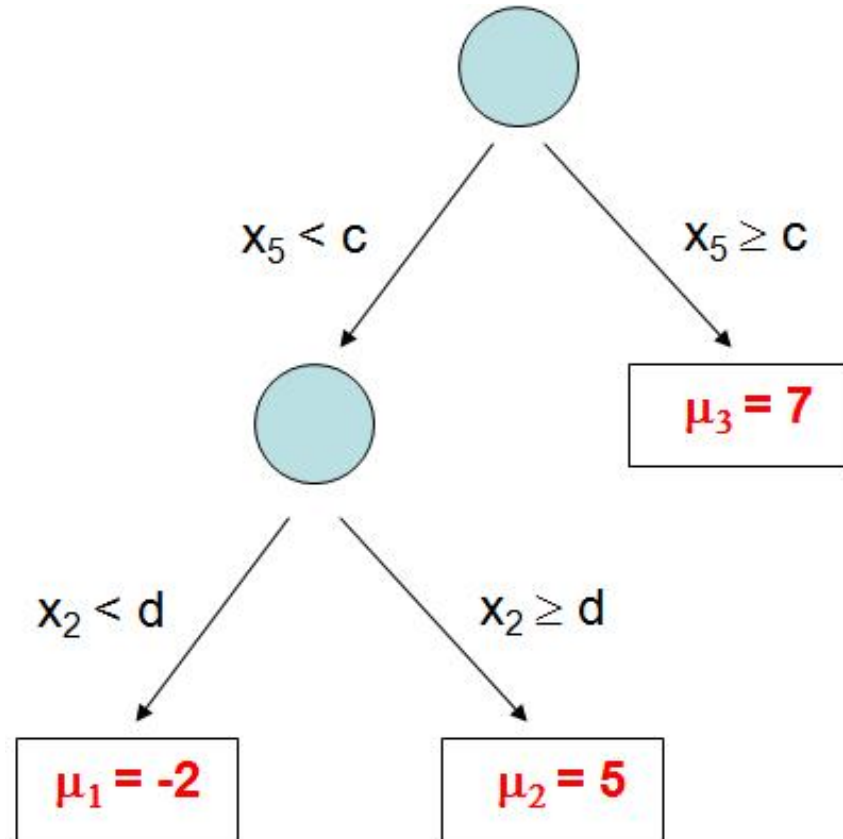
1. A single tree (or recursive partitioning model)
2. A sum of trees

## A tree model for $f(x)$

(CGM JASA 1998, Denison, Mallick & Smith *Biometrika* 1998)

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

## Tree models: Algorithms for learning



Like making sausages, you may not want to know what goes into it.

- Greedy growing algorithm recursively constructs a tree one node at a time
- Prune a large tree back to a smaller size.
- Tree size often determined by **cross-validation**,
  - build a model with part of the data
  - predict for the remainder, giving a believable estimate of accuracy.
  - **nonparametric substitute for formal inference**

# Trees as statistical models:

Parameter

Everything's a

Ciampi (1991) an early proponent...

Remember...

$$y = g(x; T, M) + \sigma Z ?$$



## Trees as statistical models:

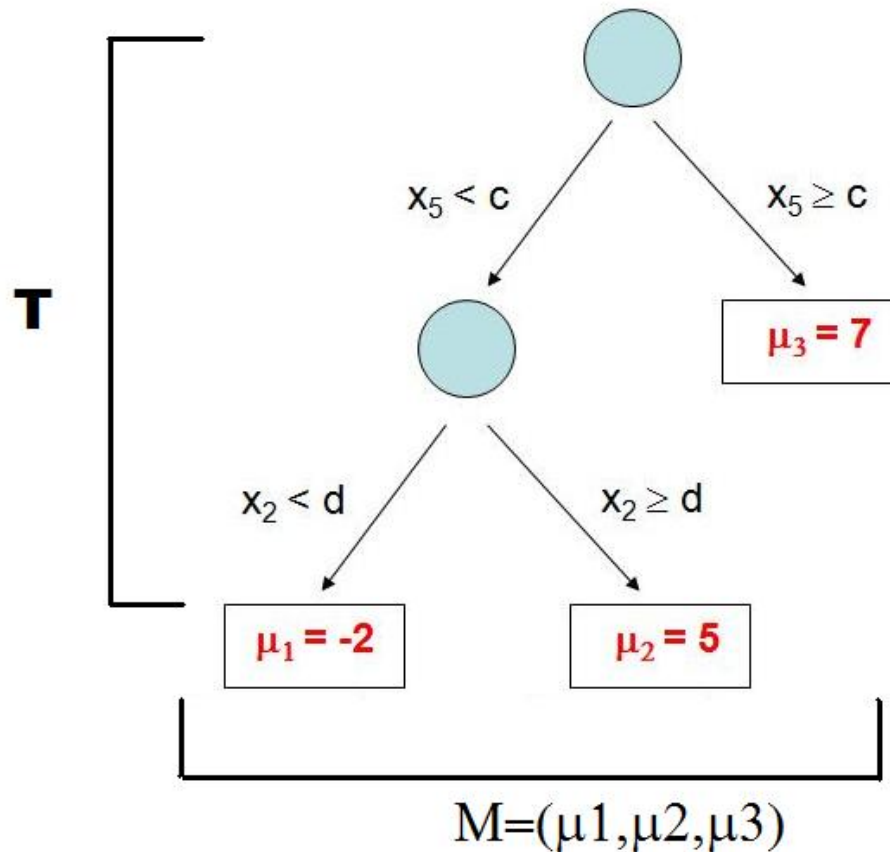
Main ingredients:

