More BART

STAT8810, Fall 2017

M.T. Pratola

November 12, 2017

Today

Heteroscedastic BART
Influence in BART Models
More on Tree Moves

Recall BART

- Recall our BART model:

  $$y(\mathbf{x}) = f(\mathbf{x}) + \sigma\epsilon$$

  where $\epsilon \sim N(0, 1)$.
- BART models the unknown mean function $f(x)$ as a sum of regression trees,

  $$f(\mathbf{x}) = \sum_{j=1}^{m} g(\mathbf{x}; T_j, M_j)$$

- Places a prior on each regression tree so that each tree makes a small contribution to the overall fit.
- The prior "regularizes" the model so that it does not overfit.
- We can quantify uncertainties by drawing from the posterior of the trees, and hence f, using an effective MCMC.

Heteroscedastic BART

- A commonly violated assumption of regression modeling is the homoscedastic, constant-variance assumption.
- Lots of flexible models for capturing complex mean behaviours available in the literature. But complex variance behaviour, not so much.
- Can we devise an equally flexible and easy-to-use heteroscedastic regression tree model?

Heteroscedastic BART

- The HBART model is
$$Y(\mathbf{x}) = f(\mathbf{x}) + s(\mathbf{x})\epsilon$$
where again $\epsilon \sim N(0,1)$.
- The unknown mean function is modeled as a sum of regression trees,
$$f(\mathbf{x}) = \sum_{j=1}^{m} g(\mathbf{x}; T_j, M_j)$$
- The unknown variance function is modeled as a product of regression trees,
$$s(\mathbf{x})^2 = \prod_{k=1}^{m'} h(\mathbf{x}; T'_k, M'_k)$$
- As before, the terminal node parameters for the mean trees are just scalars given conjugate Normal priors.
- While the terminal node parameters for the variance trees are just scalars given conjugate scaled-inverse-chi-squared priors.

HBART Model

- The HBART posterior is factored as

$$
\begin{aligned}
& \pi(\{T_j, M_j\}_{j=1}^{m}, \{T_j', M_j'\}_{j=1}^{m\prime} | \mathbf{Y}) \\
\propto\ & L(\mathbf{Y} | \{T_j, M_j\}_{j=1}^{m}, \{T_j', M_j'\}_{j=1}^{m\prime}) \\
& \times \prod_{j=1}^{m} \pi(T_j)\pi(M_j | T_j) \times \prod_{j=1}^{m'} \pi(T_j')\pi(M_j' | T_j')
\end{aligned}
$$

HBART Posterior

where

$$\pi(M_j | T_j) = \prod_k \pi(\mu_{jk})$$

and

$$\pi(\mu_{jk}) \sim N(0, \tau^2)$$

and also

$$\pi(M'_l | T'_l) = \prod_k \pi(s^2_{lk})$$

where

$$\pi(s^2_{lk}) \sim \chi^{-2}(\nu', \lambda').$$

HBART MCMC Algorithm

Our algorithm will perform the following basic steps:

1. For $j = 1, \ldots, m$
   1.1 Draw $T_j|\cdot$
   1.2 Draw $M_j|T_j, \cdot$
2. For $l = 1, \ldots, m'$
   2.1 Draw $T'_l|\cdot$
   2.2 Draw $M'_l|T'_l, \cdot$

We repeat these steps for a large number of iterations.

Drawing $M_j | T_j, \cdot$

- The likelihood function for the kth terminal node of the jth mean tree is

$$L(\mu_{jk}|\cdot) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi} s(\mathbf{x}_i)} exp \left(\frac{(r_i - \mu_{jk})^2}{s^2(\mathbf{x}_i)} \right)$$

where n is the number of observations mapping to terminal node k of mean tree j, and

$$r_i = y_i - \sum_{q \neq j} g(\mathbf{x}_i; T_q, M_q).$$

- Given our conjugate prior, the full conditional is

$$\pi(\mu_{jk}|\cdot) \sim N \left(V \sum_{i=1}^{n} \frac{r_i}{s^2(\mathbf{x}_i)}, V \right)$$

where $V^{-1} = \frac{1}{\tau^2} + \sum_{i=1}^{n} \frac{1}{s^2(\mathbf{x}_i)}$.

Drawing $T_j|\cdot$

- Sampling the mean trees takes place using our usual suite of tree-moves, namely birth-death.
- In order to apply such dimension-changing moves, we need the integrated likelihood to avoid the dimension-matching approach to RJ-MCMC.
- This can be shown to be

$$\int L(\mu_{jk}|\cdot)\pi(\mu_{jk})d\mu_{jk}$$

$$\propto \quad \left(\tau^2 \sum_{i=1}^{n} \frac{1}{s^2(\mathbf{x}_i)} + 1\right)^{-1/2} exp\left(\frac{\frac{\tau^2}{2}\left(\sum_{i=1}^{n}\frac{r_i}{s^2(\mathbf{x}_i)}\right)^2}{\tau^2\sum_{i=1}^{n}\frac{1}{s^2(\mathbf{x}_i)}+1}\right).$$

Drawing $M_j' | T_j', \cdot$

- The likelihood function for the kth terminal node of the lth variance tree is

$$L(s_{lk}^2 | \cdot) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi} s_{lk}} exp \left(-\frac{e_i^2}{2s_{lk}^2} \right)$$

where

$$e_i^2 = \frac{\left(y(\mathbf{x}_i) - \sum_{j=1}^{m} g(\mathbf{x}_i; T_j, M_j) \right)^2}{s_{-l}^2(\mathbf{x}_i)}$$

and

$$s_{-l}^2(\mathbf{x}_i) = \prod_{q \neq l} h(\mathbf{x}_i; T_q', M_q').$$

Drawing $M'_j | T'_j, \cdot$

- It can then be shown that the full conditional is

$$s^2_{lk} | \cdot \sim \chi^{-2} \left(\nu' + n, \frac{\nu' \lambda'^2 + \sum_{i=1}^{n} \frac{e_i^2}{s^2_{-l}(\mathbf{x}_i)}}{\nu' + n} \right).$$

Drawing $T_j'|\cdot$

- Sampling the variance trees also uses our usual suite of tree-moves, namely birth-death.
- The integrated likelihood can be shown to be

$$\int L(s_{lk}^2|\cdot)\pi(s_{lk}^2)ds_{lk}^2 = \frac{\Gamma\left(\frac{\nu'+n}{2}\right)\left(\frac{\nu'\lambda'^2}{2}\right)^{\nu'/2}}{(2\pi)^{n/2}\prod_{i=1}^{n} s_{-l}(\mathbf{x}_i)\Gamma(\nu'/2)(\nu'\lambda'^2+\sum_{i=1}^{n} e_i^2)^{\frac{\nu'+\lambda}{2}}}$$

Calibrating the mean prior

- As in regular BART, we assume the data are scaled to have mean 0.
- Then the prior on the μ_{jk}s implies a prior on the mean function of $f(\mathbf{x}) \sim N(0, m\tau^2)$.
- Choose the parameter k we construct the prior by setting

$$\tau = \frac{y_{max} - y_{min}}{2k\sqrt{m}}$$

- For BART the default was $k = 2$ which implied a 95% prior probability that the true mean function lies between the observed data range.
- In HBART we tend to increase to $k = 5$ or $k = 10$, or tuning it via cross-validation.

