# Monotonic BART and Monotonic Discovery

Ed George, Rob McCulloch

1. Monotone BART
2. Monotonic Trees
3. mBART Prior
4. A New BART MCMC “Christmas Tree” Algorithm
5. Example: Product of two $x$’s
6. A 5-Dimensional Example
7. Discovering Monotonicity with mBART
8. Discovering Monotonicity, Simple House Price Data

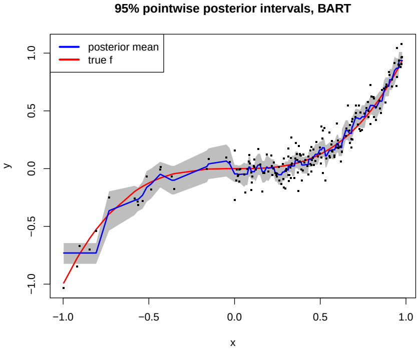
# 1. Monotone BART

A basic goal of BART modeling is to be flexible.

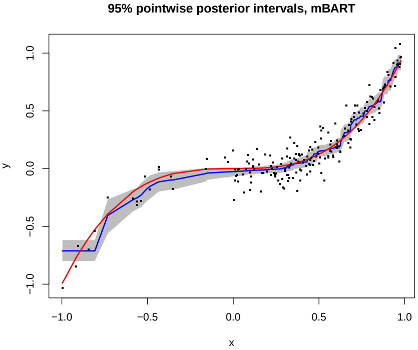
However, it is always good to introduce prior information.

*What is if we know the $f$ is monotonic, that could be usefull !!*

# A simple simulated 1-dimensional example



95% pointwise posterior intervals, BART



95% pointwise posterior intervals, mBART

Idea:

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

This works because

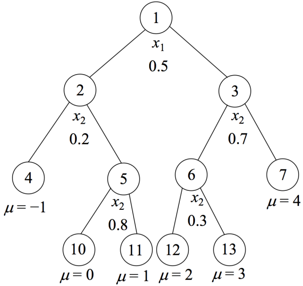
1. We can easily define a notion of "monotonic" for a single tree.
2. Because trees are simple, we can construct an MCMC which respects the constraints.

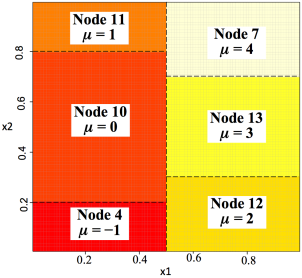
*But*,

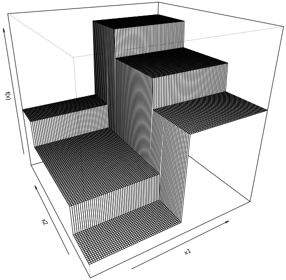
*we still use the BART/boosting approach to modeling with trees:*

*complex montonic functions are built as the sum of many single tree models, each of which is monotonic.*

# 2. Monotonic Trees

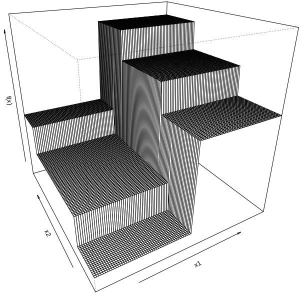






Three different views of a bivariate monotonic tree.

What makes this single tree monotonic?



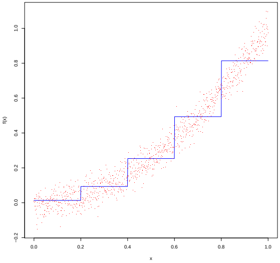
A function $g$ is said to be *monotonic* in $x_i$ if for any $\delta > 0$,

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

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

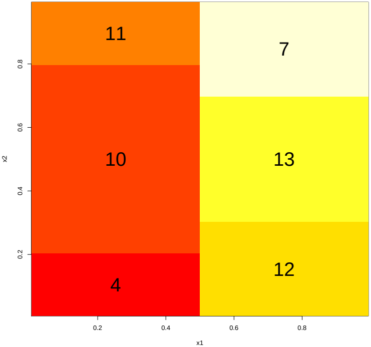
How to express this condition in “tree language”?

With just one $x$ variable, we can easily see what to do:



- for any bottom node, there may be a neighboring region above and and a neighboring region below.
- the $\mu$ level for each bottom node must be greater than that of a below neighbor, and less than that of an above neighbor.