1. Prior distributions
2. Posterior computation

## Prior specification:

We factor the joint prior on  $T, M$  as

$$P(T, M) = P(T)P(M|T)$$

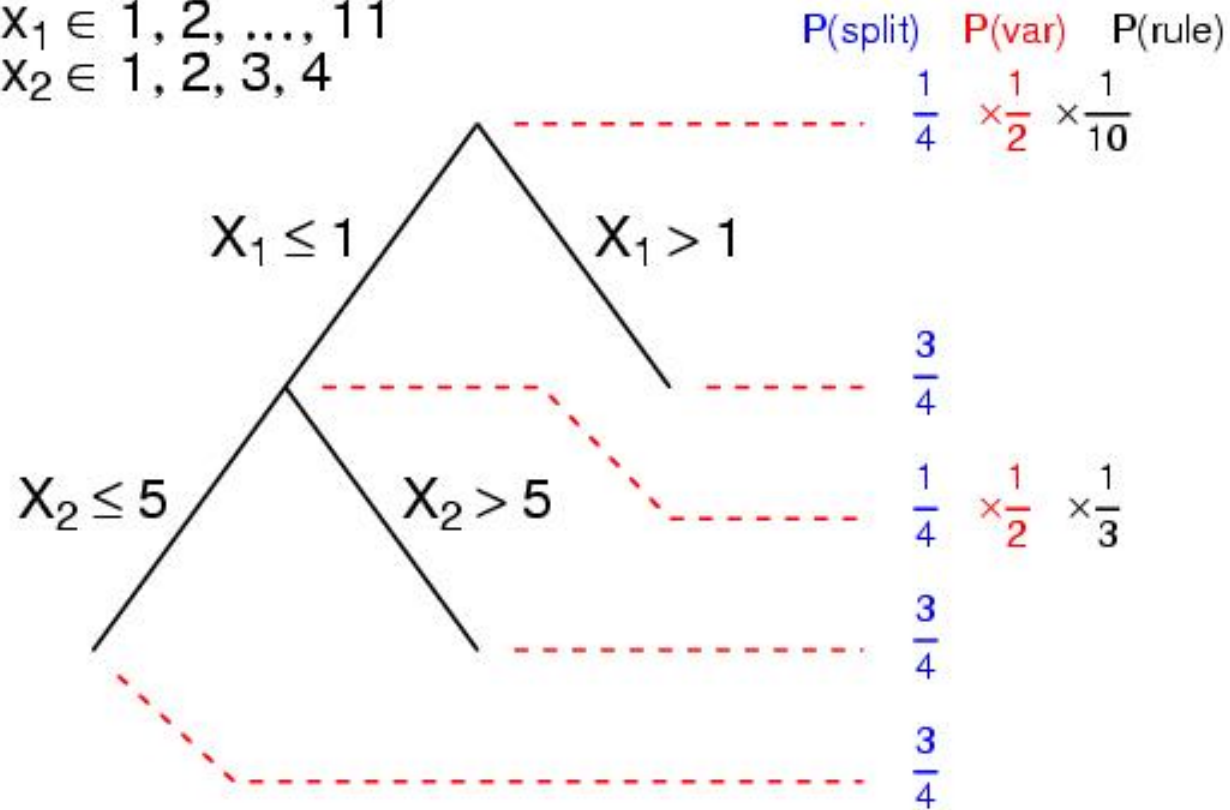


## Prior $P(M|T)$

- Form of prior depends on parametric model in terminal nodes.
- Careful choice (conjugacy) can enable analytic simplification of  $P(T, M|Y)$  to  $P(T|Y)$ .

# Tree prior: $P(T)$

$X_1 \in 1, 2, \dots, 11$   
 $X_2 \in 1, 2, 3, 4$



$$P(\text{Tree}) = \left(\frac{1}{4} \times \frac{1}{2} \times \frac{1}{10}\right) \times \frac{3}{4} \times \left(\frac{1}{4} \times \frac{1}{2} \times \frac{1}{3}\right) \times \frac{3}{4} \times \frac{3}{4}$$

## Computation: calculating the posterior

We need to evaluate

$$P(T, M|Y) \propto P(M|T, Y)P(T|Y)$$

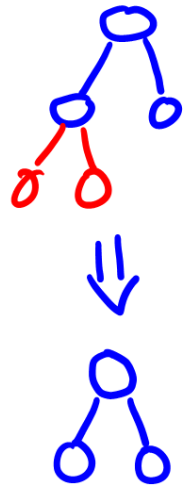
- $P(M|T, Y)$ , the posterior on  $M$  for a specific tree  $T$ , is manageable.
- The challenge is identifying trees ( $T$ 's) with high posterior probability.
  - Markov chain Monte Carlo (Metropolis-Hastings) enables us to simulate from the posterior  $P(T|Y)$ .

# Computation: calculating the posterior

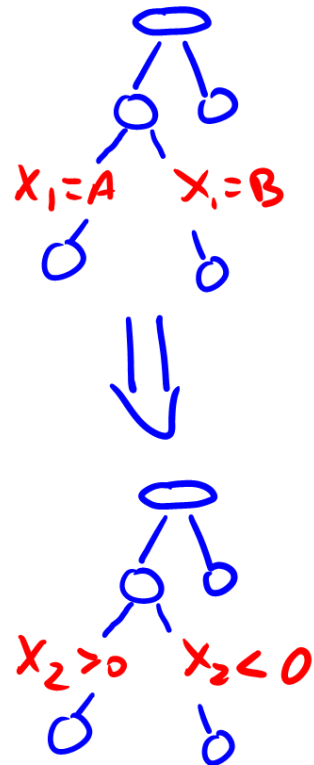
GROW



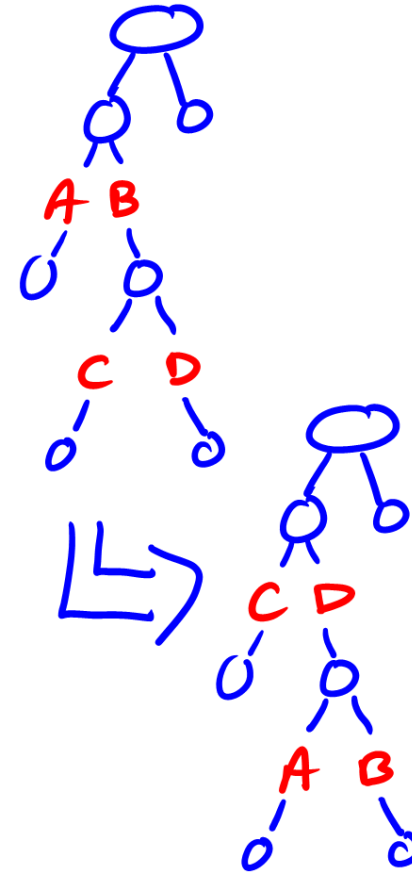
PRUNE



CHANGE



SWAP



Wu, Tjelmeland and West (2007) give fancier steps.