Calibrating the variance prior

- In BART with homoscedastic variance σ^2, we had the prior $\sigma^2 \sim \chi^{-2}(\nu, \lambda)$
  - we selected a shape via ν and then calibrated λ so that a high quantile of the prior matched the observed standard deviation of the data.
- In HBART, we are motivated by being able to have a simple prior specification much like BART.
- Consider the mean of the prior under homoscedasticity,

$$E[\sigma^2] = \lambda \frac{\nu}{\nu - 2}.$$

- And the mean under the assumed a priori independence of the heteroscedastic model is

$$E[s(\mathbf{x})^2] = \prod_{l=1}^{m'} E[s_l(\mathbf{x})^2] = \lambda'^{m'} \left(\frac{\nu'}{\nu' - 2} \right)^{m'}$$

Calibrating the variance prior

- This suggests matching the two components of the priors, resulting in

$$\lambda' = \lambda^{1/m'}$$

and

$$\nu' = \frac{2}{1 - \left(1 - \frac{2}{\nu}\right)^{1/m'}}$$

- Idea is to make specifying the variance prior in HBART as simple as in regular BART, and let the data drive the solution towards heteroscedasticity.
- Even though we are only matching the means of these two priors, we see that actually their density curves match fairly well.

Calibrating the variance prior

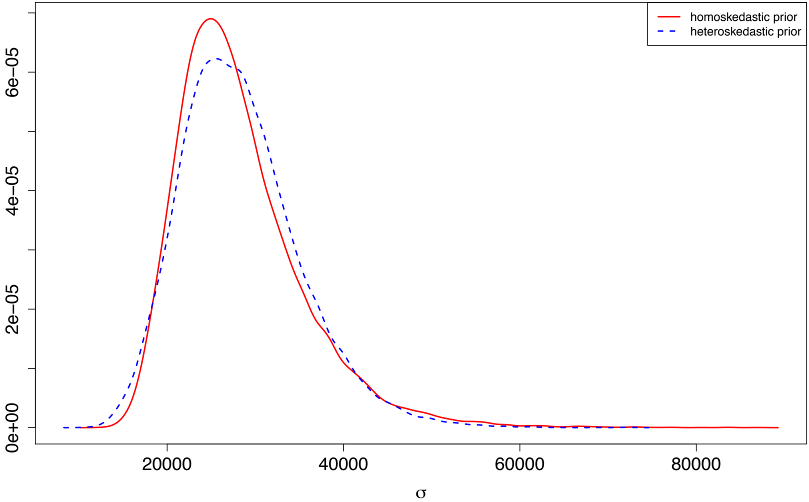


Figure 1: Variance prior under homo and hetero BART

Example: simulated data

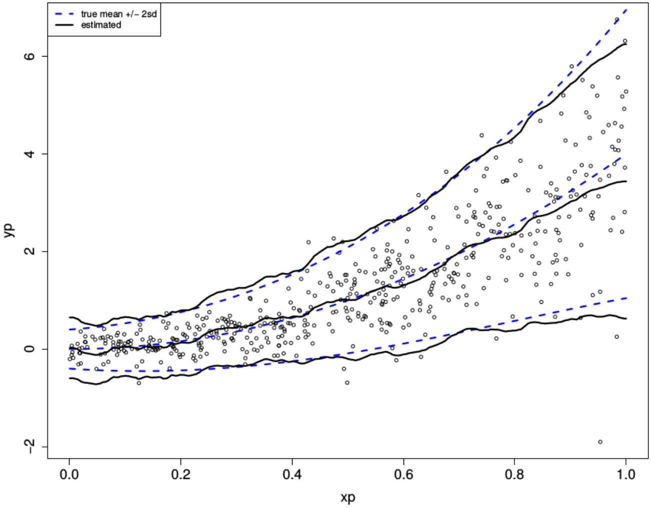


Figure 2: Simulated data where mean and sd are quadratic and

Example: simulated data

- In a low-dimensional (1D) example like this, and with enough data, it may be relatively "obvious" that heteroscedasticity is present.
- What about higher dimensions?
- Given the $\mathbf{x}_i$, sort the $\hat{s}(\mathbf{x}_i)$ where $\hat{s}$ might be the posterior mean of the fitted HBART model.
- Then plot the 95% quantile intervals for the posterior draws of $s(\mathbf{x}_i)$ from HBART versus the $\hat{s}(\mathbf{x}_i)$.
- And plot the posterior estimate of σ from BART on the same plot.
- We call this plot the H-evidence plot.

Example: simulated data

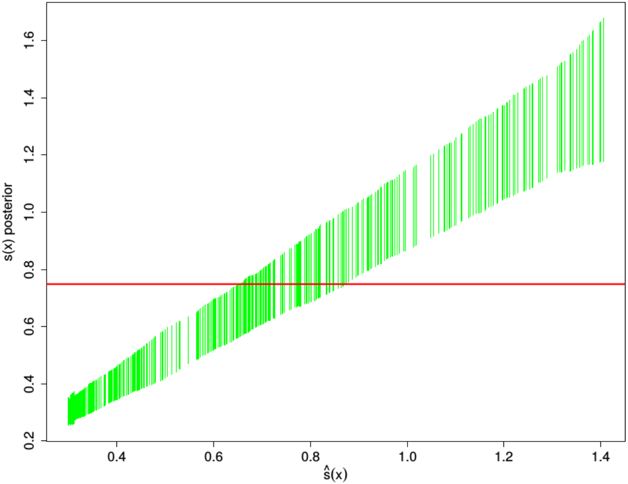


Figure 3: H-evidence for simulated example

Example: simulated data

- Here the H-evidence plot is fairly convincing that heteroscedasticity is present.
- We can also plot the posterior mean function and posterior standard deviation function from the fitted HBART model.

