Monotone BART and Monotone Discovery

Rob McCulloch
Arizona State

Collaborations with:
Hugh Chipman (Acadia), Edward George (Wharton, Penn State)
Tom Shively (UT Austin)

Plan

- Review BART.
- Introduce mBART: Monotonic BART.
- Monotonic Discovery
  (very preliminary, comments very welcome (needed)).

Part I. BART (Bayesian Additive Regression Trees)

Data: n observations of y and $x = (x_1, ..., x_p)$

Suppose: $Y = f(x) + \epsilon, \quad \epsilon$ symmetric with mean 0

Bayesian Ensemble Idea: Approximate unknown $f(x)$ by the form

$$f(x) = g(x; \theta_1) + g(x; \theta_2) + ... + g(x; \theta_m)$$

$$\theta_1, \theta_2, \ldots, \theta_m \quad \text{iid} \sim \pi(\theta)$$

and use the posterior of f given y for inference.

BART is obtained when each $g(x; \theta_j)$ is a regression tree.

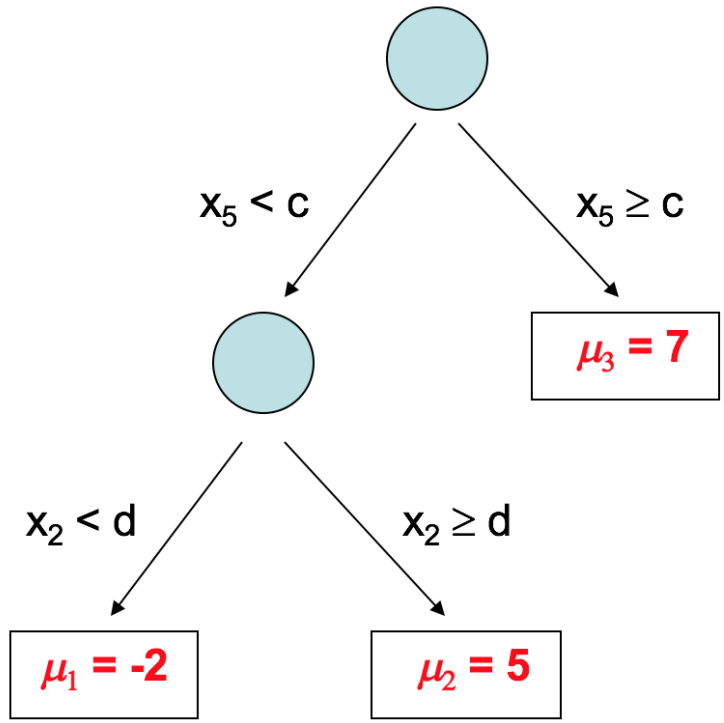
Key calibration: Using y, set $\pi(\theta)$ so that $\text{Var}(f) \approx \text{Var}(y)$.

Beginning with a Single 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 $g(x; T, M)$ be a regression tree function that assigns a μ value to x



A single tree model:

$$Y = g(x; T, M) + \sigma z, \quad z \sim N(0,1)$$

Bayesian CART: Just add a prior $\pi(M, T)$

Bayesian CART Model Search
(Chipman, George, McCulloch 1998)

$$\pi(M, T) = \pi(M \mid T)\pi(T)$$

$\pi(M \mid T) : (\mu_1, \mu_2, \ldots, \mu_b)' \sim N_b(0, \tau^2 I)$

$\pi(T)$: Stochastic process to generate tree skeleton plus uniform prior on splitting variables and splitting rules.

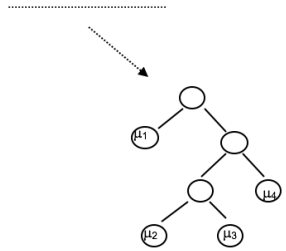
Closed form for $\pi(T \mid y)$ facilitates MCMC stochastic search for promising trees.

Moving on to BART

Bayesian Additive Regression Trees
(Chipman, George, McCulloch 2010)

The BART ensemble model

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



Each (T_i, M_i) identifies a single tree.

$E(Y \mid x, T_1, M_1, \ldots, T_m, M_m)$ is the sum of m bottom node μ's, one from each tree.

Number of trees m can be much larger than sample size n.

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

Complete the Model with a Regularization Prior

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

π applies the Bayesian CART prior to each (T_j, M_j) independently so that:

- Each T small.
- Each μ small.
- σ will be compatible with the observed variation of y.

The observed variation of y is used to guide the choice of the hyperparameters for the μ and σ priors.

π is a "regularization prior" as it keeps the contribution of each $g(x; T_i, M_i)$ small, explaining only a small portion of the fit.

Connections to Other Modeling Ideas

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

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

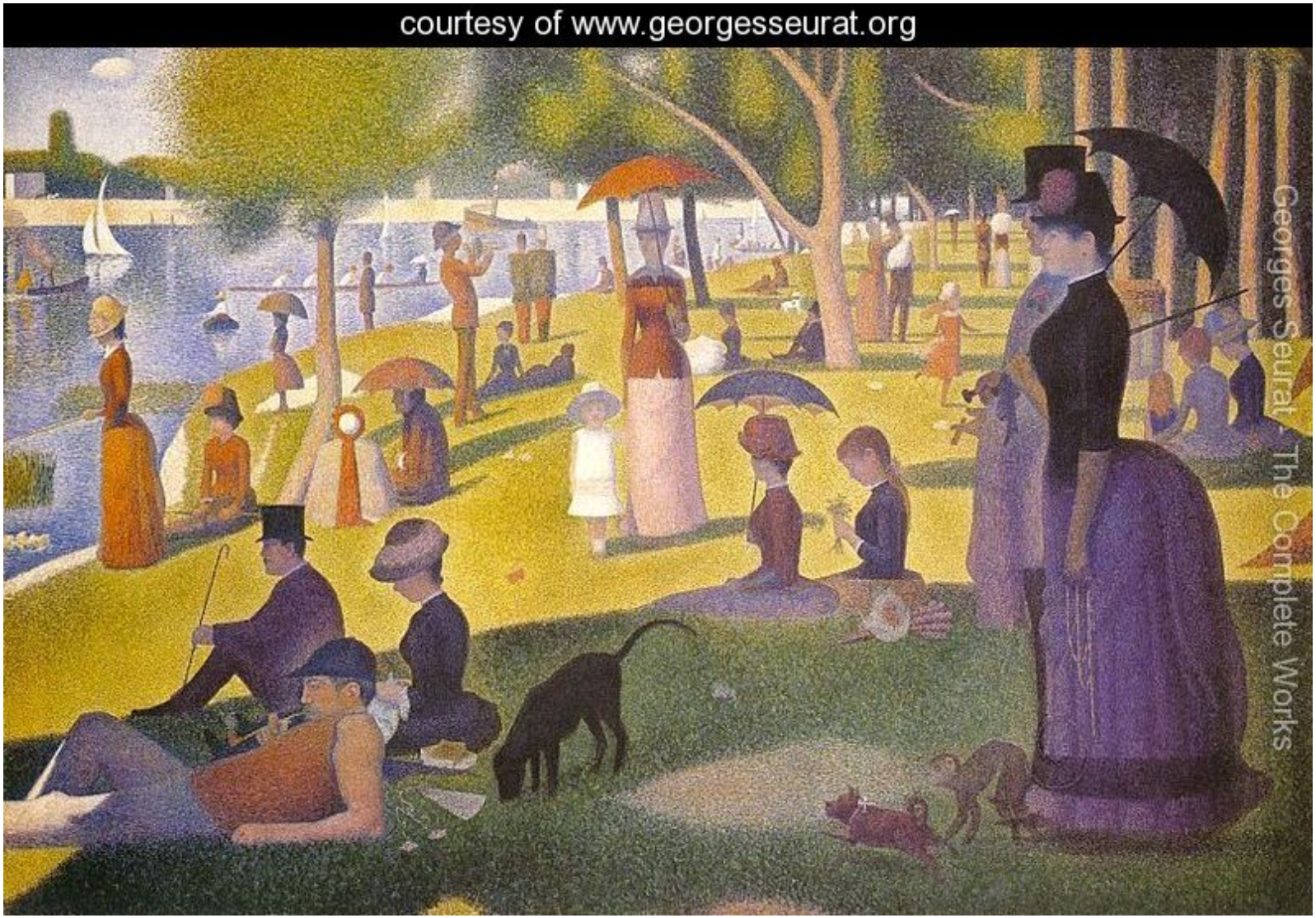
Dynamic Random Basis Elements:

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

Gradient Boosting:

- Fit becomes the cumulative effort of many weak learners

Build up the fit, by adding up tiny bits of fit ..



A Sketch of the BART MCMC Algorithm

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

First, it is a "simple" Gibbs sampler:

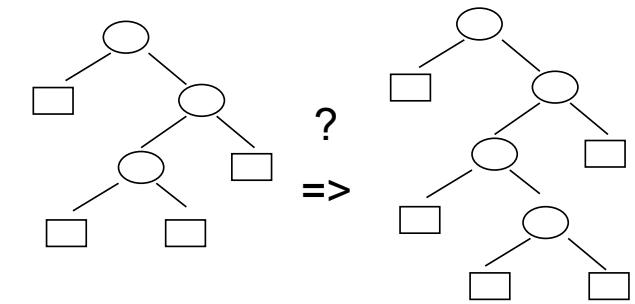
$$\begin{aligned} (T_i, M_i) \quad &| \quad \text{all other } (T_j, M_j), \text{ and } \sigma) \\ \sigma \quad &| \quad (T_1, M_1, \ldots, \ldots, T_m, M_m) \end{aligned}$$

To draw (T_i, M_i) above, subtract the contributions of the other trees from both sides to get a simple one-tree model.

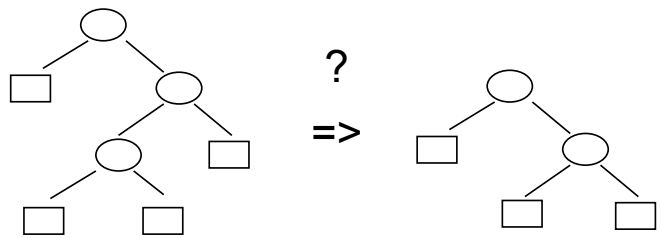
We integrate out M to draw T and then draw $M \mid T$.

For the draw of T we use a Metropolis-Hastings within Gibbs step.

Our proposal moves around tree space by proposing local modifications such as the "birth-death" step:



propose a more complex tree



propose a simpler tree

... as the MCMC runs, each tree in the sum will grow and shrink, swapping fit amongst them ....