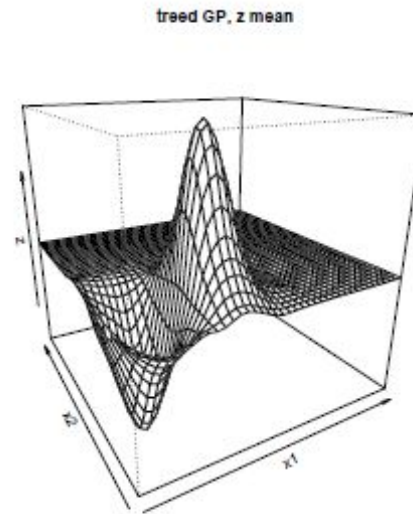
## Does it work?

Sometimes.

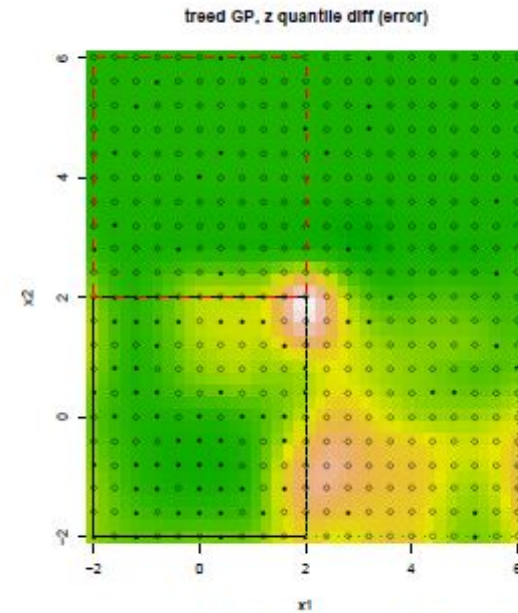
- A piecewise constant model ( $M = (\mu_1, \dots, \mu_b) =$  terminal node  $E(Y|X)$ 's) isn't always the most accurate.
  - but Bayesian model averaging helps (like Breiman's Random Forests)
- Trees can be interpretable.
- Extensions to richer terminal node models are especially useful and interpretable...
  - Linear regression or GLM in each terminal node (CGM *Machine Learning* 2002; Valencia 7, 2003)
  - Gaussian process model in each terminal node (Gramacy & Lee, 2008 *JASA*)

# Example: Treed Gaussian Process model

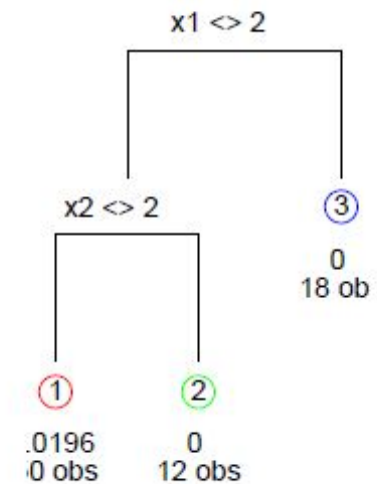
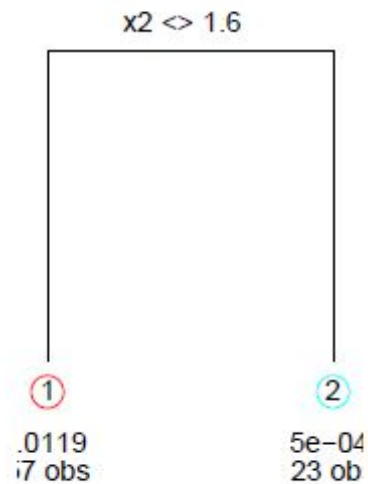
(Gramacy and Lee  
2008)



height=2, log(p)=204.743



height=3, log(p)=259.932



# Many Trees

## A problem with trees

- If

$$y = f(x) = x_1 + x_2 + x_3 + x_4 + x_5 + \varepsilon,$$

we need at least a 32-node tree (depth 5) to capture this structure (first split on  $X_1$ , then  $X_2$ , etc).

- A tree's strength is in interactions, but weakness is additivity.

## A solution: “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

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

$$Z \sim N(0, 1).$$

- 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 mean response  $\bar{Y}$ .
  - 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  $m$ .
- BART adaptively selects the basis as part of model-fitting.