Example: simulated data

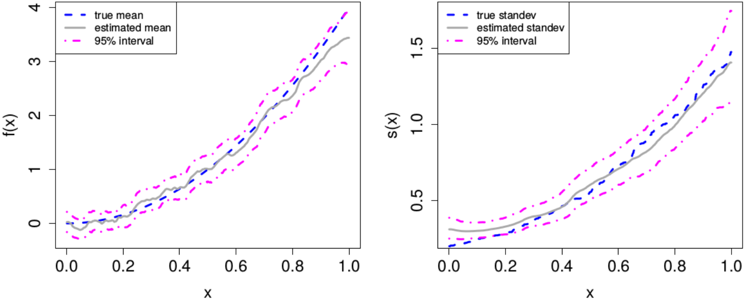


Figure 4: Fitted HBART model

Example: simulated data

- How do we asses the quality of the models fit?
- Usually we would just calculate MSPE on a held-out test set of data. But here that isn't enough.
- Given a test set, for every posterior sample of f_d, s_d let

  $$y_{id} = f_d(\mathbf{x}_i) + s_d(\mathbf{x}_i) z_d$$

  where $z_d \sim N(0,1)$.
- Then for each i compute the percentile of the test data y_i in the draws y_{id}.
- If the model is correct, these percentiles should appear as draws from the uniform distribution
  - do a qq-plot!

Example: simulated data

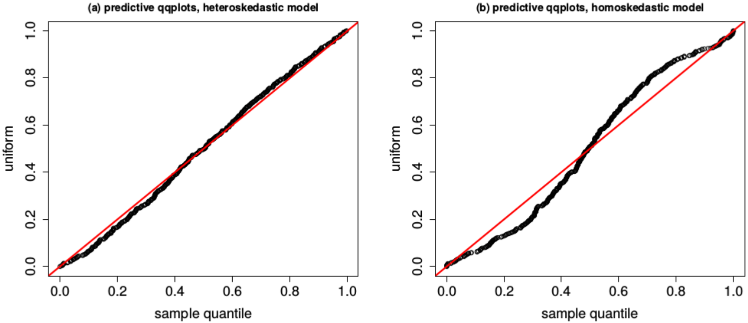


Figure 5: qq-plots for HBART and BART

Example: simulated data

- The qq-plot serves as a visual diagnostic of our model.
- One can also look at distributional distance metrics to get a scalar representation of the difference.
- Such a scalar metric is useful for calibrating important hyperparameters (namely k) in the model rather than using MSPE to calibrate hyperparameters.
- Idea is we are now interested in checking the *distributional fit* rather than just the fit of the mean.
    - well, in statistics we were always interested in this, but it was easier in the homoscedastic world where we can just look at MSPE and a qq-norm of the residuals.

Example: cars data

- Real data taking from used car sales from 1994-2013.
- Dataset consists of $n = 1000$ observations of 18 predictor variables.
- Response is `price`.
- Continuous predictors are `mileage, year`.
- Categorical predictors are `trim, color, displacement, isOneOwner`.