Using the MCMC Output to Draw Inference

Each iteration d results in a draw from the posterior of f

$$\hat{f}_d(\cdot) = g(\cdot; T_1, M_1) + \cdots + g(\cdot; T_m, M_m)$$

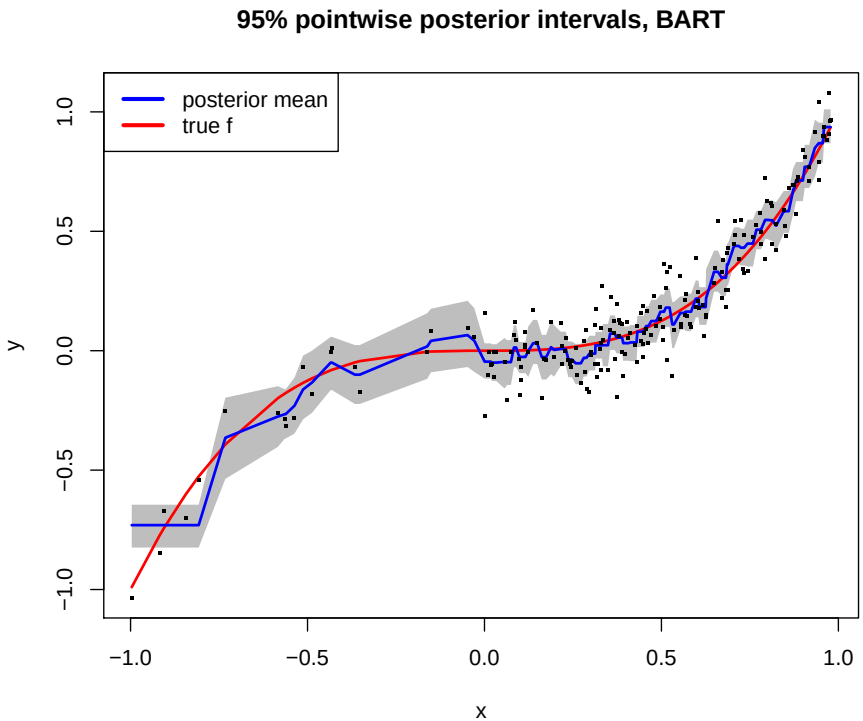
To estimate $f(x)$ we simply average the $\hat{f}_d(\cdot)$ draws at x

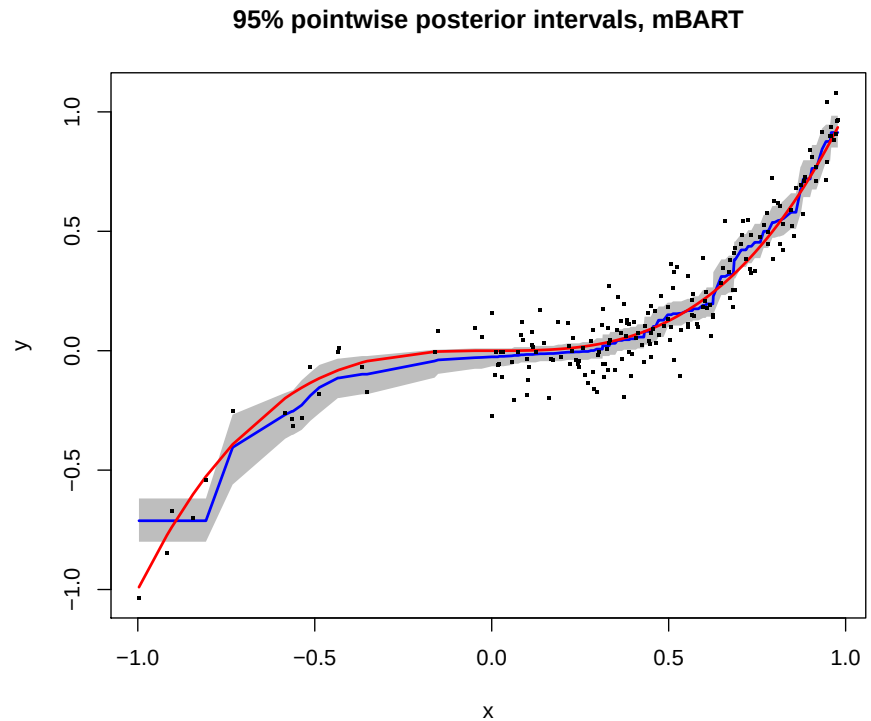
Posterior uncertainty is captured by variation of the $\hat{f}_d(x)$
eg, 95% HPD region estimated by middle 95% of values

Can do the same with functionals of f.

Automatic Uncertainty Quantification

A simple simulated 1-dimensional example





Note: mBART on the right plot still to be discussed

Example: Friedman's Simulated Data

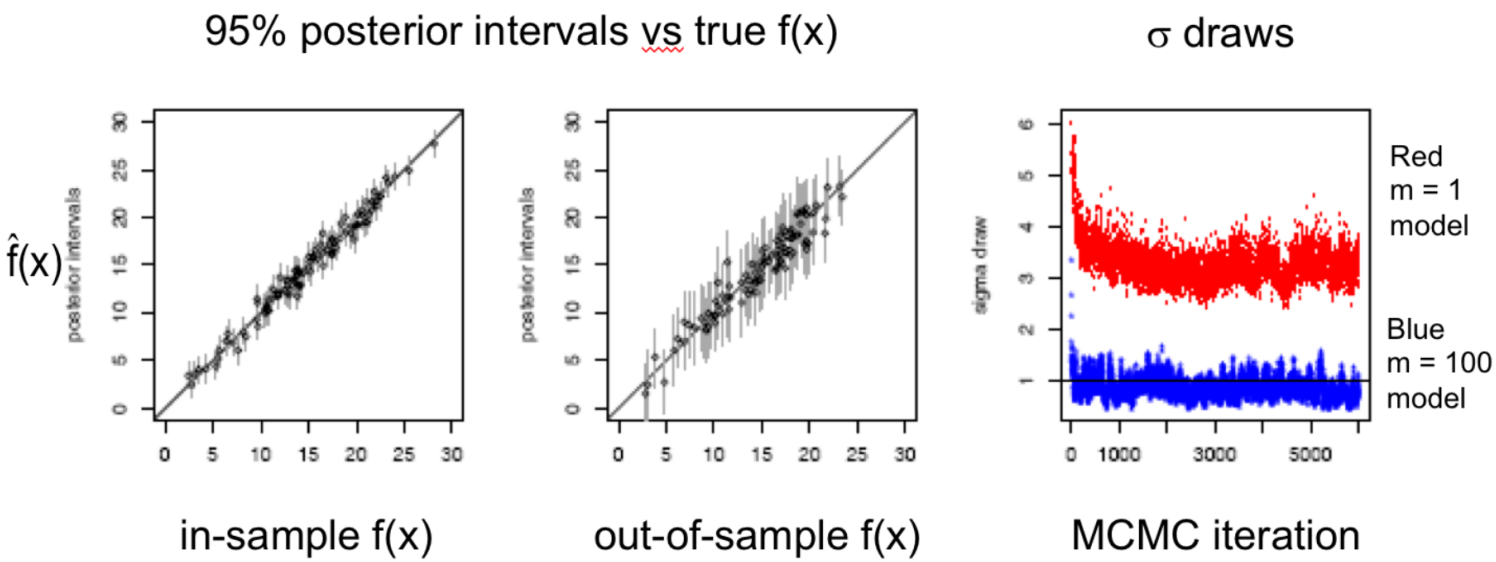
$$Y = f(x) + \epsilon, \quad \epsilon \sim N(0, 1)$$

where

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

- x_i's iid $\sim$ *Uniform*$(0, 1)$
- Only the first 5 x_i's matter!
- Friedman (1991) used $n = 100$ observations from this model to illustrate the potential of MARS
- BART handily outperforms competitors including random forests, neural nets and gradient boosting on this example.

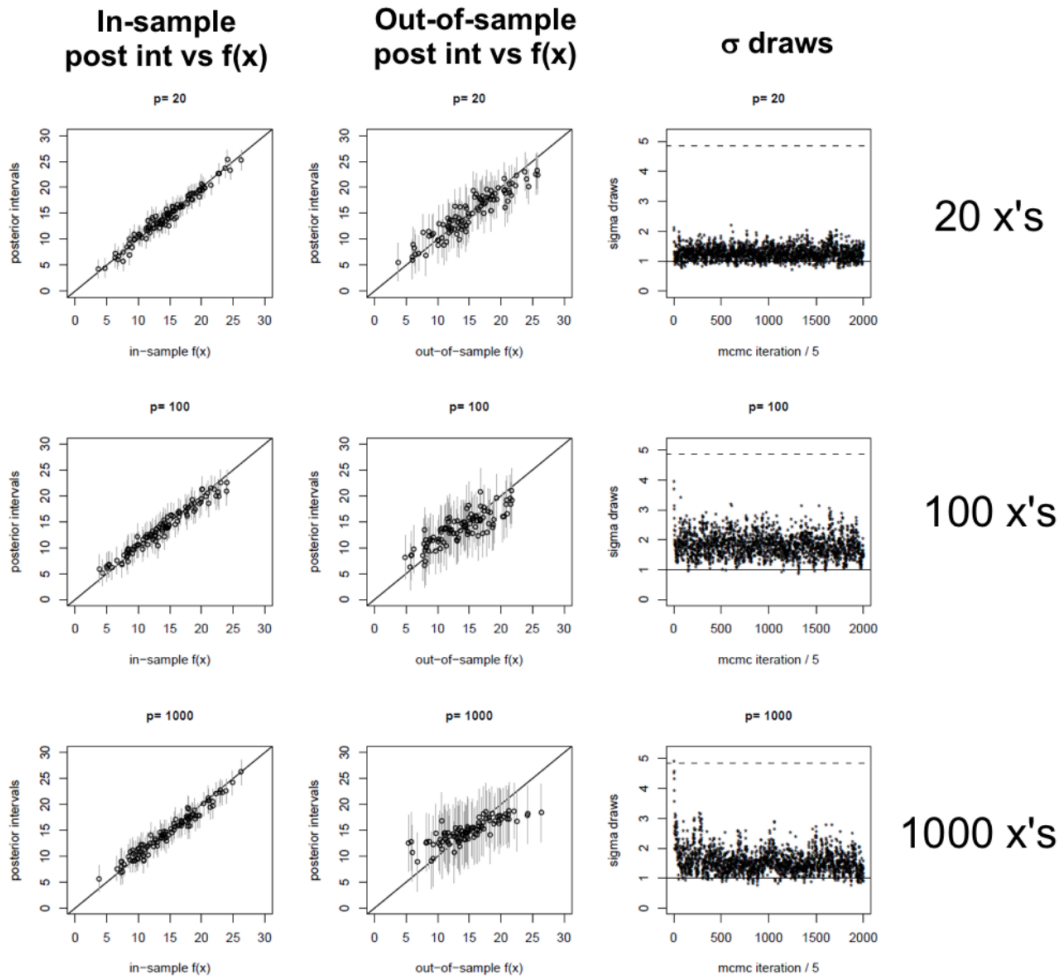
Applying BART to the Friedman Data



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 !!!!