# 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

## Predictive accuracy results

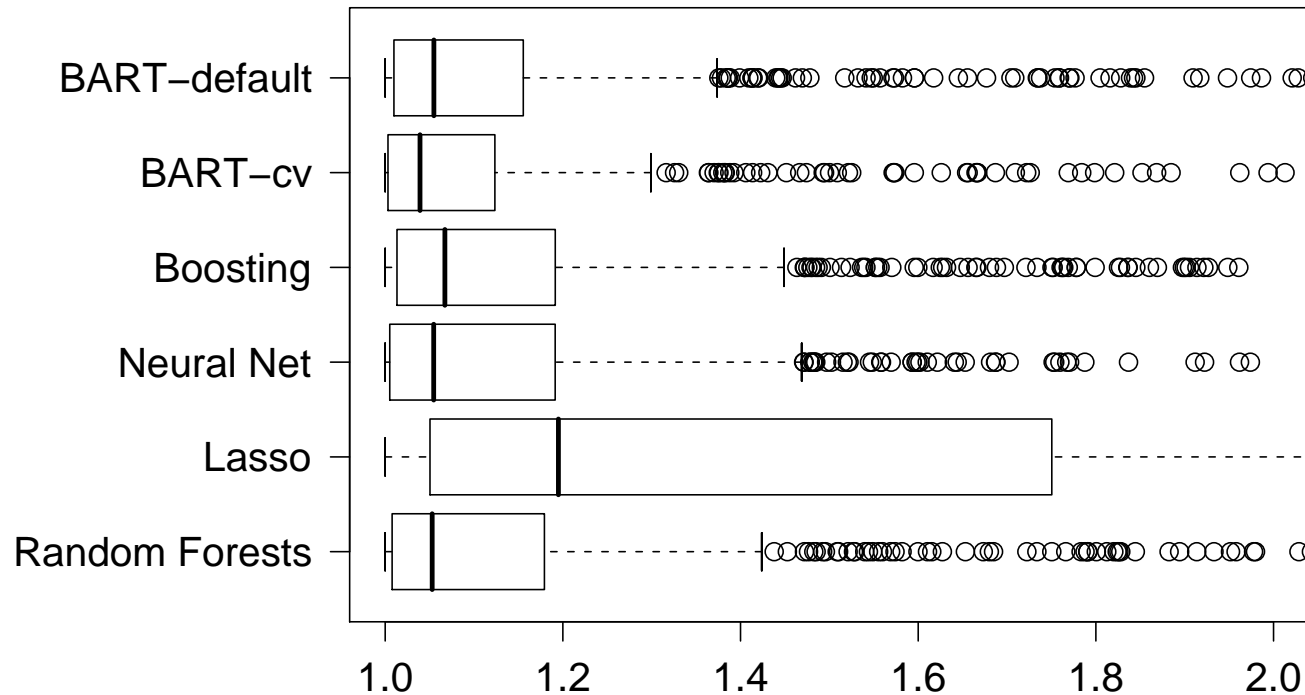
RMSE across 42 datasets (after standardizing  $Y$ ):

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

**Relative RMSE:** for each replicate on each data set, we identify best model, and all RMSEs are divided by the RMSE of the best model.

⇒ “1.0” is best, “2.0” is a RMSE twice as large as best model.

## Results: Relative Root Mean Square Errors



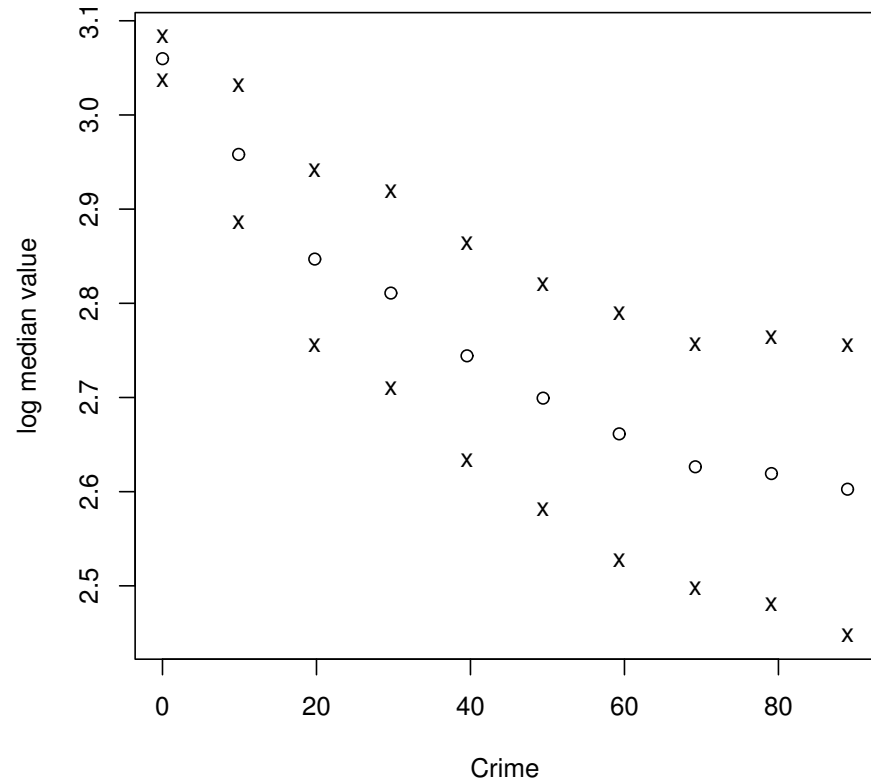
### Some comments:

- BART does quite well
- Cross-validated BART (BART-cv) is a modest improvement.
- It's surprising how close the different models are.

## 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 part II ...

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

All the extensions in part II...

1. variable selection,
  2. sequential design, and
  3. nonparametric error distributions
- ... aren't effective if your model has the wrong functional form .