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.

Note:

For any bottom node, we can figure out (and easily code) the constraint interval for the mean level $\mu$ of that bottom node given the rest of the tree.

> *Above your belows, below your aboves.*

Because we only make local changes via the new MCMC algorithm, (as we'll see), this criterion suffices for all computations.

Fortunately no further conditions on the constrained set of bottom node $\mu$'s is needed.

# 3. mBART Prior

Recall the BART parameter

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

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

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 $I_S(\theta)$ is 1 if *every* tree in $\theta$ is montonic.

# 4. 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 $(y, \sigma)$ and other $(T_j, M_j)$'s

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

- Denote move as $(T^0, M^0_{Common}, M^0_{Old}) \rightarrow (T^1, M^0_{Common}, M^1_{New})$
- 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 $M^0_{Old} \rightarrow M^1_{New}$ needs to be updated.

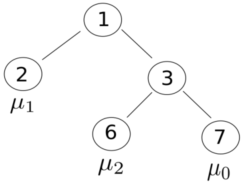
Works for both BART and mBART.

$M^0_{Common} = \{\mu_1, \mu_2\}$

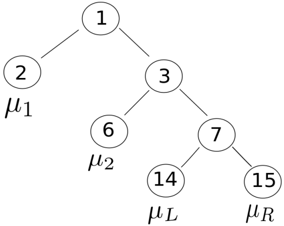
Old BART algorithm:
integrate out all the $\mu$'s and then play around with the tree.

Christmas Tree:
condition on all the $\mu$ not affected by the proposed tree move.

$(T^0, M^0)$:



$(T^*, M^*)$:



## Comments on Algorithm

- MCMC works on a single tree at a time.
- MCMC makes local moves so we only have to think about at most two bottom nodes at a time

  $\Rightarrow$

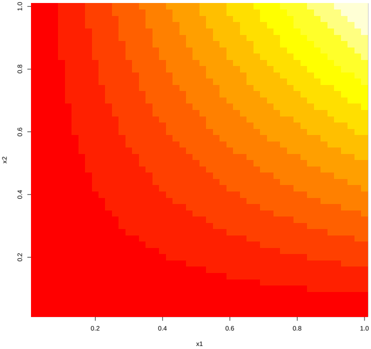
  *don't* have to understand the full set of constrained $\mu_i, i = 1, 2, \ldots, b$ for $b$ bottom nodes.
- In constrained problems where $M_{New}$ cannot be integrated out, the small size of $M_{New}$ makes numerical approximation by discretization feasible.

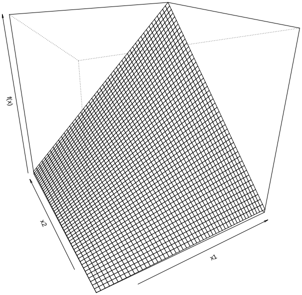
# 5. Example: Product of two *x*'s

Let's consider a very simple simulated monotone example:

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

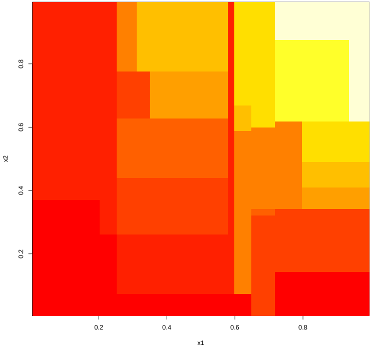
Here is the plot of the true function $f(x_1, x_2) = x_1\, x_2$

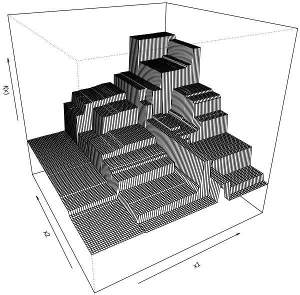




First we try a single (just one tree), unconstrained tree model fit to some simulated data.