Example: cars data

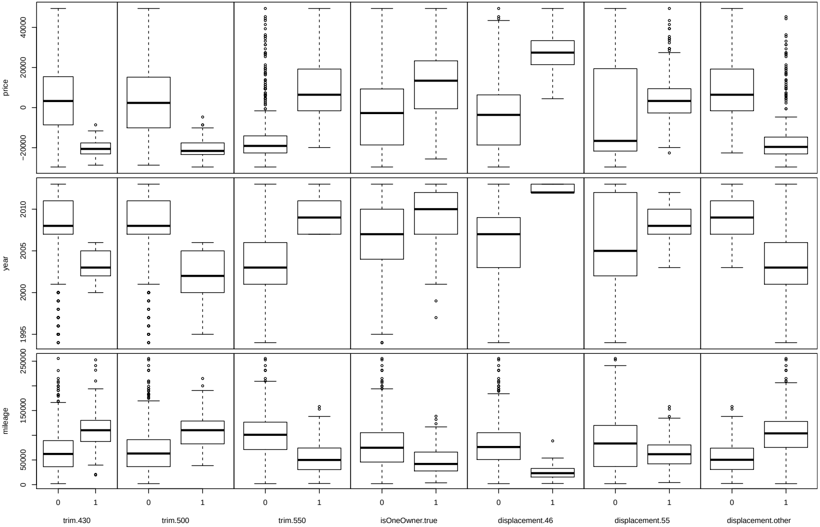


Figure 6: Discrete variables from Cars data

Example: cars data

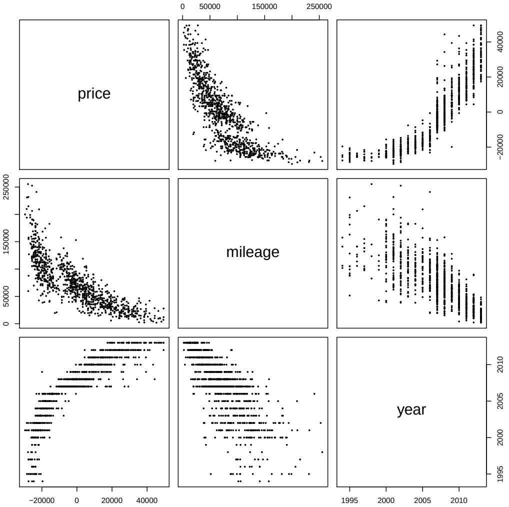


Figure 7: Continuous variables from Cars data

Example: cars data

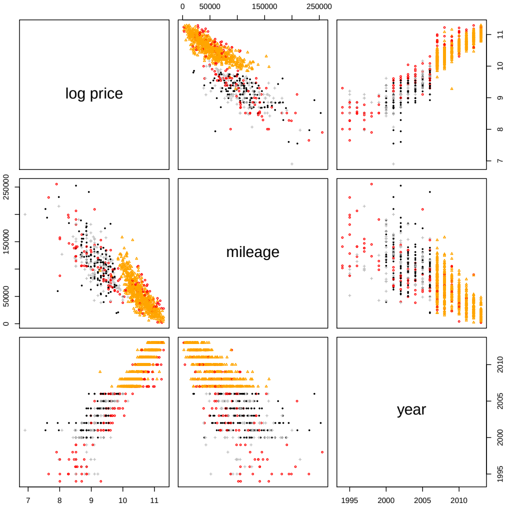


Figure 8: Log transform? Continuous variables from Cars data by trim

Example: cars data

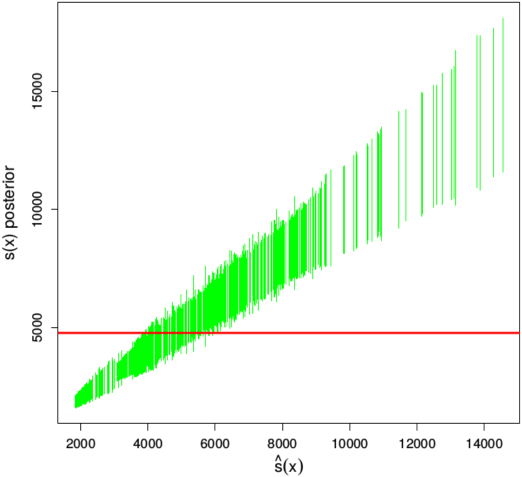


Figure 9: H-evidence plot for cars data

Example: cars data

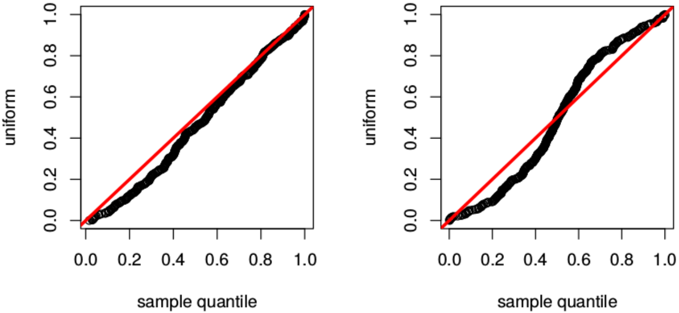


Figure 10: qq plot for cars data

Example: cars data

- The qq-plot suggests we are doing better than the homoscedastic model although not perfect (this dataset likely has skewness as well).
- The cool thing about our flexible non-parameteric heteroscedastic BART model is the capability to explore more complex relationships involving the first two moments of our data.
- For example, how does the *level* and *volatility* of our response change with the predictors?
- Which variables are important for the mean function? Which ones for the variability? Are they the same?
- And we can do this while quantifying the uncertainties.

Example: cars data

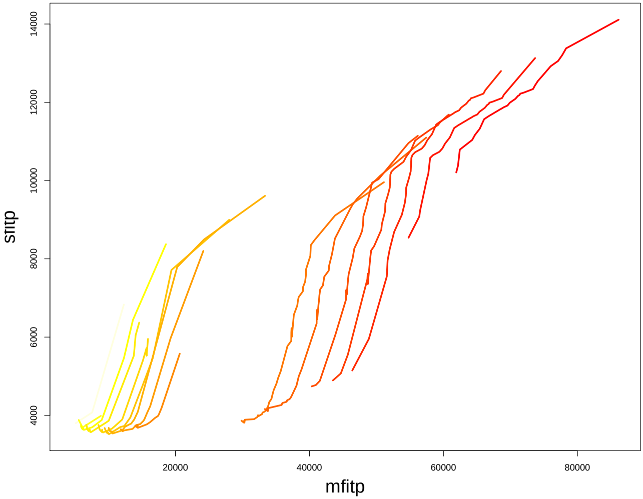


Figure 11: Posterior mean f versus s color coded by `year`

Example: cars data

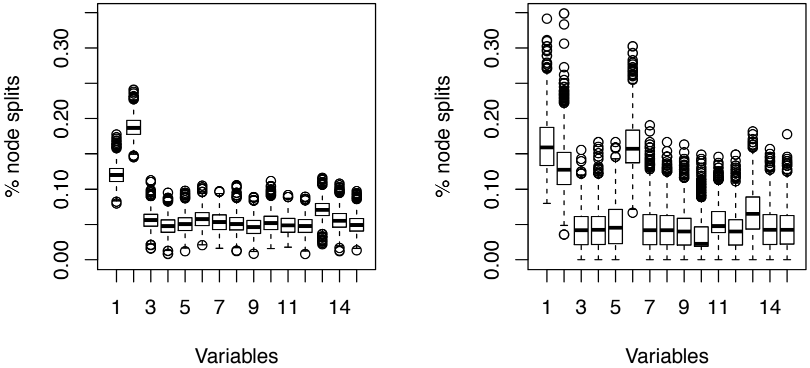


Figure 12: Variability activity information for f and s. `mileage` and `year` important for f while `trim.other` also important for s.

Leverage and Influence in BART

- Diagnostics are an important part of applied regression modeling. One of the more popular ones in linear regression is influence.
- A popular measure of influence is Cook's Distance:

$$D_i = \frac{e_i^2}{p\hat{\sigma}^2}\frac{h_{ii}}{(1 - h_i i)^2}$$

- A large value of D_i means that observation i unduly influences the fit of our model.

Cook's D for Single Tree Model

- For a single tree model, conditional on the tree being fixed, known, we can express our model as

$$y(\mathbf{x}_i) = \sum_{k=1}^{B} I_k(\mathbf{x}_i)\mu_k + \epsilon_i$$

  where

  - $I_j(\mathbf{x}_i) = 1$ if observation $\mathbf{x}_i$ is in bottom node k; 0 otherwise
  - μ_k is the mean level assigned to botto node k
  - $\epsilon_i \sim N(0, \sigma^2)$.

- In other words, conditional on the tree, we can write our model in the form of an MLR regression. Our predictor variables are simply indicators indicating which bottom node an observation maps to.

Cook's D for Single Tree Model

- Taking a simple approach, we compute Cook's D for this conditional setup. The diagonal entry of the hat matrix is then

$$\begin{aligned} h_i i &= \mathbf{x}_i^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i \\ &= \frac{1}{n[i]} \end{aligned}$$

  where $n[i]$ is the total number of observations mapping to the terminal node that observation i belongs to.

- Cook's D can then be expressed as

$$D_i = \frac{1}{B}\frac{e_i^2}{\sigma^2}\frac{n[i]}{(1-n[i])^2}$$

  where we take σ^2 to be a draw from the MCMC, B is the number of terminal nodes in the tree and e_i is the residual.

Cook's D for Single Tree Model

- In practice, we will compute the D_i at each iteration of the MCMC, so we have a posterior sample of Cook's Distances.
- We can then use this to get a sense of which observations suggest abnormal levels of influence relative to most observations.