## 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 **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

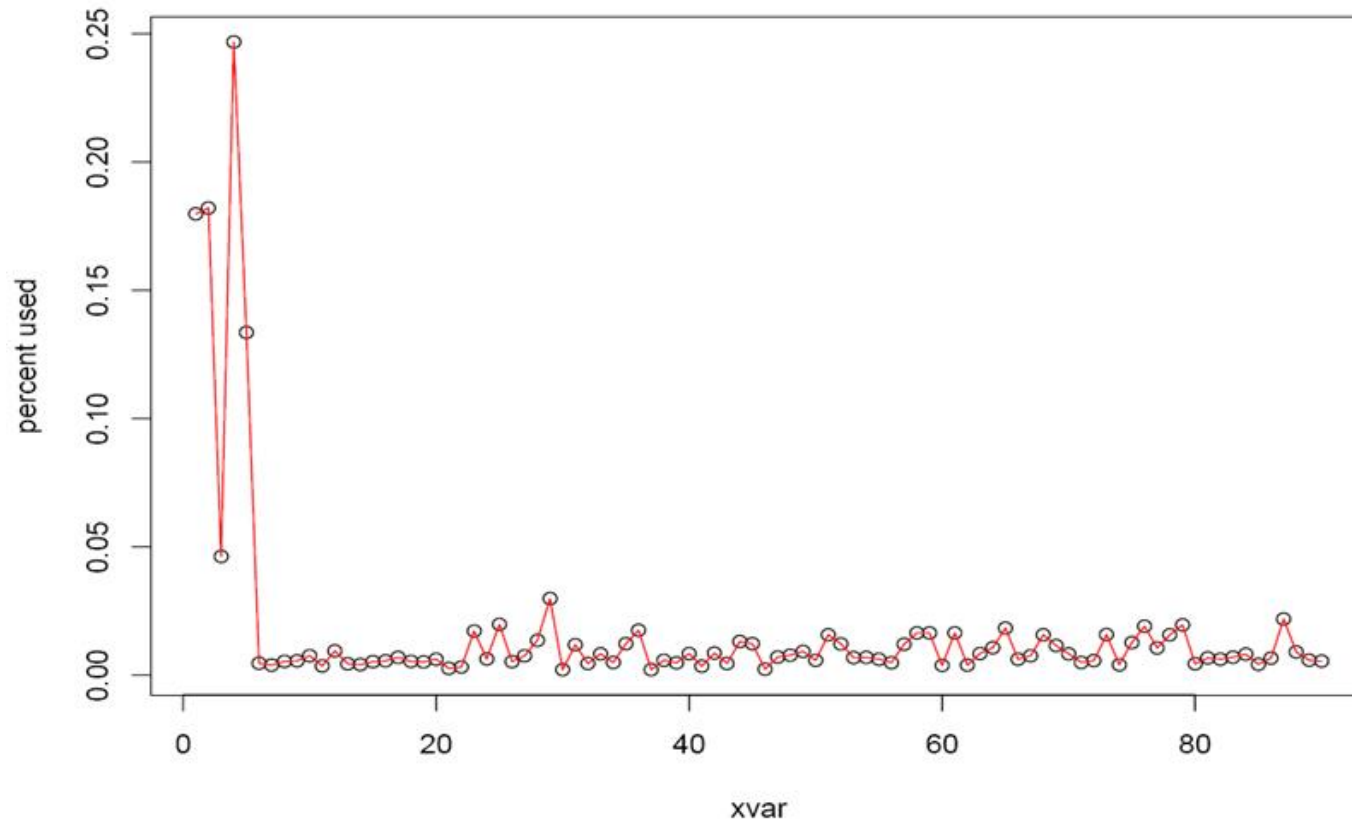
$$f(x) = 10 \sin(\pi x_1 x_2) + 20(x_3 - .5)^2 + 10x_4 + 5x_5 + 0x_6 + \dots + 0x_{10}$$

- 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

BART remains effective with  $n = 100$  and  $p = 90$   $X$ 's.



## Another simulated example: aliased variables

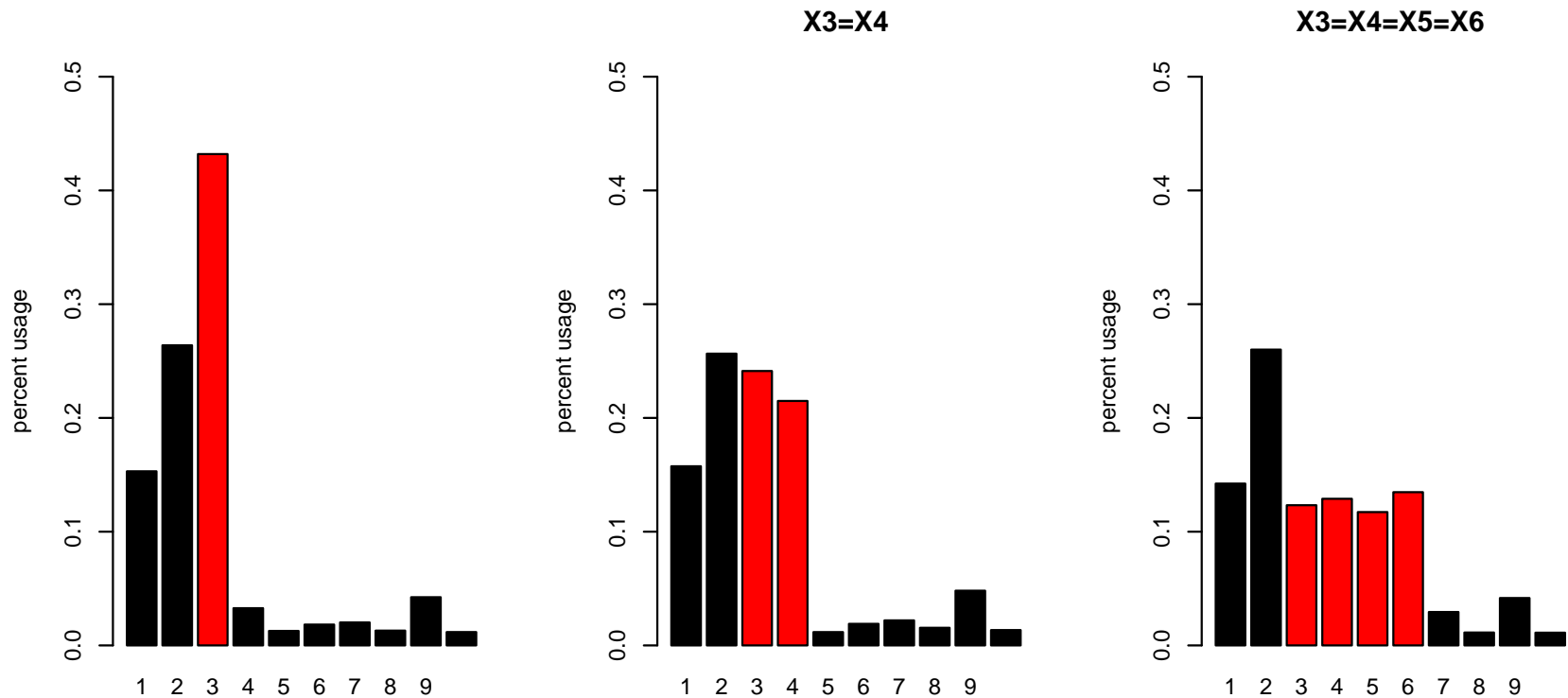
$$Y = X_1 + 2X_2 + 3X_3 + 0X_4 + \dots + 0X_{10} + \varepsilon, \quad \varepsilon \sim N(0, 1)$$

- $n = 100$  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.