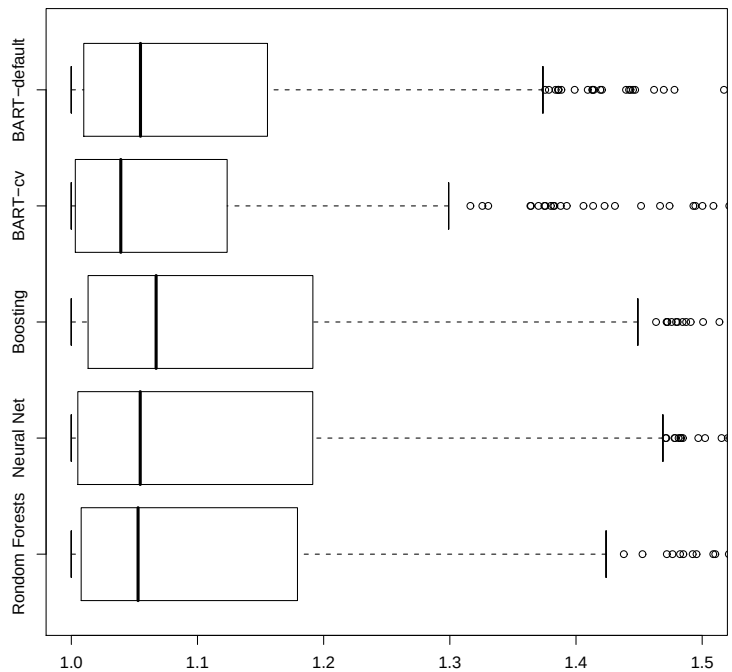
Out of Sample Prediction

Predictive comparisons on 42 data sets.

Data from Kim, Loh, Shih and Chaudhuri (2006) (thanks Wei-Yin Loh!)

- $p = 3$ to 65, $n = 100$ to 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*42 = 840 **out-of-sample predictions**, for each prediction, divide rmse of different methods by the smallest

+ 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!!

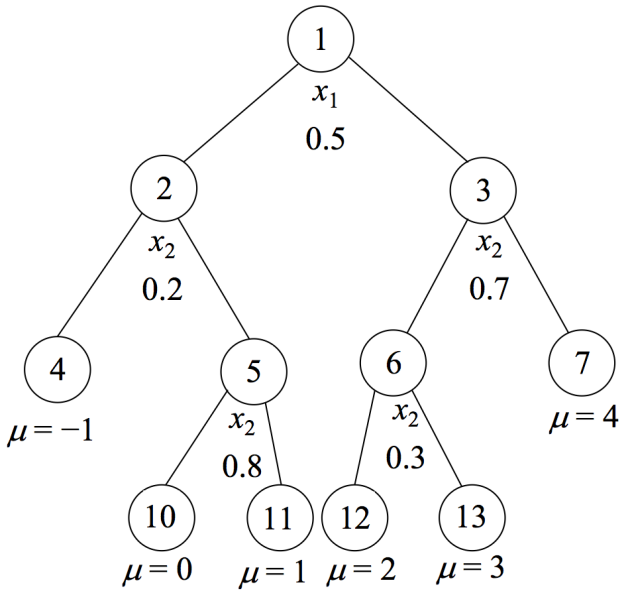


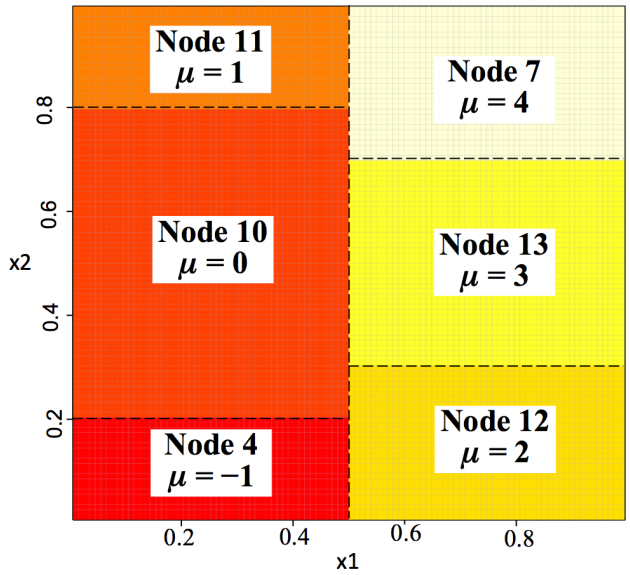
Part II. mBART - Monotone BART

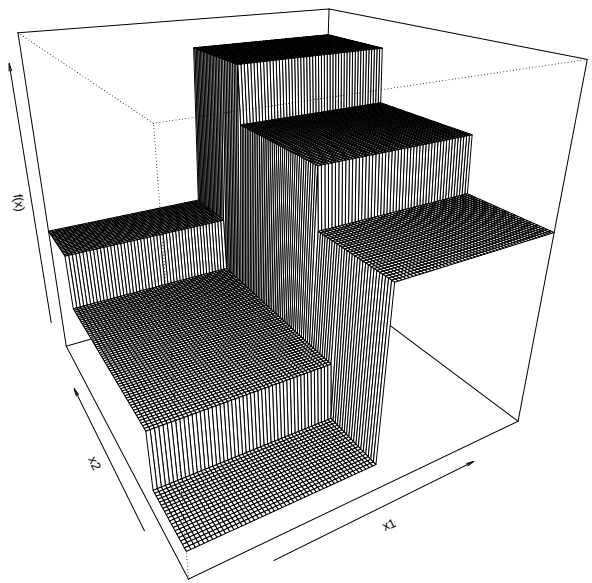
Multidimensional Monotone BART
(Chipman, George, McCulloch, Shively 2018)

Idea:
Approximate multivariate monotone functions by the sum of many single tree models, each of which is monotonic.

An Example of a Monotonic Tree

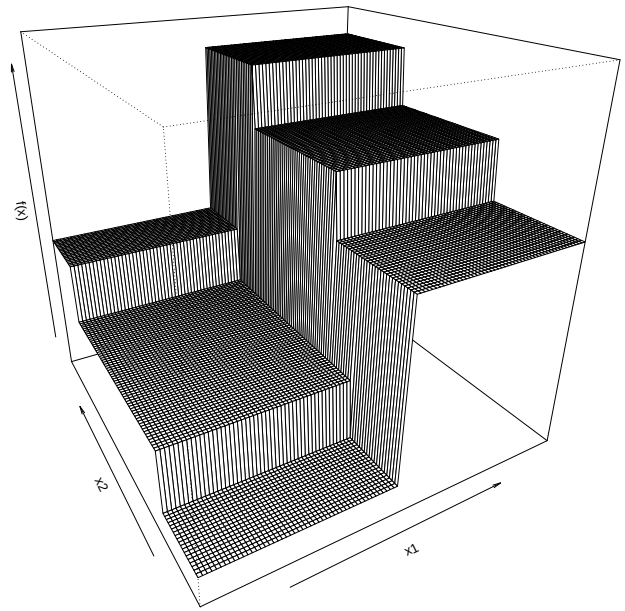






Three different views of a bivariate monotonic tree.

What makes this single tree monotonic?

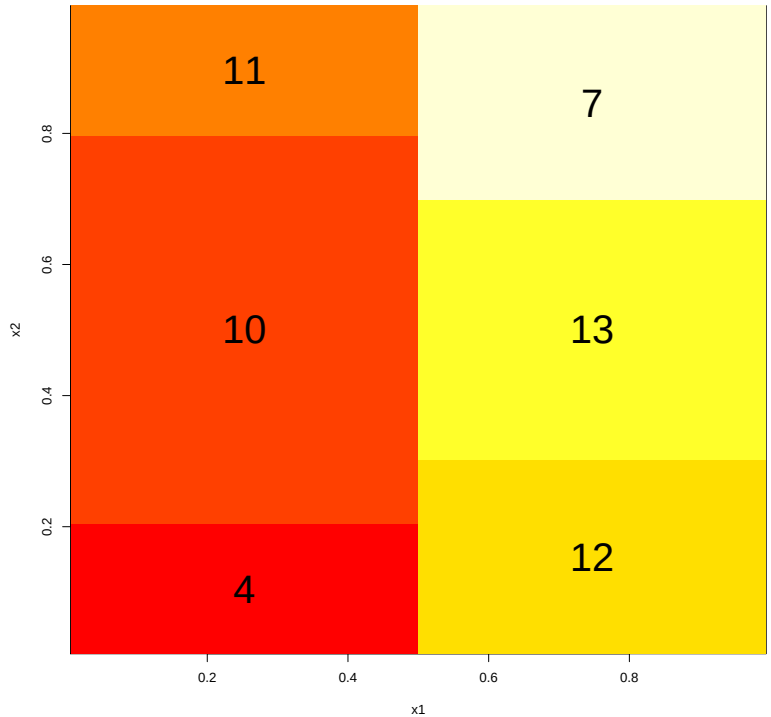


A function g is said to be *monotonic* in x_i if for any $\delta > 0$,

$$g(x_1, x_2, \ldots, x_i + \delta, x_{i+1}, \ldots, x_k; T, M) \geq g(x_1, x_2, \ldots, x_i, x_{i+1}, \ldots, x_k; T, M).$$

For simplicity and wlog, let's restrict attention to monotone nondecreasing functions.

To implement this monotonicity in "tree language" we simply constrain the mean level of a node to be greater than those of it below neighbors and less than those of its above neighbors.



- node 7 is disjoint from node 4.
- node 10 is a below neighbor of node 13.
- node 7 is an above neighbor of node 13.

The mean level of node 13 must be greater than those of 10 and 12 and less than that of node 7.

The mBART Prior

Recall the BART parameter

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

Let $S = \{\theta : \text{each tree is monotonic in a desired subset of } x_i's\}$

To impose the monotonicity we simply truncate the BART prior $\pi(\theta)$ to the set S

$$\pi^*(\theta) \propto \pi(\theta)\, I_S(\theta)$$

where $\chi_S(\theta)$ is 1 if *each* tree is montonic.

A New BART MCMC "Christmas Tree" Algorithm

$$\pi((T_1, M_1), (T_2, M_2), \ldots, (T_m, M_m), \sigma \mid y))$$

Bayesian backfitting again: Iteratively sample each (T_j, M_j) given (y, σ) and other (T_j, M_j)'s

Each $(T^0, M^0) \to (T^1, M^1)$ update is sampled as follows:

- Denote move as $(T^0, M^0_{Common}, M^0_{Old}) \to (T^1, M^0_{Common}, M^1_{New})$
- Propose T^* via birth, death, etc.
- If M-H with $\pi(T, M \mid y)$ accepts (T^*, M^0_{Common})
  - Set $(T^1, M^1_{Common}) = (T^*, M^0_{Common})$
  - Sample M^1_{New} from $\pi(M_{New} \mid T^1, M^1_{Common}, y)$

Only $M^0_{Old} \to M^1_{New}$ needs to be updated.