Example

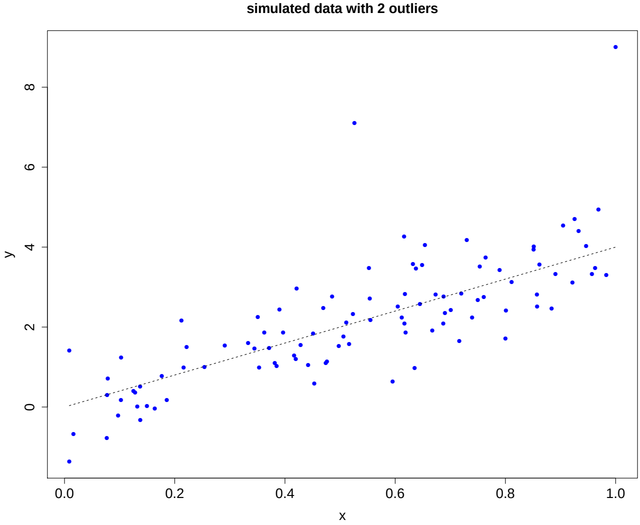


Figure 13: Simulated SLR model with 2 influential observations.

Example

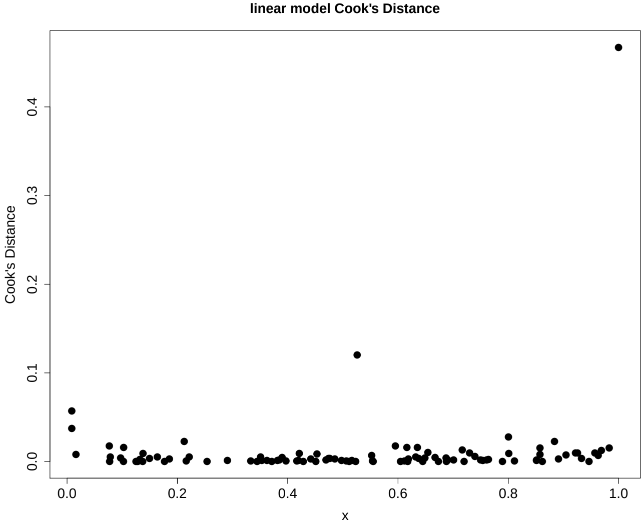


Figure 14: Regular MLR Cook's Distance

Example

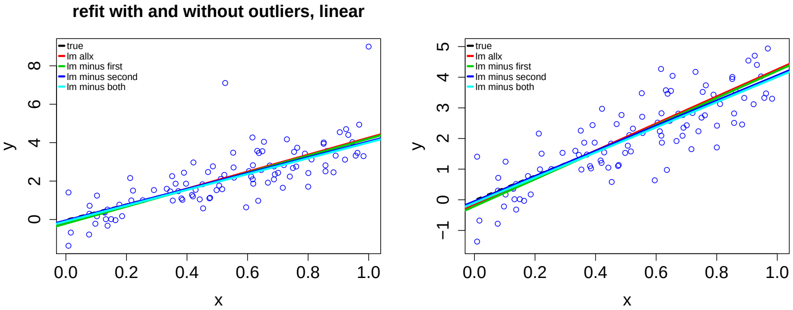


Figure 15: Effect on SLR fit to data

Example

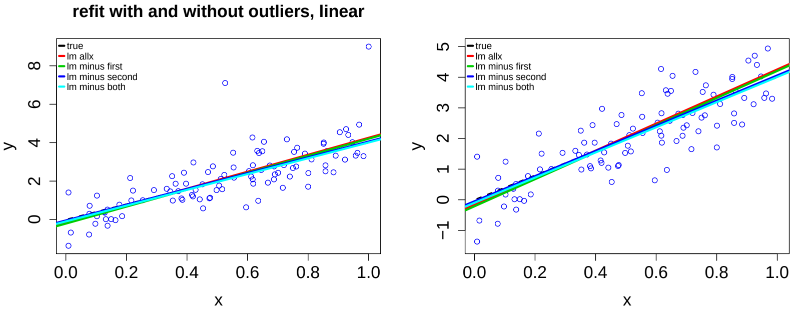


Figure 16: Effect on SLR fit to data

Example

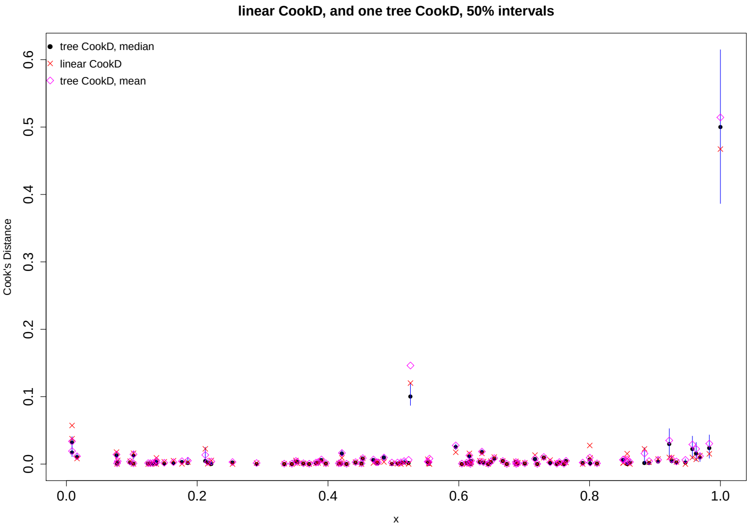


Figure 17: Cook's D for single-tree model

Example

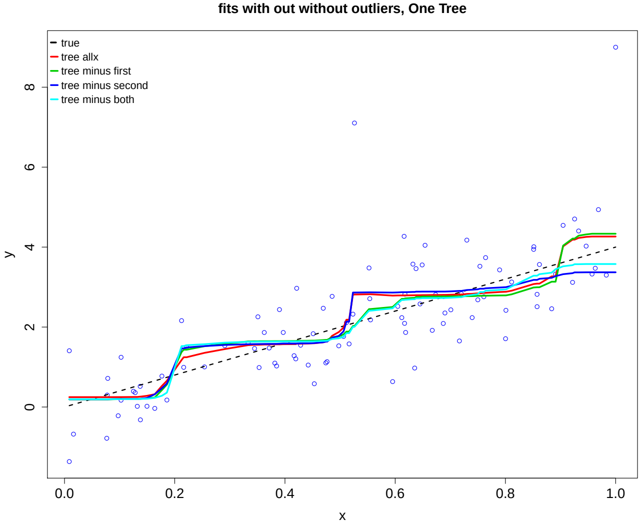


Figure 18: Effect of influential observations on single-tree fit

Cook's D for BART

- Extend to the BART setup in a simple way.
- Take the partial residuals for tree j,

$$\tilde{y}(\mathbf{x}_i) = y(\mathbf{x}_i) - \sum_{k \neq j} g(\mathbf{x}_i; T_k, M_k)$$

- Use the partial residual $\tilde{y}$ as our data to compute Cook's D for tree j
- Average these overall all m trees in our BART model.