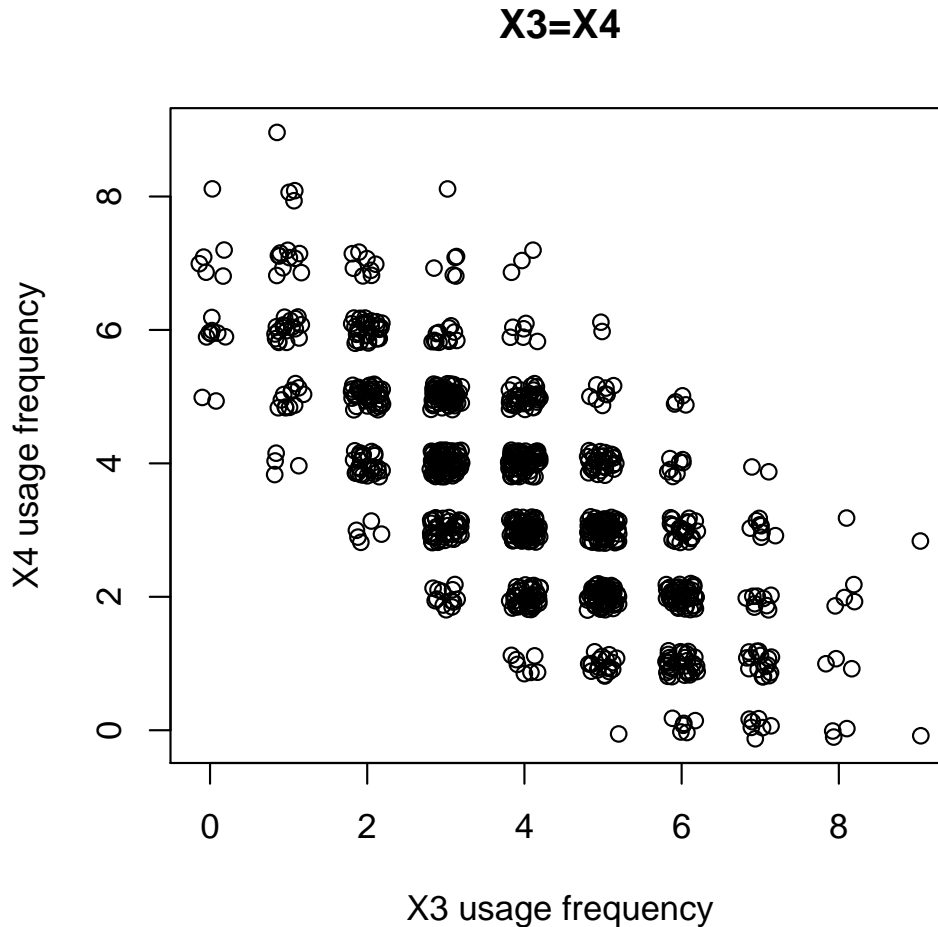


**“Dilution” Problem:** aliasing of  $X$ 's reduces “percent use”.

- 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  $X_3 + X_4 + X_5 + X_6$ .

# Sequential Design

The game we play:

- Same “regression” scenario as before: predict  $Y$  using  $X$ .
- The difference is that **we can sequentially choose the  $x$ 's at which we measure  $Y$ .**
- By “actively learning” (ie sequentially gathering data) we hope to build a better model with less data.
- Main idea: re-build the model as data are collected, and use it to decide where to sample more data.

# Sequential Design

Using the model to decide what data to collect:

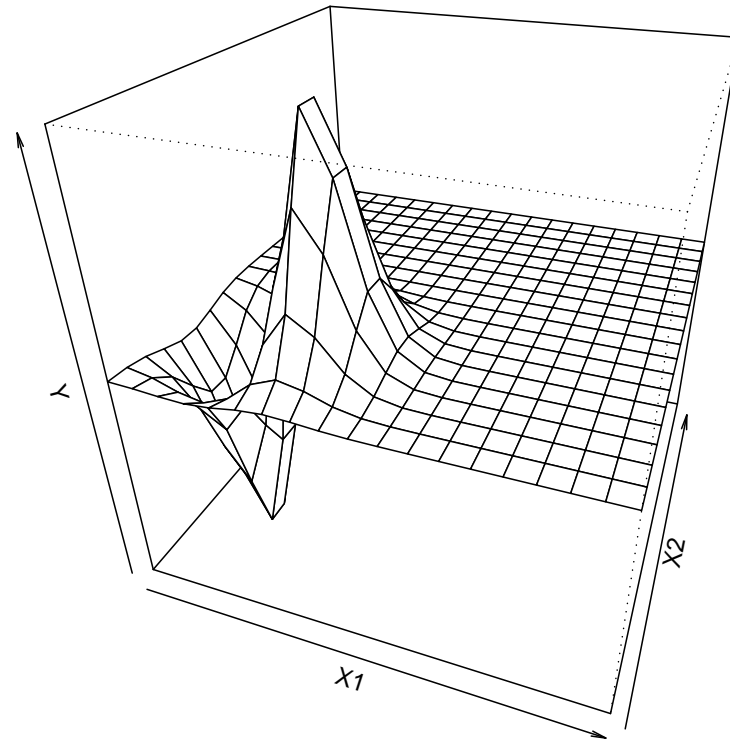
Two possible criteria:

1. Maximize variance of predicted response (“ALM” - MacKay (1992))  
(*where do I know the least about  $Y$ ?*).
2. Maximize expected reduction in variance of predicted response, averaged over a candidate set  $C$  (“ALC” - Cohn (1996))  
(*what data point will improve my model's predictions most?*)