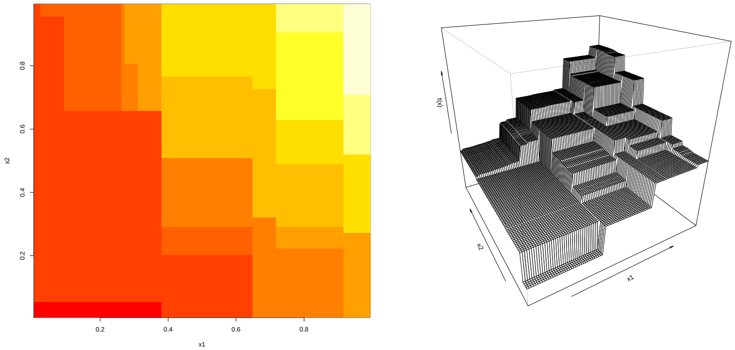
Here is the graph of the fit.





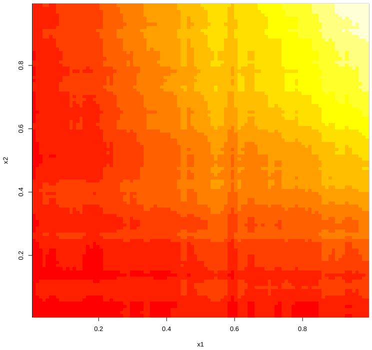
The fit is not terrible, but there are some aspects of the fit which violate monotonicity.

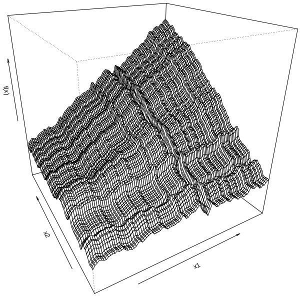
Here is the graph of the fit with the monotone constraint:



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

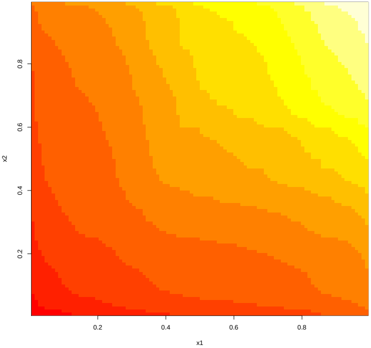
Here is the unconstrained BART fit:

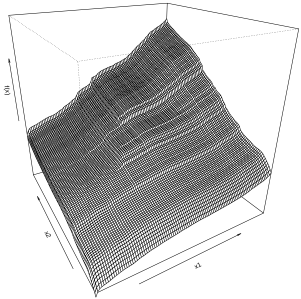




Much better (of course) but not monotone!

And, finally, the constrained BART fit:





*Not Bad!*

*Same method works with any number of x's!*

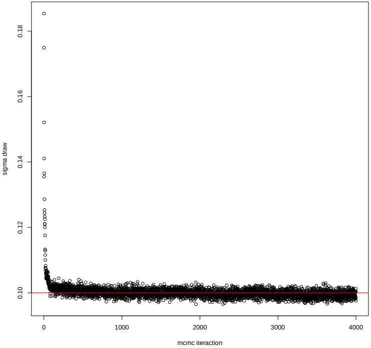
# 6. A 5-Dimensional Example

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

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

For various values of $\sigma$, we simulated 5,000 observations.

Here are the MCMC draws of sigma:

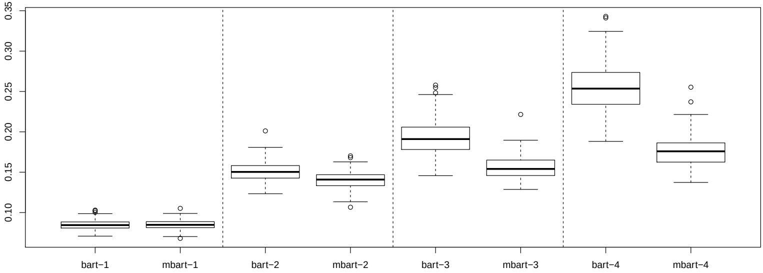


The horizontal (red) line is drawn at the true value.

We see that the sampler quickly burns in and then varies about the true value.

# RMSE improvement over unconstrained BART

| $\sigma$ | Monotone BART RMSE | Unconstrained BART RMSE | Percentage Increase |
|---|---|---|---|
| 0.5 | 0.14 | 0.16 | 14% |
| 1.0 | 0.17 | 0.28 | 65% |



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

# 7. Discovering Monotonicity with mBART

Suppose we don't know if $f(x)$ is monotone up, monotone down or even monotone at all.

Of course, a simple strategy would be simply compare the fits from BART and mBART.

Good news, we can do much better than this!