Example

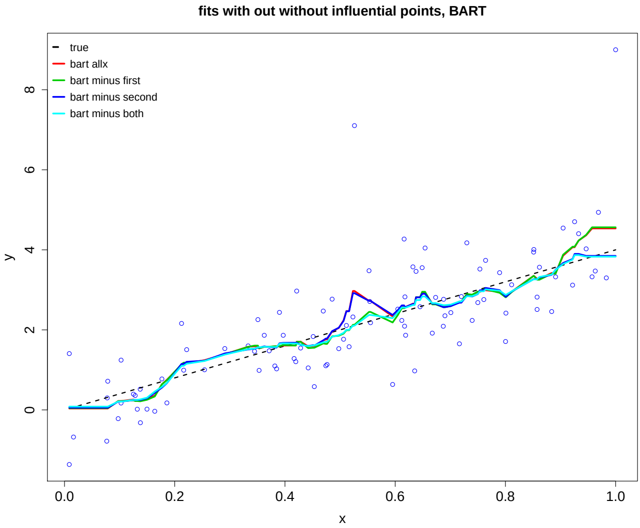


Figure 19: Cook's D for BART

Example

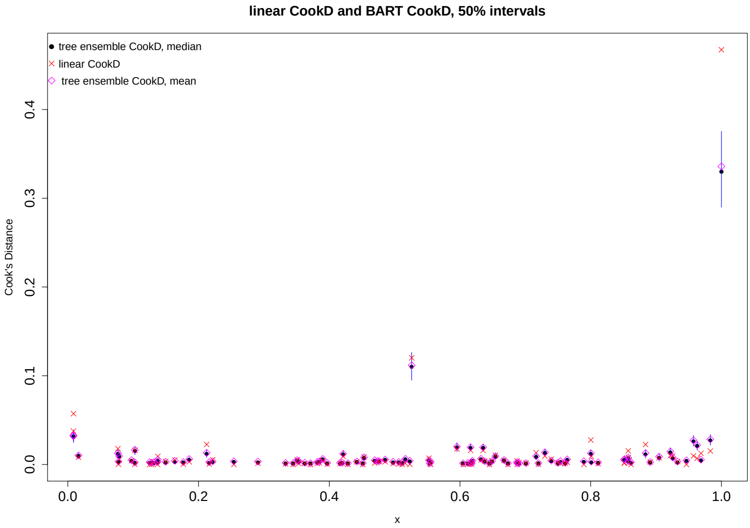


Figure 20: Effect of influential observations on BART fit

Cars Example

- What does this look like for our Cars data?
- Considered Cook's D for single-tree model, BART model and Heteroscedastic BART model.

Cars Example

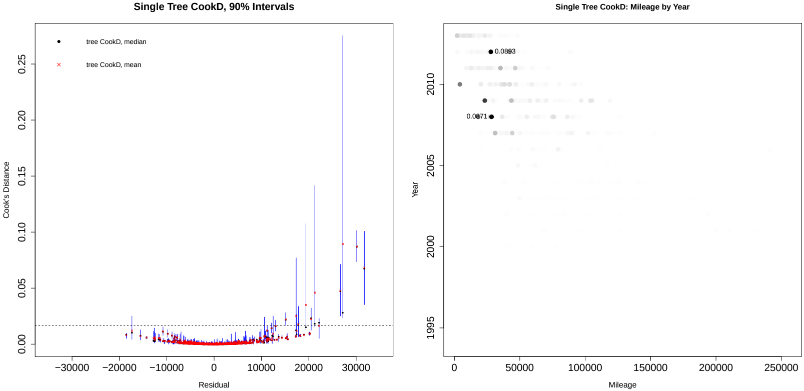


Figure 21: Cook's D for Cars data, single tree homoscedastic model

Cars Example

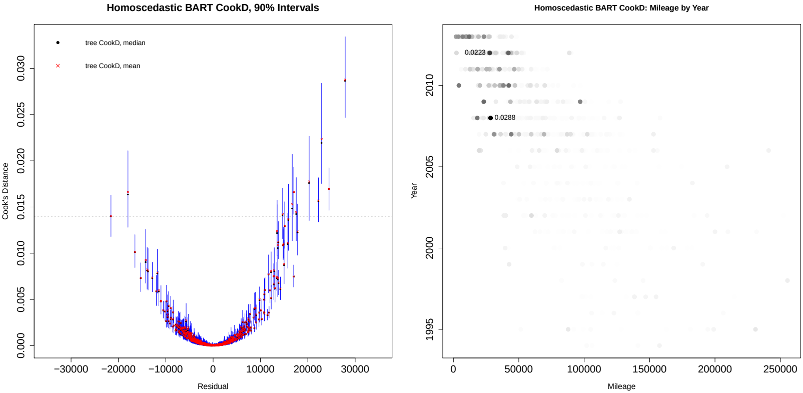


Figure 22: Cook's D for Cars data, homoscedastic BART model with $m = 200$ trees

Cars Example

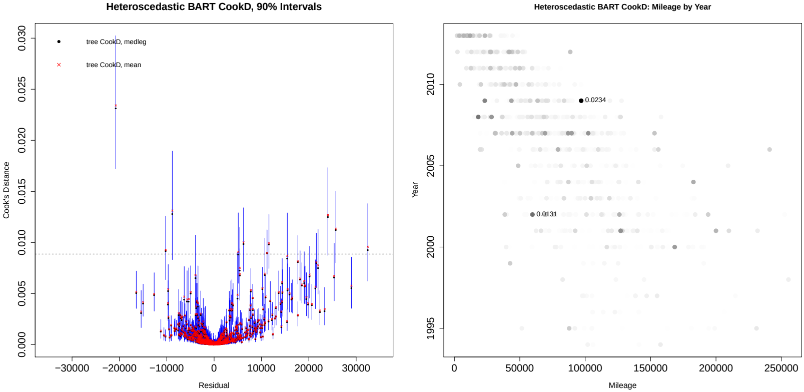


Figure 23: Cook's D for Cars data, heteroscedastic BART model with $m = 200$ mean trees

Sampling Trees

- So far we have just considered birth/death moves for exploring tree-space.
- Birth/death are important for growing/pruning the tree to reach a model of appropriate complexity for the data observed.
- However, the tree is made up of many parameters which are not explored at each iteration of the MCMC if we only perform birth/death.
- The existing literature also would perform change/swap moves (we will ignore the restructure move).