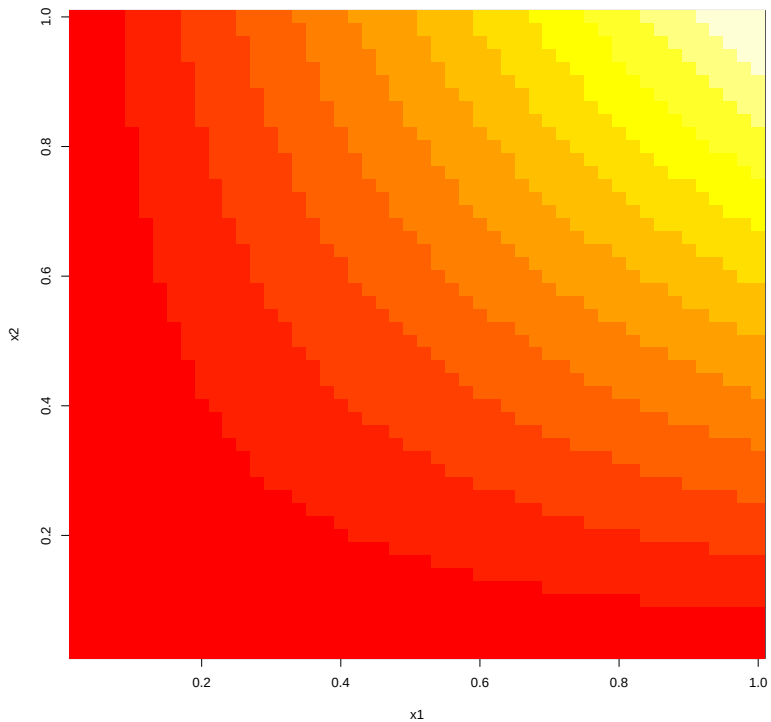
Works for both BART and monotone BART.

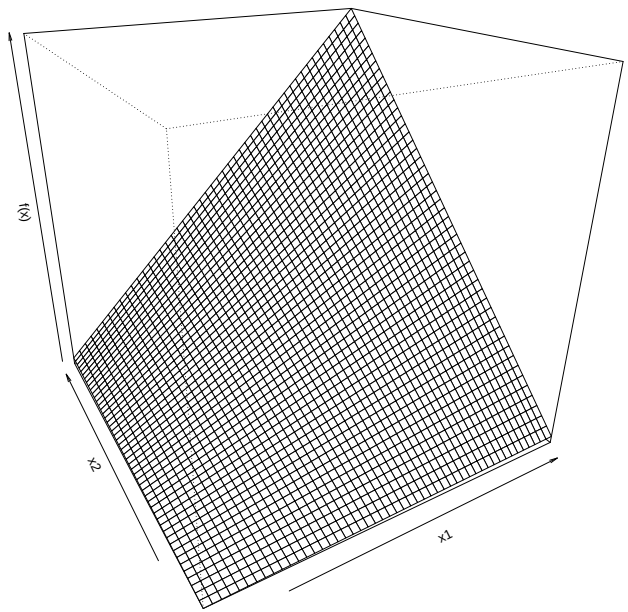
Example: Product of two x's

Let's consider a very simple simulated monotone example:

$$Y = x_1\, x_2 + \epsilon, \quad x_i \sim \text{Uniform}(0, 1).$$

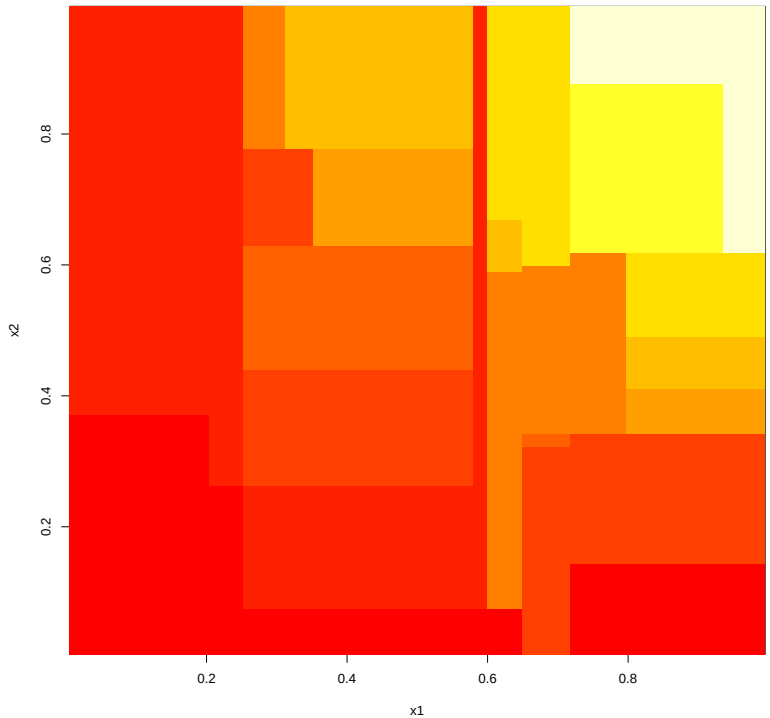
Here is the plot of the true function $f(x_1, x_2) = x_1\, x_2$

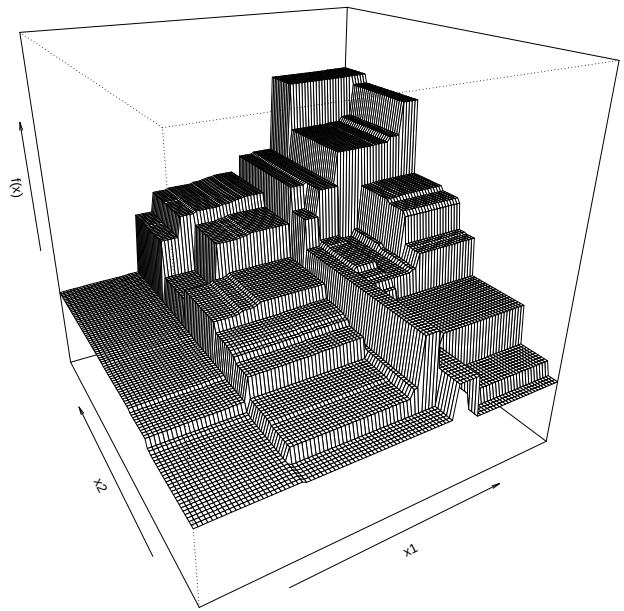




First we try a single (just one tree), unconstrained tree model.

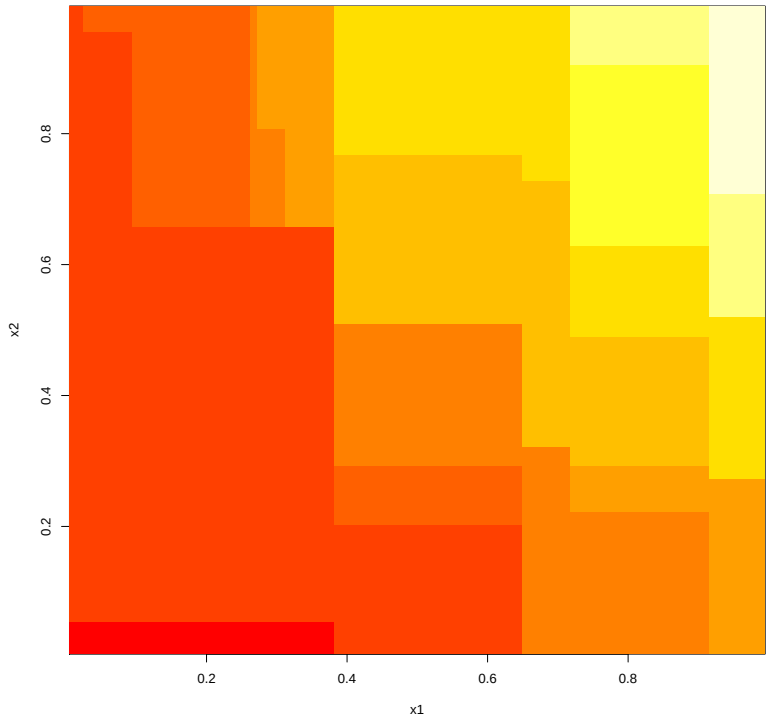
Here is the graph of the fit.

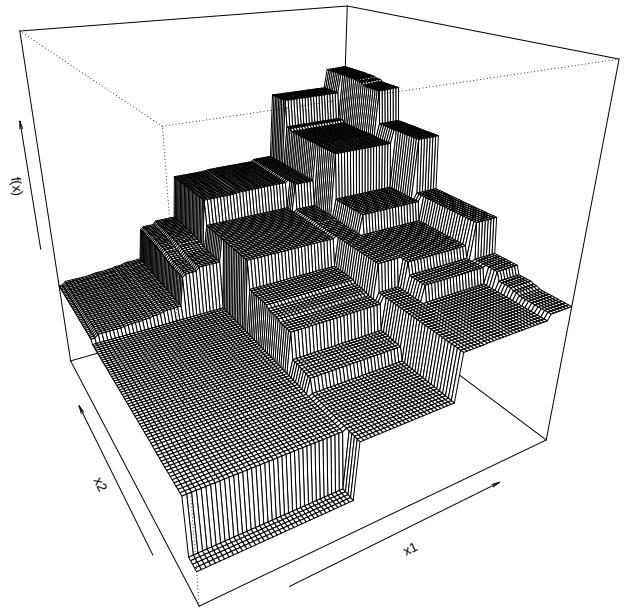




The fit is not terrible, but there are some aspects of the fit which violate monotonicity.

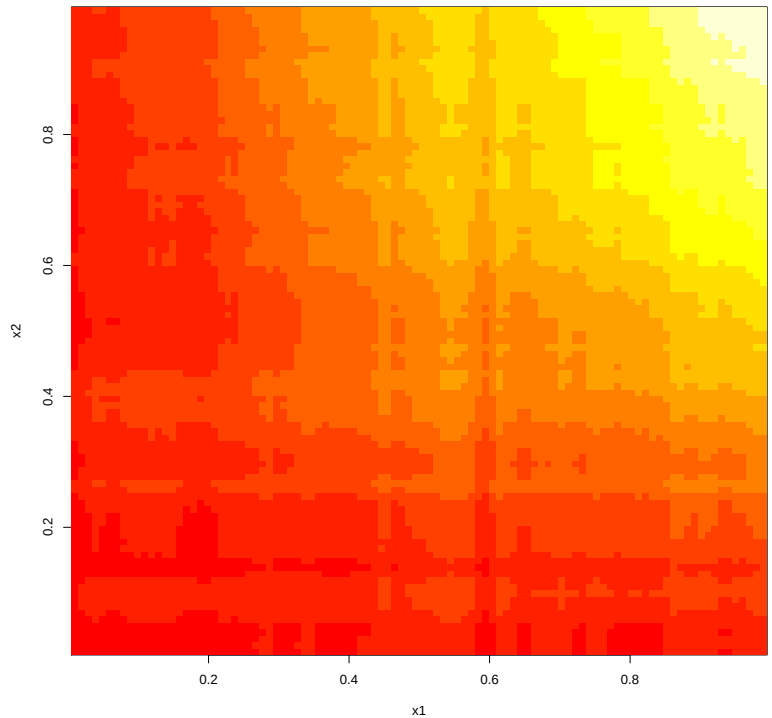
Here is the graph of the fit with the monotone constraint:

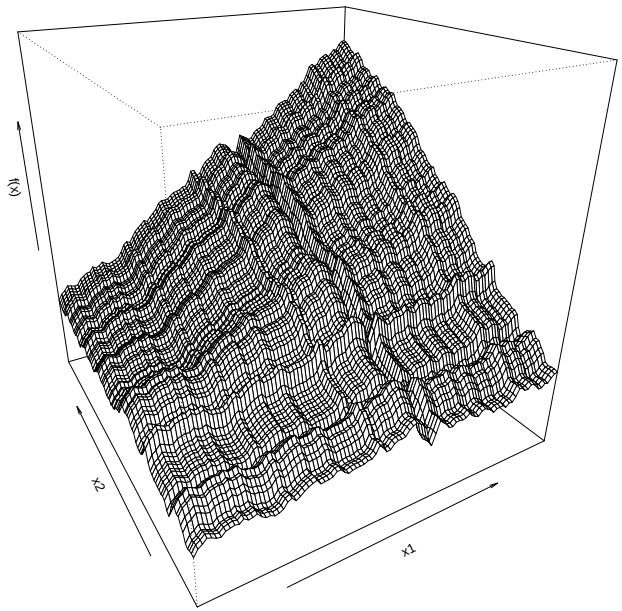




We see that our fit is monotonic, and more representative of the true f.