As we'll now see, mBART can be deployed to simultaneously estimate all the monotone components of $f$.

With this strategy, monotonicity can be discovered rather than imposed!

# The Monotone Decomposition of a Function

To begin simply, suppose $x$ is one-dimensional and $f$ is of bounded variation.

> *Any such $f$ can be uniquely written (up to an additive constant) as the sum of a monotone up function and a monotone down function*
>
> $$f(x) = f_{up}(x) + f_{down}(x)$$
>
> *where*
>
> - *when $f(x)$ is increasing, $f_{up}(x)$ increases at the same rate and is flat otherwise,*
> - *when $f(x)$ is decreasing, $f_{down}(x)$ decreases at the same rate and is flat otherwise.*

# The Discovery Strategy with mBART

Key Idea: To discover the monotone decomposition of $f$, we simply treat $f(x)$ as a two-dimensional function in $R^2$,

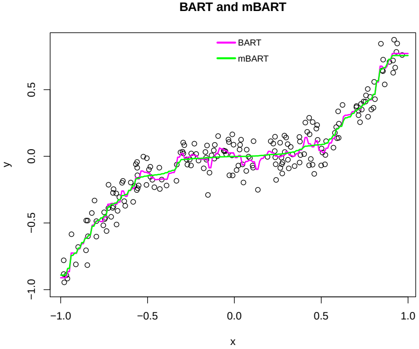
$$f(x) = f^*(x, x) = f_{up}(x) + f_{down}(x).$$

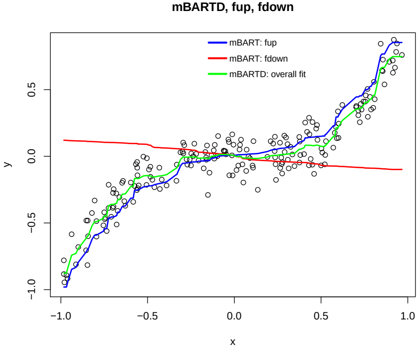
Letting $x_1 = x_2 = x$ be duplicate copies of $x$, we apply mBART to estimate $f^*(x_1, x_2)$

- constrained to be monotone up in the $x_1$ direction, and
- constrained to be monotone down in the $x_2$ direction.

Let's look at some illuminating one-dimensional examples.

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

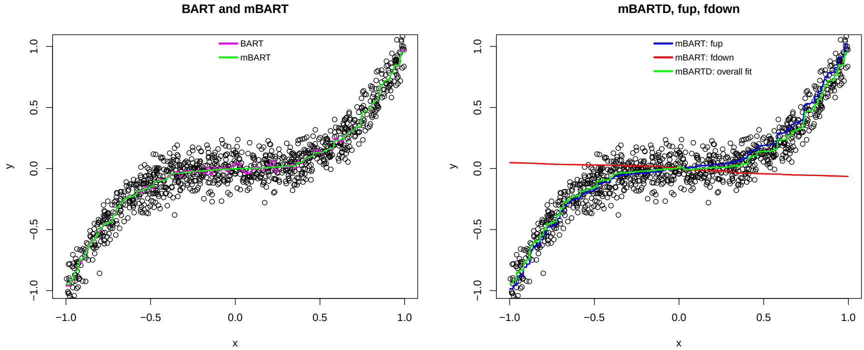




Note that $\hat{f}_{down} \approx 0$ (the red in the right plot), as we would expect when $f$ is monotone up.

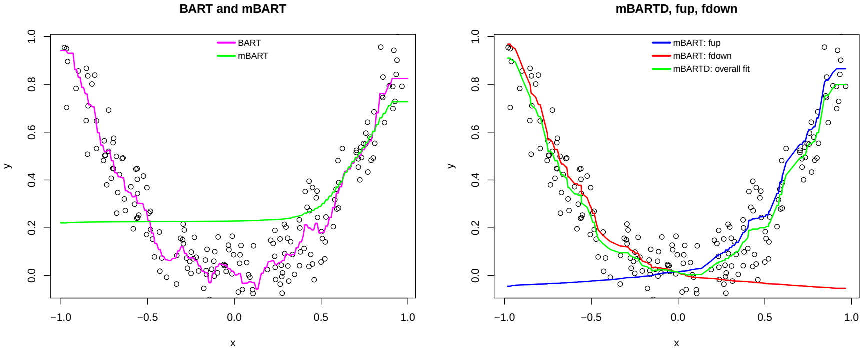
Note: mBART looks nicer than BART, not restricted!