We'll use # 1 here

## 2-D example (Gramacy and Lee 2008)

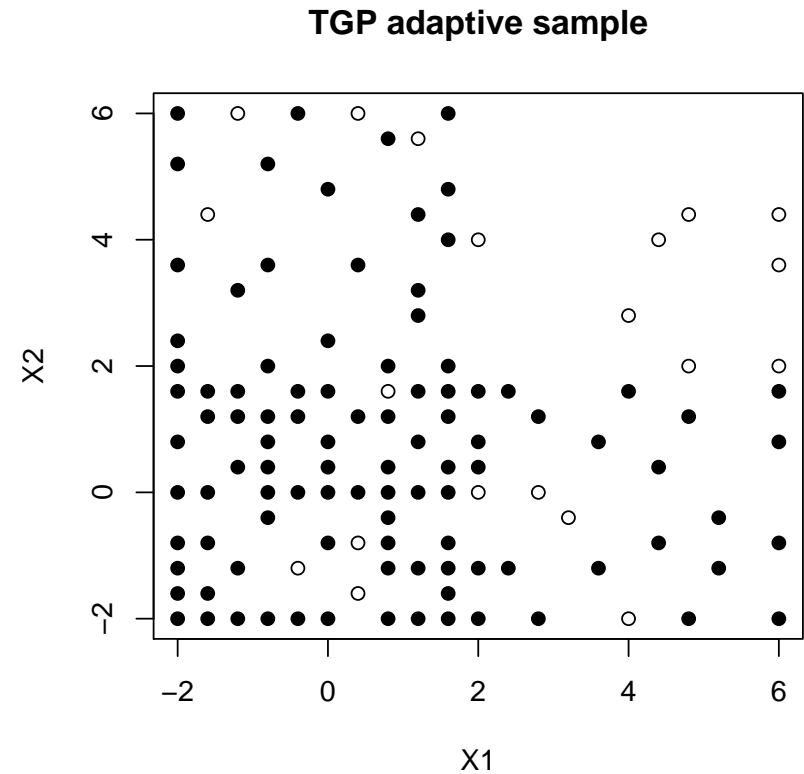
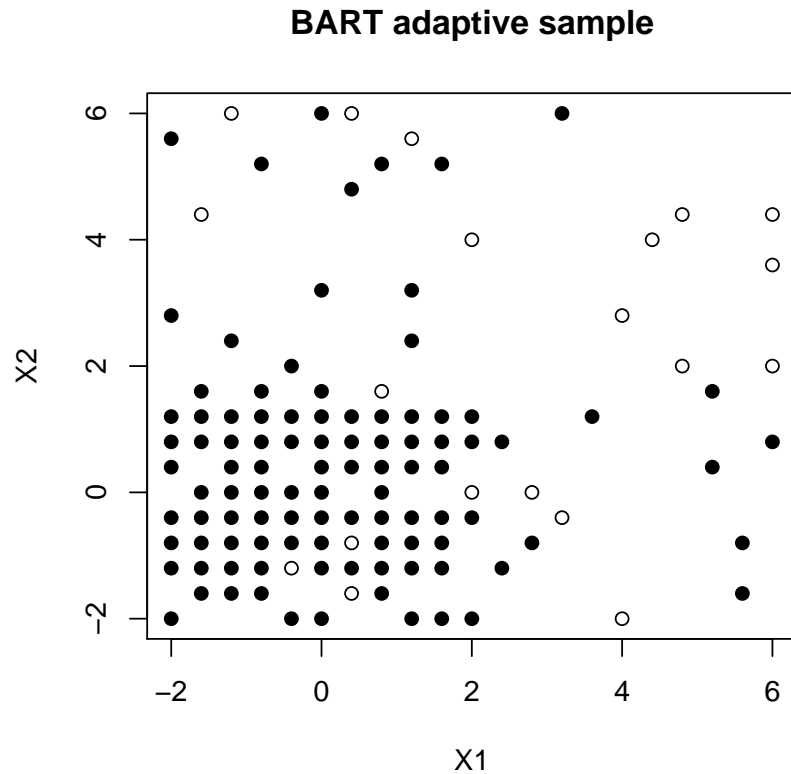
- Two predictors ( $X_1, X_2$ )
- Function (right) nearly constant in 75% of input space
- Initial random sample of 20 observations, followed by adaptive sampling of 100 observations.



We'll make comparisons with Gramacy and Lee's "Treed Gaussian Processes" (TGP)

## 2-D example (Gramacy and Lee 2006)

Points sampled by BART and TGP:



Test-set MSE's (right) indicate

- Both select good samples
- TGP fits better (smooth)

MSE	Model	
	TGP	BART
SRS sample	3.66	15.96
BART sample	0.35	2.84
TGP sample	0.40	2.93

## Nonparametric error distributions:

“Basic” BART model:

$$Y_i = \sum_{j=1}^m g(x_i; T_j, M_j) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma)$$

**Problem:** iid Normal assumption may be too restrictive.

Replace  $\epsilon_i \sim N(0, \sigma)$  assumption with:

$$\epsilon_i \sim N(0, \sigma_i)$$

$$\sigma_i \sim G$$

$$G \sim DP(\alpha, G_0)$$

where  $DP$  is the Dirichlet Process.

Right now,  $\alpha$  is fixed at a reasonable value given  $n$  and  $G_0$  is the same prior we used for the default  $\sigma$  prior. (See Escobar & West 1998)

## Friedman example:

$$y_i = f(x_i) + \epsilon, \quad \epsilon \sim N(0, \sigma_i)$$

where

$$f(x) = 10 \sin(\pi x_1 x_2) + 20(x_3 - .5)^2 + 10x_4 + 5x_5$$

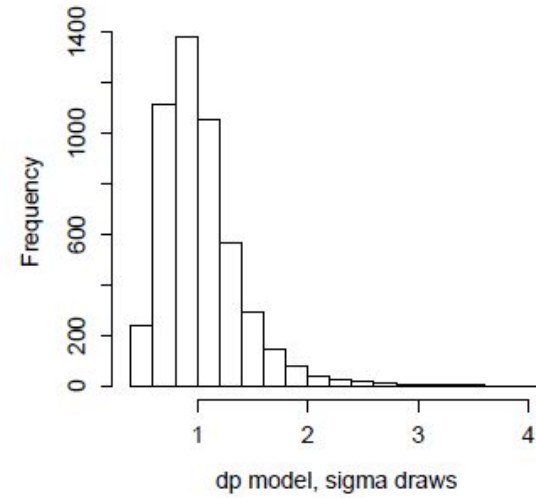
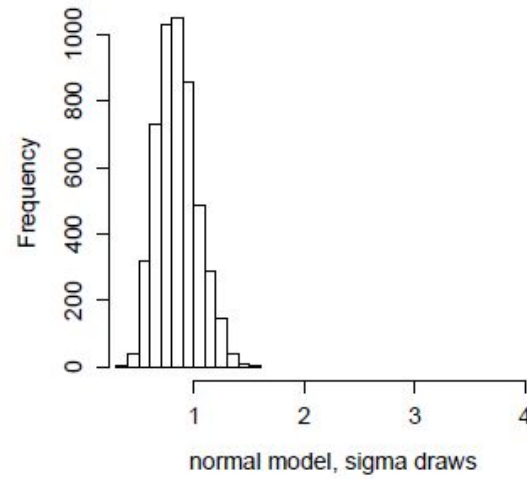
(10 x's but only the first 5 matter)