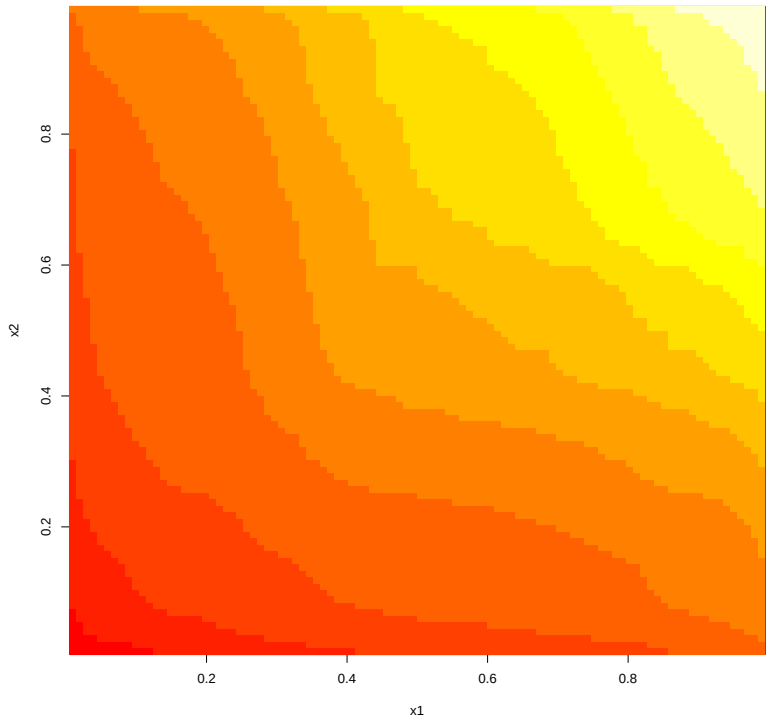
Here is the unconstrained BART fit:

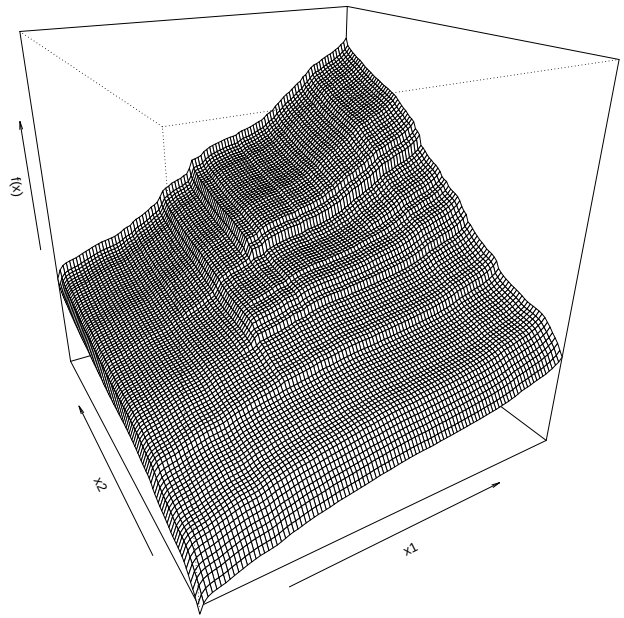




Much better (of course) but not monotone!

And, finally, the constrained BART fit:





Not Bad!

Same method works with any number of x's!

A 5-Dimensional Example

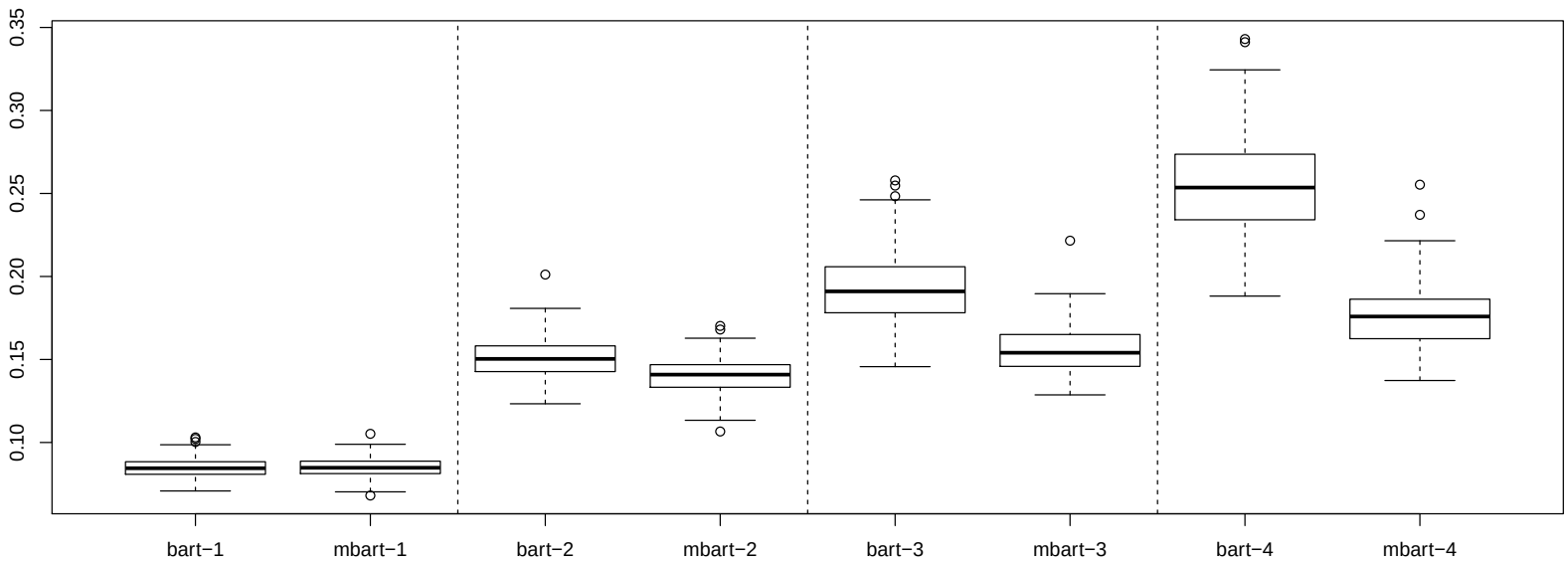
$$Y = x_1\, x_2^2 + x_3\, x_4^3 + x_5 + \epsilon,$$

$$\epsilon \sim N(0, \sigma^2), \quad x_i \sim \text{Uniform}(0, 1).$$

We simulated 5,000 observations, with $\sigma = .1$.

RMSE improvement over unconstrained BART

σ	Monotone BART RMSE	Unconstrained BART RMSE	Percentage Increase
0.5	0.14	0.16	14%
1.0	0.17	0.28	65%



$$\sigma = 0.2, 0.5, 0.7, 1.0$$

Part III. Discovering Monotonicity with mBART

Suppose we are not sure if a function is monotone.

Good news! mBART can be deployed to estimate the monotone components of f.

Thus monotonicity can be discovered rather than imposed!

One way to think about it is:

> *any function can be written as the sum of a montonone up function and a monotone down function:*

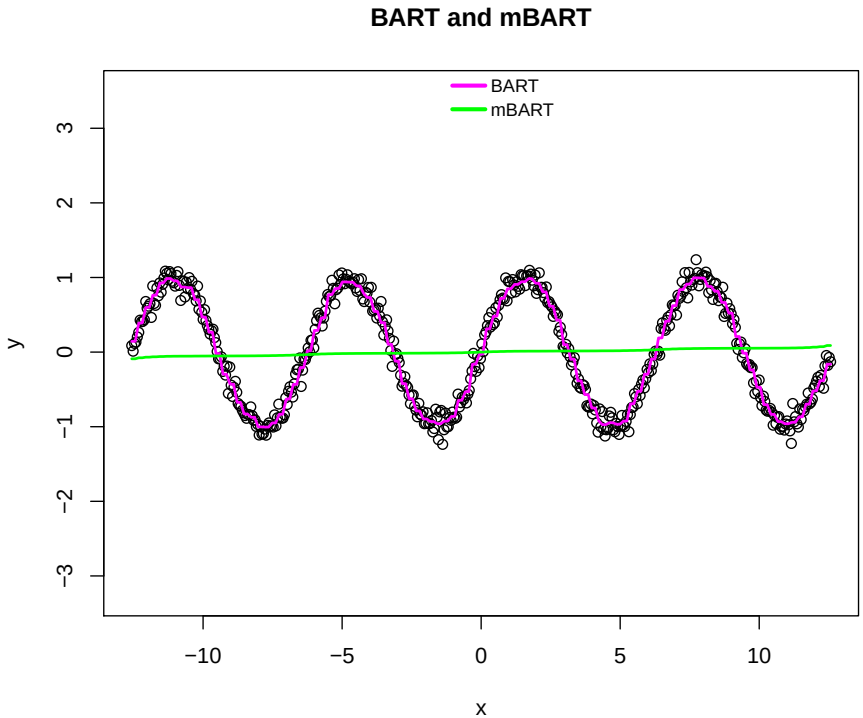
$$f(x) = f^{up}(x) + f^{down}(x)$$

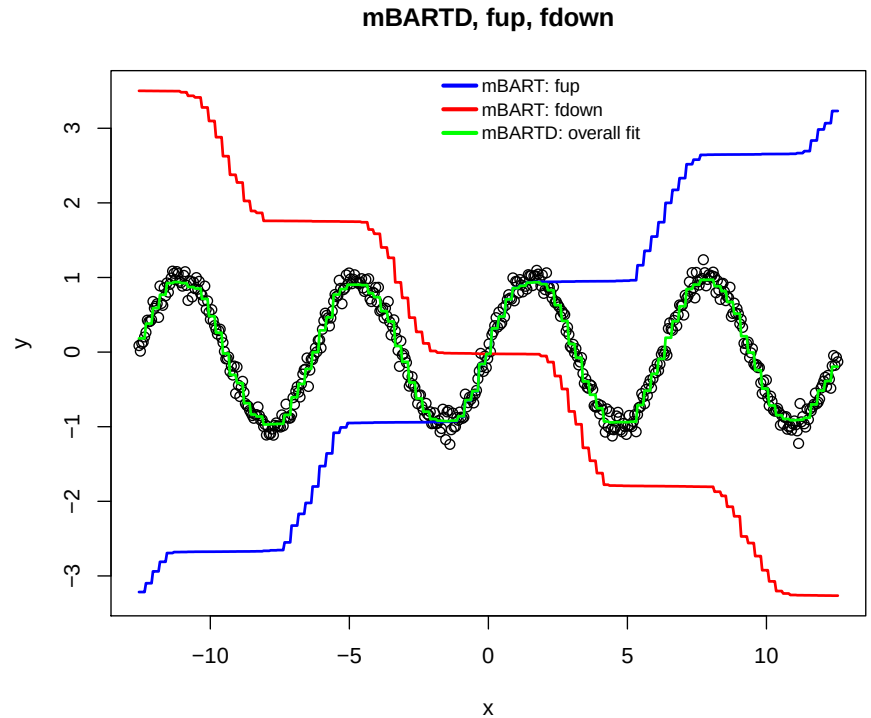
So, for example, we can *discover* that f is montonic increasing by estimating the model