Change and Swap

- Change aims to explore the variables or variable/cutpoints that make up a tree.
- q_c proposes to change internal node η_i's rule from $v < c$ to $v' < c'$.
- v' and c' are drawn from the prior.
- Or, v' and c' are drawn from the prior conditional on the ancestral part of the tree above node η_i.
- Swap aims to explore the structure of the existing tree by taking a parent/child pair of internal nodes and exchanging their $v < c$, $v' < c'$ rules.

Sampling Trees

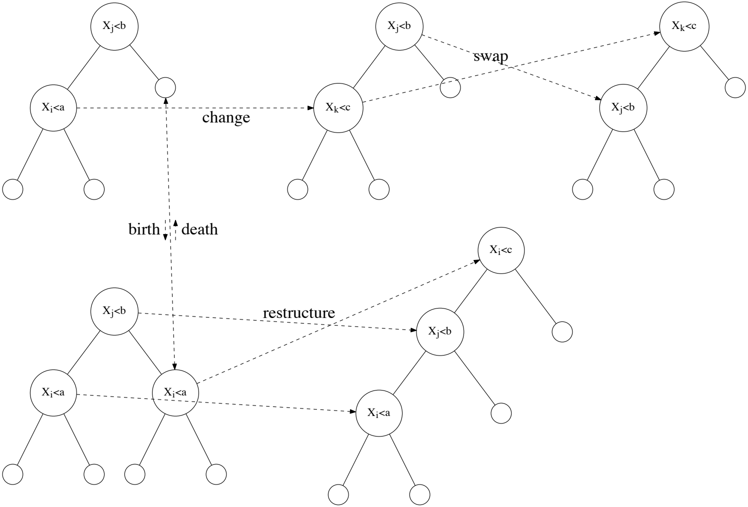


Figure 24: Tree proposals

Change and Swap

- Both can be relatively inefficient. Especially in the (usual) case where each terminal node must contain a minimum number of observations.
- Change ignores the tree constraint from nodes *below* the current node η_i.
- Swap essentially assumes the rest of the “function” described by the tree *below* the current node η_i will be unaffected by the swap operation.
- One can easily imagine examples where swap would lead to empty terminal nodes which, by default, would be rejected proposals.

Sampling Trees

- Let's be a bit more clear about our definition of a tree, T.
- T consists of internal nodes, $\{\eta_i\}$ for $i = 1, \ldots, |T|$.
- Each internal node is made up of a variable, cutpoint pair: v_i, c_i.
- And we also have a *tree structure*. Let's call this τ.
- Sampling T might more clearly be thought of involving the following components.
  - Sampling τ (e.g. birth/death)
  - Sampling $v_i|\tau, \{v_{-i}, c_{-i}\}, c_i$
  - Sampling $c_i|\tau, \{v_{-i}, c_{-i}\}, v_i$

Sampling Cutpoints

- Exploring cutpoints is important. This is the part of the model that captures “wiggly” response surfaces versus “flat” response surfaces.
- Our proposal distribution for cutpoints at node η_i should account for the existing tree structure, $T \setminus c_i$.
- Let $C^{v_i}_{p(i)}$ be the collection of all cutpoints used in nodes ancestral to η_i that split on variable v_i.
- Let $C^{v_i}_{l(i)}$ be the collection of all cutpoints used in nodes that are left descendants of η_i that split on variable v_i.
- Let $C^{v_i}_{r(i)}$ be the collection of all cutpoints used in nodes that are right descendants of η_i that split on variable v_i.

Sampling Cutpoints

- Factor the ancestral nodes into “left” ancestors, $C^{v_i}_{p_l(i)}$ and “right” ancestors, $C^{v_i}_{p_r(i)}$.
- A **left ancestor** is an ancestor of η_i such that η_i is on the ancestor node’s right subbranch.
- A **right ancestor** is an ancestor of η_i such that η_i is on the ancestor node’s left subranch.

Sampling Cutpoints

- Assume that the predictors are scaled to $[0, 1]$.
- Then, a (discrete) uniform proposal for the cutpoint at η_i that is consistent with $T \setminus c_i$ is

$$c_i' \sim Unif(a, b)$$

where

$$a = max\left(0, max(C^{v_i}_{p_l(i)}), max(C^{v_i}_{l(i)})\right)$$

$$b = min\left(1, min(C^{v_i}_{p_r(i)}), min(C^{v_i}_{r(i)})\right)$$

Example

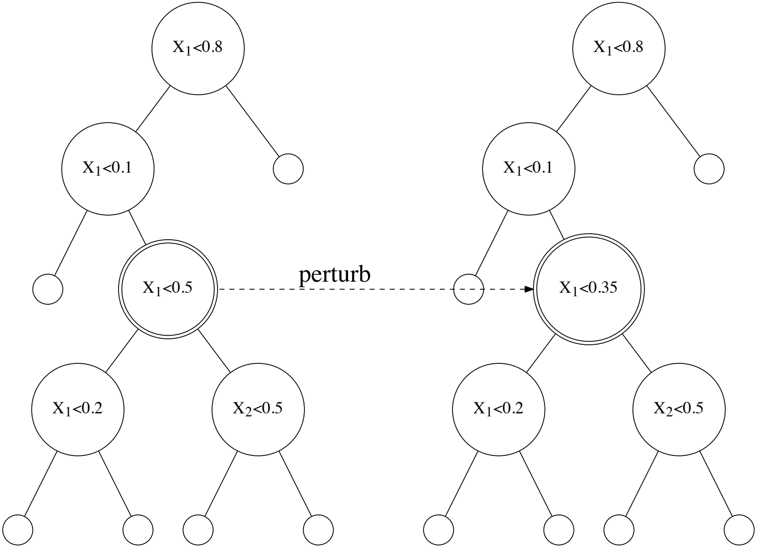


Figure 25: Propose a new cutpoint at node η_5

Sampling Variables

- Suppose our true mean response is a function like $E[Y(\mathbf{x})] = f_1(x_1) + f_2(x_2)$
- Suppose we have a collection of predictor variables in our dataset, $x_1, \ldots, x_p$.
- Does it make sense to try fitting the function (for example),

$$f_1(x_1) + f_2(x_4)?$$

Sampling Variables

- To improve proposals for the internal node's variables, we can use the idea of pre-conditioning.
- That is, we will prefer to propose transitions to some variable, say v', that is perhaps related, or correlated, with the current variable, v.
- Propose a transition to v' with probability proportional to

$$\frac{|Cor(v_i, v')|\mathcal{I}_{(a^{v'}, b^{v'}) \neq \{\}}}{\sum_j |Cor(v_i, v_j)|\mathcal{I}_{(a^{v_j}, b^{v_j}) \neq \{\}}}$$