As the sample size is increased from 200 to 1,000, $\hat{f}_{down}$ gets even flatter.



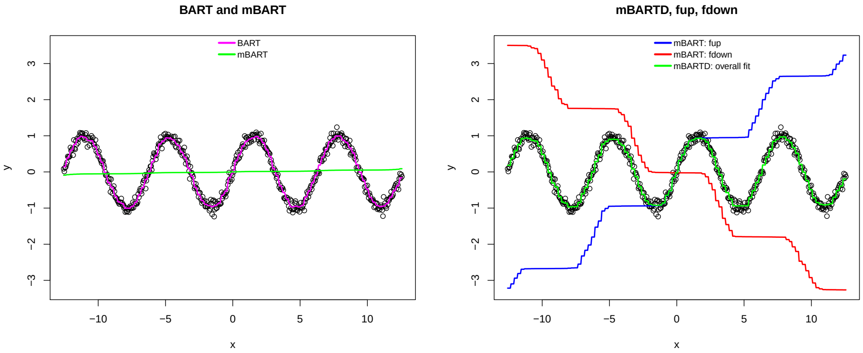
*Suggests consistent estimation of the monotone components!!*

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



- On the left, BART is good, but simple mBART is not.
- On the right, $\hat{f}_{up}$ and $\hat{f}_{down}$ are spot on.
- And mBARTD $= \hat{f}_{up} + \hat{f}_{down}$ seems even better than BART!

Example: Suppose $Y = sin(x) + \epsilon$.



- BART is good, but simple mBART reveals nothing.
- $\hat{f}_{up}$ and $\hat{f}_{down}$ have discovered the monotone decomposition.
- And mBARTD $= \hat{f}_{up} + \hat{f}_{down}$ is great too.

*To extend this approach to multidimensional $x$, we simply duplicate each and every component of $x$ !!!*

# 8. 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.

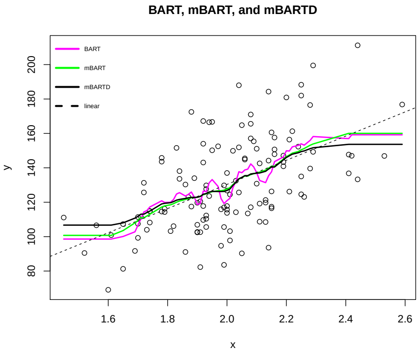
x: three neighborhoods, thousands of square feet, brick or not.

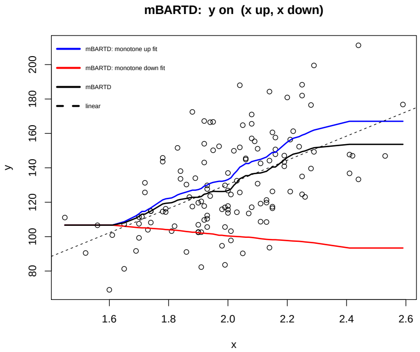
```
Call:
lm(formula = price ~ nbhd + size + brick, data = hdat)

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
```

If the linear model is correct, we are monotone up in all three variables.

Remark: For the linear model we have to dummy up *nbhd*, but for BART and mBART we can simply leave it as an ordered numerical categorical variable.

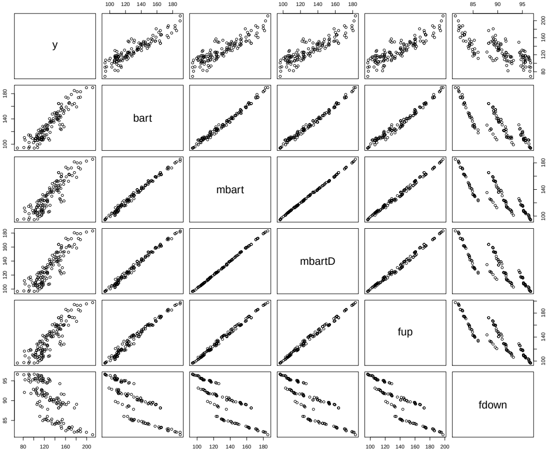
Just using $x = \textit{size of the house}$, $y = \textit{price}$ appears to be marginally increasing in *size*. ($\hat{f}_{down} \approx 0$).





mBART and mBARTD seem much better than BART.