$$Y_i = f^{up}(x) + f^{down}(x) + \epsilon_i$$

and looking to see if $f^{down} \approx 0$.

Example: Suppose $Y = sin(x) + \epsilon$.
(ignore the title of the right panel *for now*).

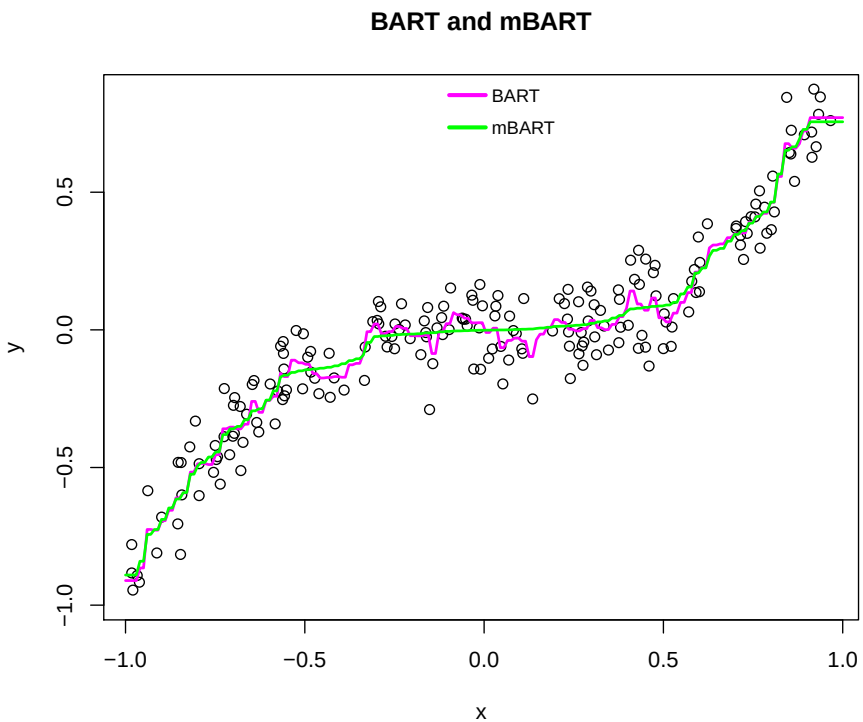


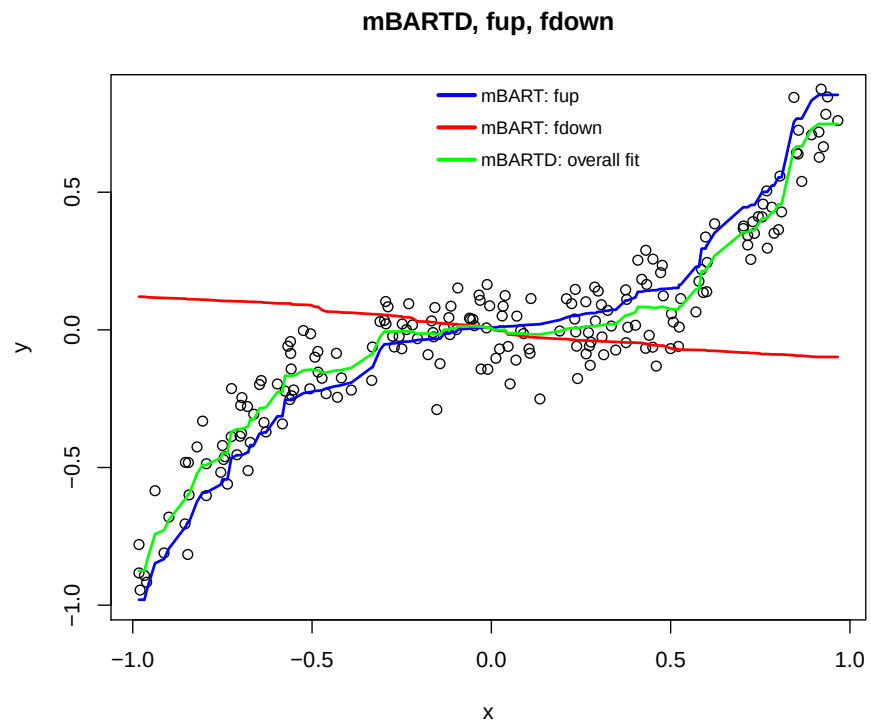


f is not monotone because both f^{up} and f^{down} are clearly non-zero.

Of course, the idea is that the same approach will work for high-dimensional x !!!

Example: Suppose $Y = x^3 + \epsilon$.





We have (the red in the left plot)
$f^{down} \approx 0$ so that f is monotone up.

Note that it does not quite work because the decomposition is not identified:

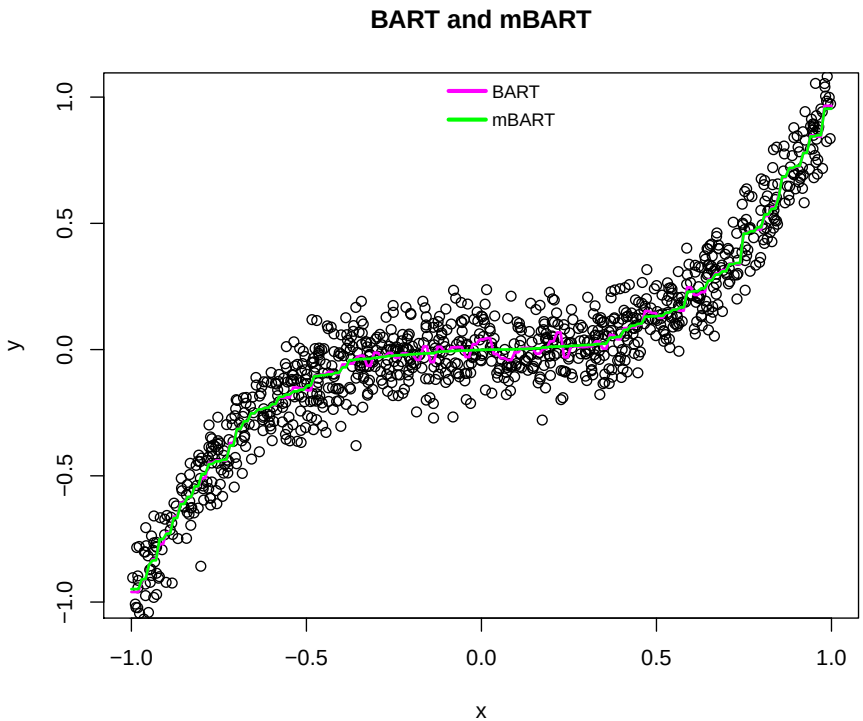
$$f^{up}(x) + f^{down}(x) = (f^{up}(x) + \tilde{f}(x)) + (f^{down}(x) - \tilde{f}(x))$$

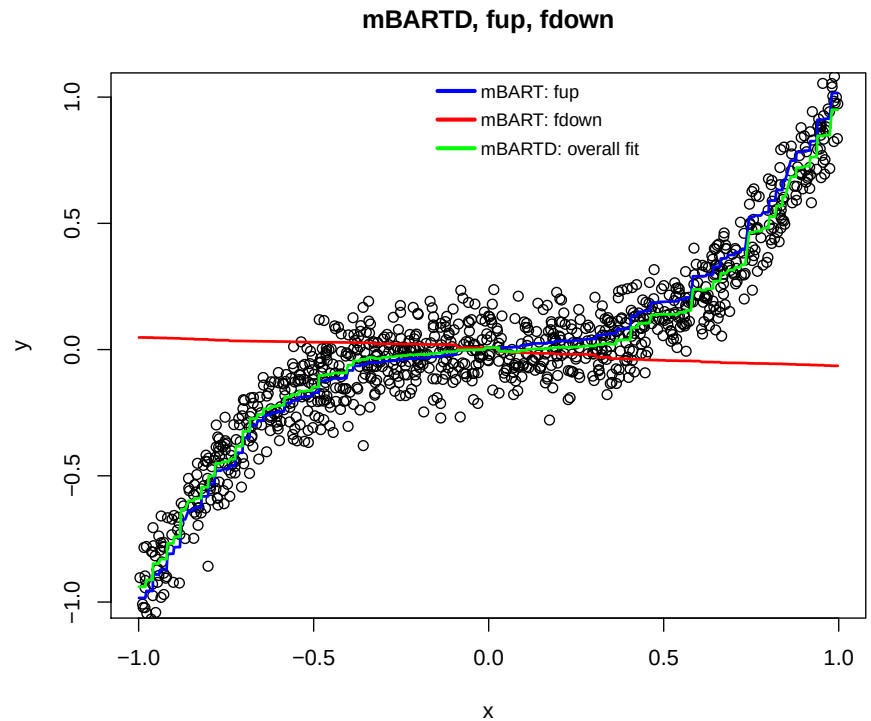
As long as $\tilde{f}$ is monotone up $f^{up} + \tilde{f}$ is monotone up and $f^{down} - \tilde{f}$ is monotone down and we have the same fit.

In all our examples so far (not too many!!) this is not too much of an issue, but we need a simple way to clean things up.

Or, it could just be the information level in the data.