- We consider two scenarios:
  1.  $\sigma_i = 1$  (iid errors)
  2.  $\sigma_i = \begin{cases} 1 & \text{with probability 0.75} \\ 10 & \text{with probability 0.25} \end{cases}$
- In the second case, the errors are drawn from a mixture distribution.

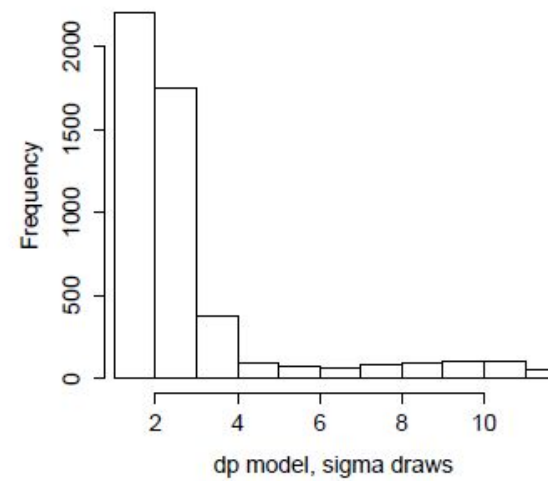
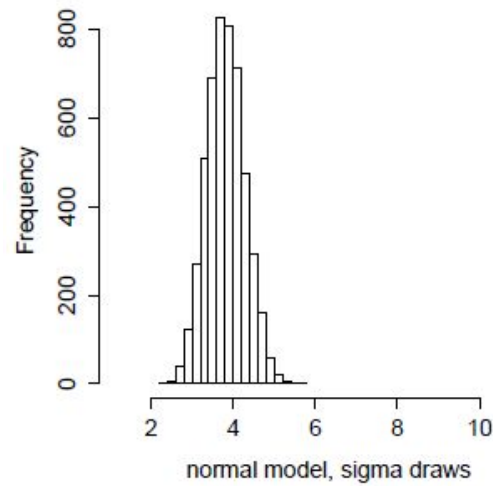
## Normal BART

## DP BART

Data with iid errors



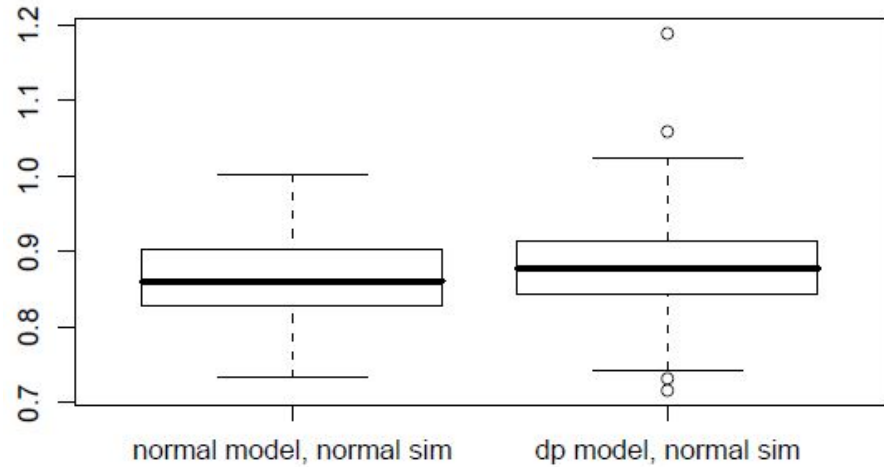
Data with mixed errors



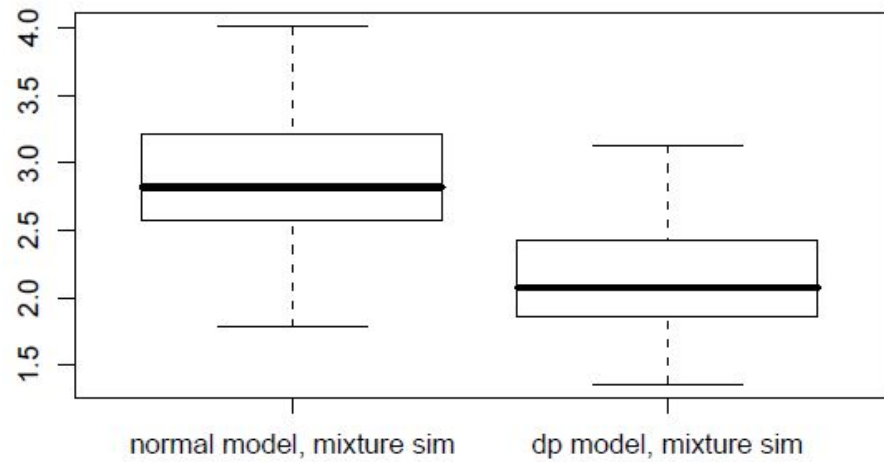
# RMSE values:

Data with iid errors

Normal BART / DP BART



Data with mixed errors



## Conclusions:

- Statistical inference is attainable.
- Adaptations for the specific model are required.
- We gain lots of statistical methodology (e.g. design of experiments and extensions below)
- Straightforward extensions:
  - Variable selection
  - Sequential design
  - Nonparametric errors
  - Binary regression
  - Spatial data

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

- competitive for predictive accuracy,
- robust to prior specification,
- flexible,
- fast (no cv or bootstrap required).

**THE END**