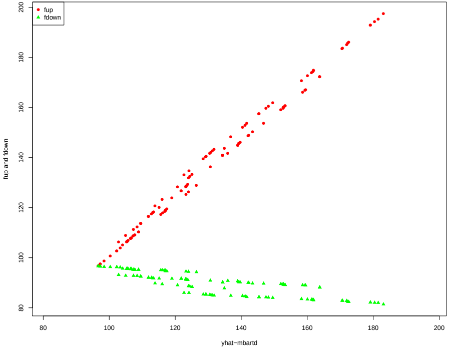
Using $x = (nbhd, size, brick)$, here are the relationships between the fitted values from various models.



Note the high correlation between mBART, mBARTD and $\hat{f}_{up}$.

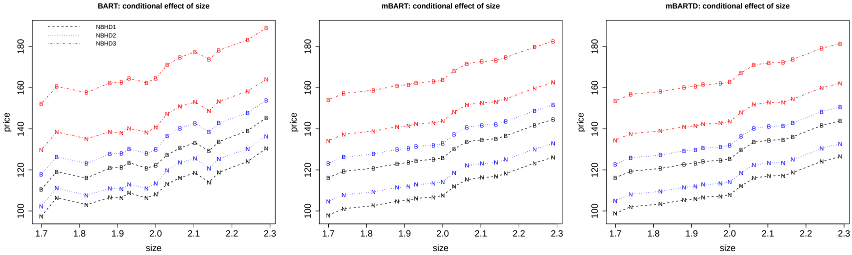
x axis: mBARTD $= \hat{f}_{up} + \hat{f}_{down}$.

y axis: red: $\hat{f}_{up}$, green: $\hat{f}_{down}$.



mBARTD $\approx \hat{f}_{up}$ suggests $f$ is multivariate monotonic !!!

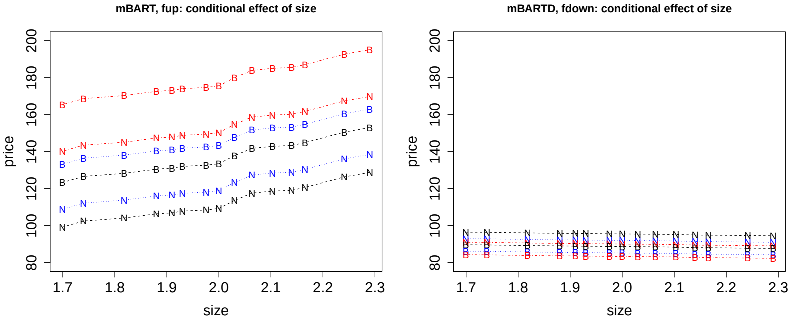
Let's now look at the effect of *size* conditionally on the six possible values of (*nbdh*, *brick*)



mBART and mBARTD *look very similar !!*

The conditionally monotone effect of *size* is becoming clearer!

And finally, the effect of *size* conditionally on the six possible values of $(nbdh, brick)$ via $\hat{f}_{up}$ and $\hat{f}_{down}$



$\hat{f}_{up}$ and mBARTD *look very similar !!*

Price is clearly conditionally monotone up in all three variables!

*By simultaneously estimating $\hat{f}_{up} + \hat{f}_{down}$, we have discovered monotonicity without any imposed assumptions!!!*

# Concluding Remarks

- mBARTD $= \hat{f}_{up} + \hat{f}_{down}$ provides a new assumption free approach for the discovery of the monotone components of $f$ in multidimensional settings.
- Discovering such regions of monotonicity may of scientific interest in real applications.
- We have used informal variable selection to identify the monotone components here. More formal variable selection can be used in higher dimensional settings.
- As a doubly adaptive shape-constrained regularization approach,
    - mBARTD will adapt to mBART when monotonicity is present,
    - mBARTD will adapt to BART when monotonicity absent,
    - mBARTD will be at least as good and maybe better, than the best of mBART and BART in general.

# Concluding Remarks

- The fully Bayesian nature of BART greatly facilitates extensions such as mBART, mBARTD and many others.

- Despite its many compelling successes in practice, theoretical frequentist support for BART only now beginning to appear.

- For example, 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.

- Monotone BART paper is available on Arxiv. Software for mBART is available at https://bitbucket.org/remcc/mbart.

Thank You!