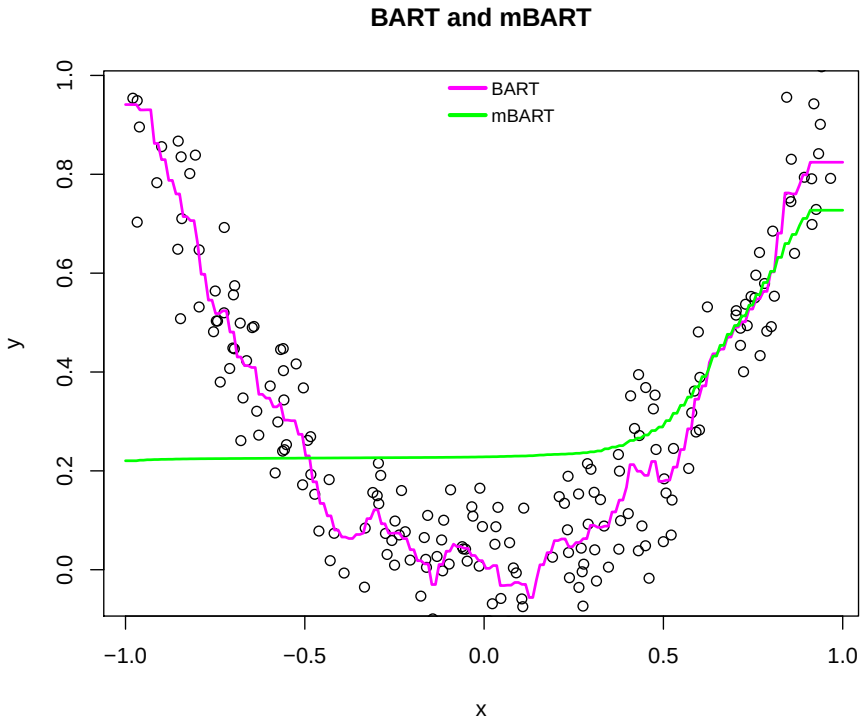
The sample size is increased from 200 to 1,000.

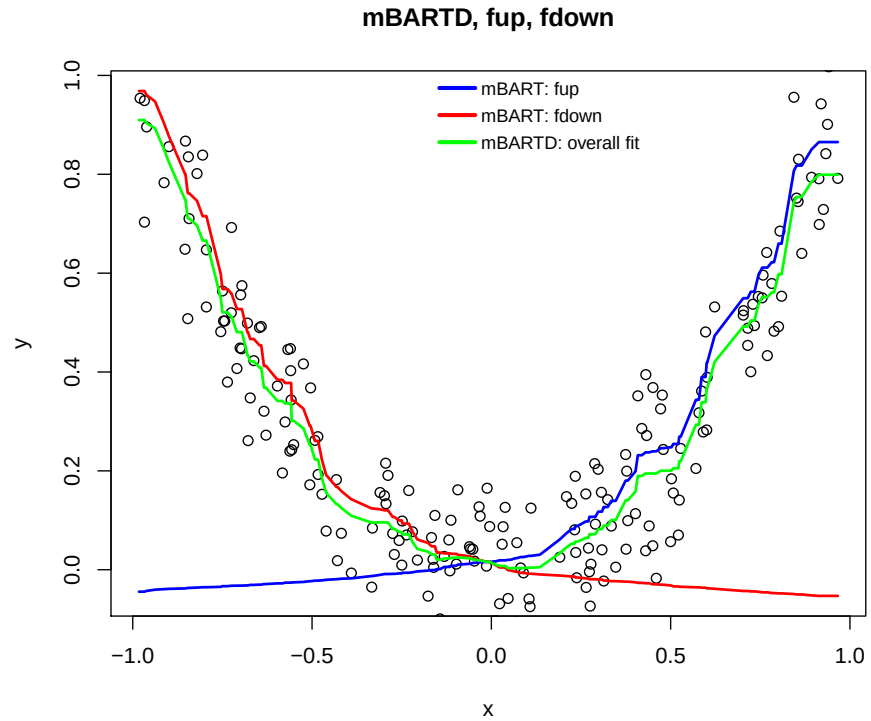




Much better !!

Example: Suppose $Y = x^2 + \epsilon$.





Clearly, f is not montonic, since both f^{up} and f^{down} are appreciable.

f has regions where $f^{up} \approx 0$ and $f^{down} \approx 0$.

Discovering Monotonicity, Simple House Price Data

Let's look at a very simple example where we relate y=house price to three characteristics of the house.

```
> head(x)
     nbhd size brick
[1,]    2 1.79     0
[2,]    2 2.03     0
[3,]    2 1.74     0
[4,]    2 1.98     0
[5,]    2 2.13     0
[6,]    1 1.78     0
> dim(x)
[1] 128   3
> summary(x)
      nbhd            size           brick
 Min.   :1.000   Min.   :1.450   Min.   :0.0000
 1st Qu.:1.000   1st Qu.:1.880   1st Qu.:0.0000
 Median :2.000   Median :2.000   Median :0.0000
 Mean   :1.961   Mean   :2.001   Mean   :0.3281
 3rd Qu.:3.000   3rd Qu.:2.140   3rd Qu.:1.0000
 Max.   :3.000   Max.   :2.590   Max.   :1.0000
> summary(y)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   69.1   111.3   126.0   130.4   148.2   211.2
```

y: thousands of dollars.

x: thousands of square feet.

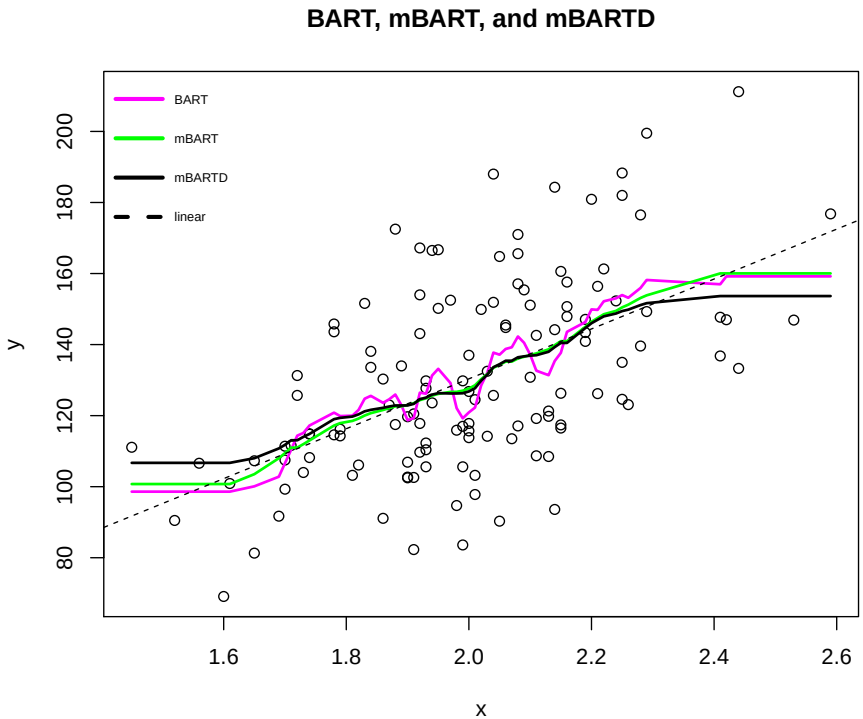
three neighborhoods, brick or not.

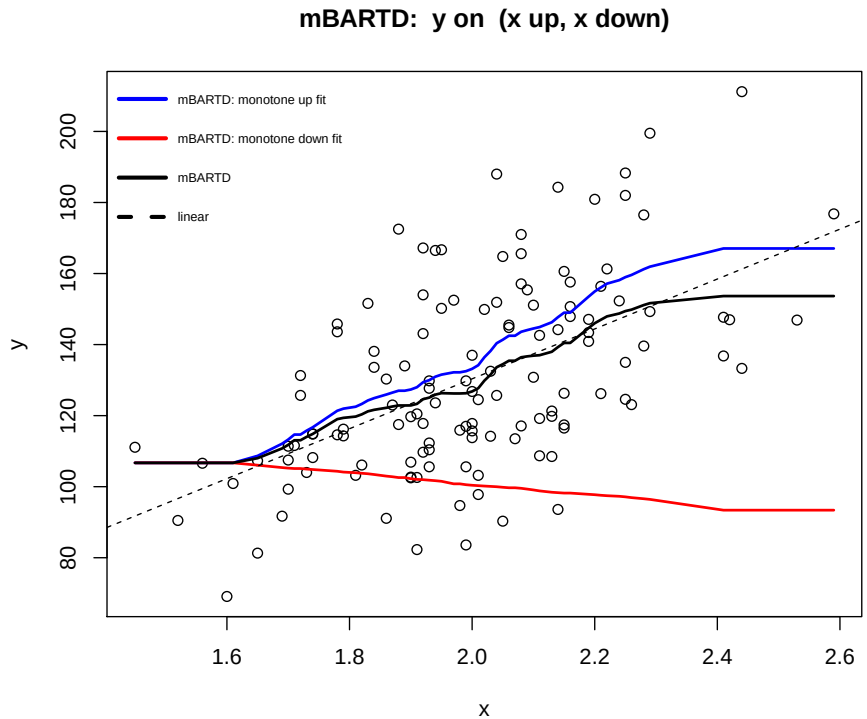
```
Call:
lm(formula = price ~ nbhd + size + brick, data = hdat)

Residuals:
    Min      1Q  Median      3Q     Max
-30.049  -8.519   0.137   7.640  36.912

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   18.725     10.766   1.739   0.0845 .
nbhd2          5.556      2.779   1.999   0.0478 *
nbhd3         36.770      2.958  12.430  < 2e-16 ***
size          46.109      5.527   8.342 1.25e-13 ***
brickYes      19.152      2.438   7.855 1.69e-12 ***
---

Residual standard error: 12.5 on 123 degrees of freedom
Multiple R-squared:  0.7903,Adjusted R-squared:  0.7834
F-statistic: 115.9 on 4 and 123 DF,  p-value: < 2.2e-16
```

According to the linear regression we are monotone up in all three variables.

Note that for the linear model we have to dummy up nbhd, for bart we can just put it in a numeric variable and we get an ordered categorical variable.

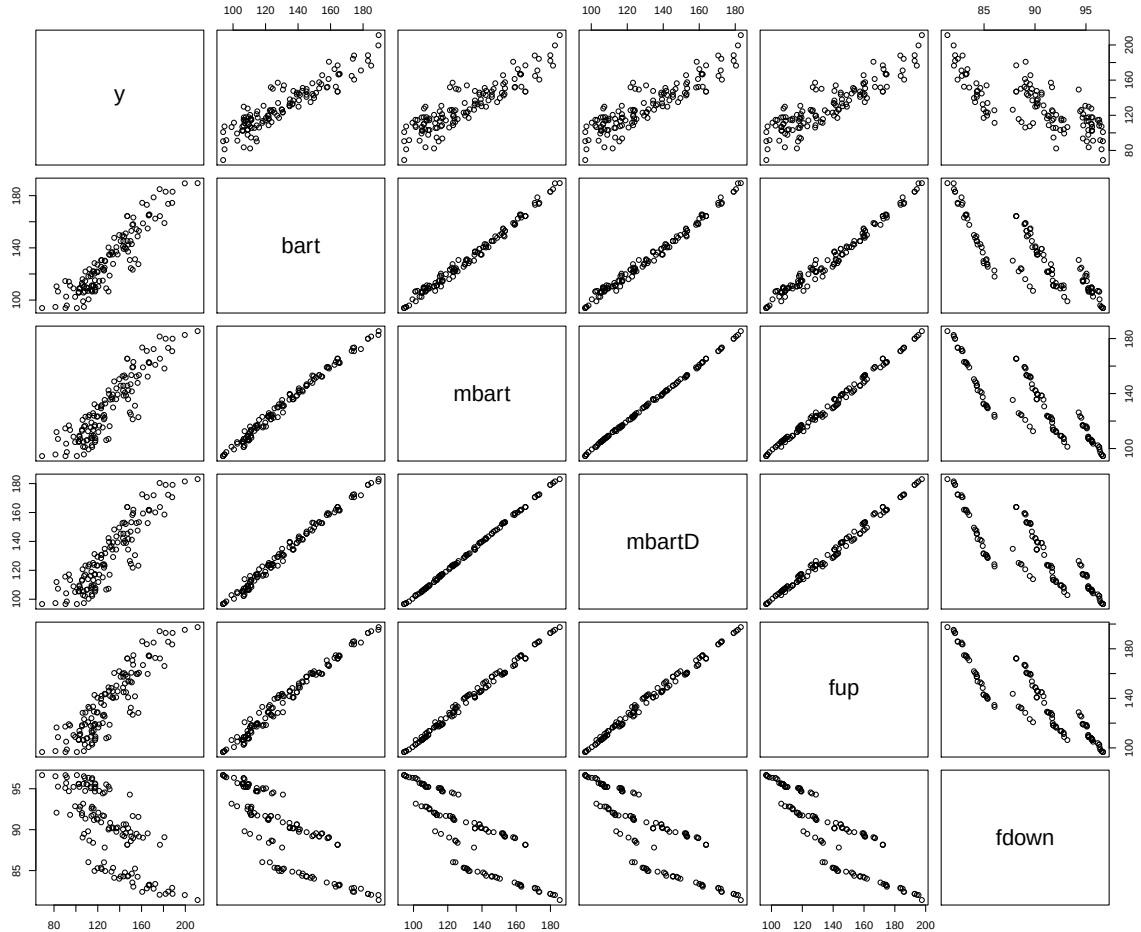
Here are some results when we just use the size of the house:





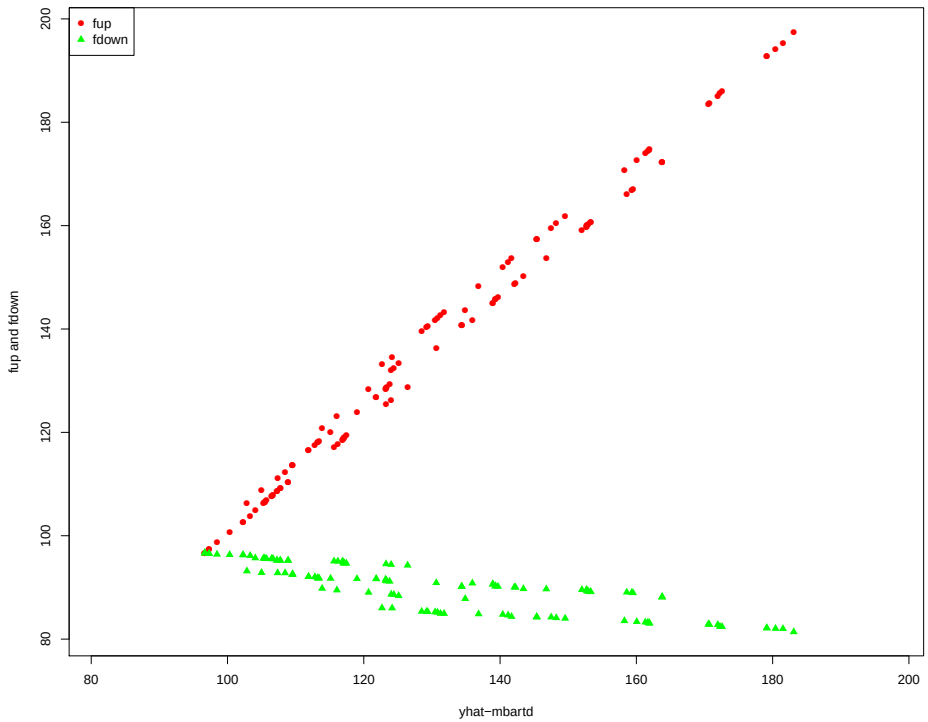
- BART looks bad.
- $f^{down} \approx 0 \Rightarrow f$ monotone up.

Here are some fitted values from various models using all three x variable.



$f \approx f^{up} \Rightarrow f$ is multivariate monotonic !!!

x axis: $\hat{f} = \hat{f}^{up} + \hat{f}^{down}$.
y axis: red: $\hat{f}^{up}$, green: $\hat{f}^{down}$.



$f \approx f^{up} \Rightarrow f$ is multivariate monotonic !!!

A Simple Adjustment for the Identification Issue

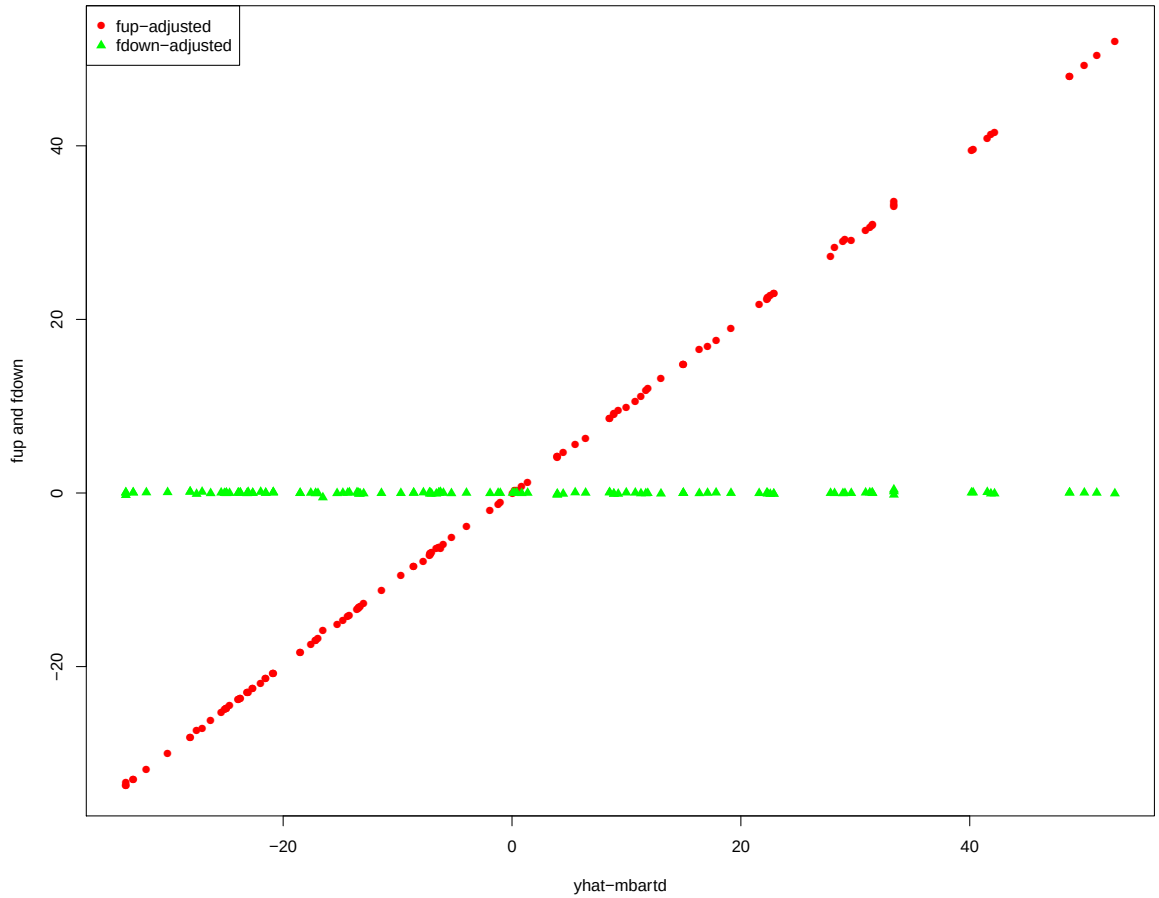
We:

- linearly regress $\hat{f}^{down}$ on x.
- Subtract the fitted values from the regression from $\hat{f}^{down}$ and add them to $\hat{f}^{up}$.
- Demean $\hat{f}$, $\hat{f}^{down}$, and $\hat{f}^{up}$.

I checked that all the signs in the linear regression are negative so that it represents a downward monotonic function.

We are trying to "tilt" the functions to account for the possible non-indentification.

Here is the plot after the adjustment.



$f = f^{up} \Rightarrow f$ is multivariate monotonic !!!

Is mBARTD better than BART + mBART?

Why not just fit bart and mBart and look at the difference?

We are excited about potential benefits of mBARTD (monotonic discovery):

- "regularization": $f^{up} + f^{down}$ can fit anything, but the sum of smooth is smooth so we replace BART with a smoother high dimensional fitter which automatically seeks out regions of monotonicity and takes advantage of this simplifying structure.
- "regularization": we can use variable selection to figure which variables f is montonic in.
- May be scientifically interesting to learn regions of monotonicity.

We haven't got these working yet !!!

Variable Selection

How do we fit

$$Y_i = f^{up}(x) + f^{down}(x) + \epsilon_i \ ?$$

We could do a Gibbs sampler:

$$f^{up} \mid f^{down}, \sigma \qquad f^{down} \mid f^{up}, \sigma$$

We may try this, but it is not what we have been doing.

How have we been fitting mBARTD?

$$Y_i = f(x, -x) + \epsilon_i$$

For each x variable we put both x and $-x$ in and then us mBART!!!

To get f^{up} we just evaluate $f(x, 0)$.
To get f^{down} we just evaluate $f(0, -x)$.

We just have to do variable selection in this setting (could try Linero or Carvalho, Hahn, McCulloch or Chipman, George, and McCulloch).

Concluding Remarks

- The fully Bayesian nature of BART greatly facilitates extensions such as mBART and now many others (Murray, Linero, Sparapani et. al).
- Despite its many compelling successes in practice, theoretical frequentist support for BART is only now just beginning to appear.
- In particular, Rockova and van der Pas (2017) *Posterior Concentration for Bayesian Regression Trees and Their Ensembles* recently obtained the first theoretical results for Bayesian CART and BART, showing near-minimax posterior concentration when $p > n$ for classes of Holder continuous functions.
- Software for mBART is available at https://bitbucket.org/remcc/mbart.

Thank You!