Sampling Variables

- Here, $Cor(\cdot, \cdot)$ is some measure of relatedness between variables. Typical choices would be Pearson correlation or Spearman rank correlation.
- And $\mathcal{I}_{(a^{v'}, b^{v'}) \neq \{\}}$ is simply the indicator function that takes the value 1 when there are cutpoints available for the new variable v' at node η_i, 0 otherwise.
- For independent predictors, this proposal will rarely, if ever, propose transitions.
- For related predictors, this proposal will often propose transitions between related predictors.
- Minor detail: what if the predictors are negatively correlated?

Sampling the tree structure, τ.

- Proposals like change/swap are bad because they can easily lead to trees that are inconsistent with the birth/death process.
  - That is, you could propose a tree that you could never end up at via birth/death. In this sense, it is inadmissible.
- But besides birth/death, what can we do?
- Rotation is a more complex move that preserves consistency with the existing tree. That is, it will only propose trees that *are* admissible via the birth/death process.

Tree Rotation Proposals

- Let's call our proposal generating operator, $\mathcal{R}$. Then our tree proposal is generated by applying this to the existing tree,

$$T' = \mathcal{R}[T].$$

- We will look at *right-rotations* of an internal node η_i which is the left child of it's parent node $p(\eta_i)$.
- The rotation operation is the composition of simpler operations,

$$\mathcal{R}[T] = \mathcal{R}^L_{merge}\mathcal{R}^R_{merge}\mathcal{R}^L_{cut}\mathcal{R}^R_{cut}\mathcal{R}^R_{rot}[T]$$

where superscripts L, R denote left,right respectively.

Tree Rotation Proposals

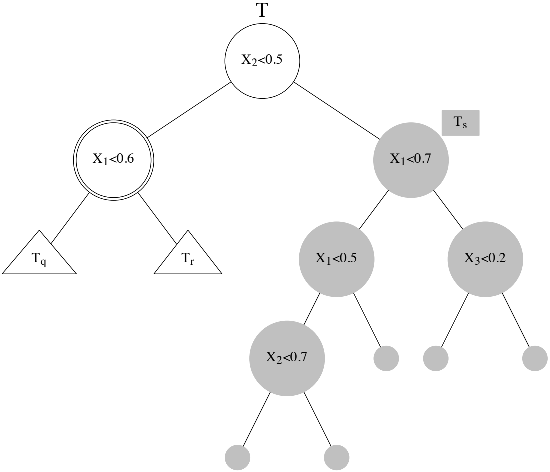


Figure 26: Our starting tree, T. We will generate the proposal by rotating at η_2.

Tree Rotation Proposals

- The $\mathcal{R}_{rot}^{R}$ part of generating the proposal involves swapping η_i's and $p(\eta_i)$'s rules.
- But also, $p(\eta_i)$'s rules end up in a new node that is the right child of $p(\eta_i)$.
- The existing subtree of $r(p(\eta_i))$, call it T_s, is now copied into two locations, as we'll see in the next figure.

Tree Rotation Proposals

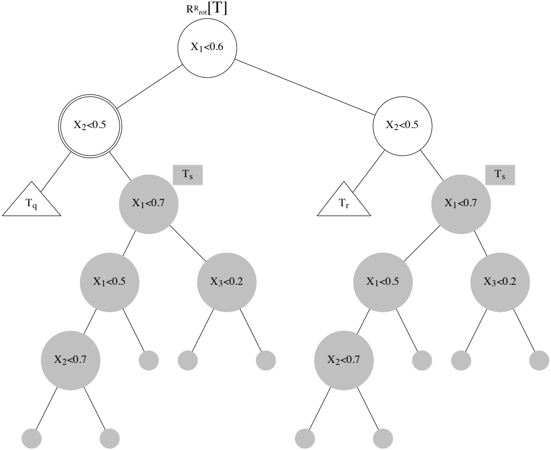


Figure 27: $\mathcal{R}^R_{rot}[T]$

Tree Rotation Proposals

- Now we apply the $\mathcal{R}^L_{cut}$ and $\mathcal{R}^R_{cut}$ operations to both copies of T_s.
- This makes the T^L_s, T^R_s consistent with the ancestral part of the tree that has changed by our rotation operation which introduced a new rule above T_s that was not there before.

Tree Rotation Proposals

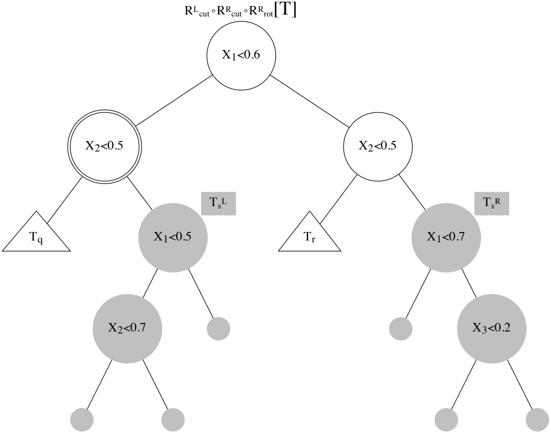


Figure 28: $\mathcal{R}^L_{cut}\mathcal{R}^R_{cut}\mathcal{R}^R_{rot}[T]$

Tree Rotation Proposals

- So far, once we have selected a certain internal node at which to perform a rotation move, the proposal has been deterministic.
- Next are the $\mathcal{R}^L_{merge}, \mathcal{R}^R_{merge}$ operations. These are **not** deterministic.
- We apply the merge operation to the subtrees of η_i and $r(p(\eta_i))$.

Tree Rotation Proposals

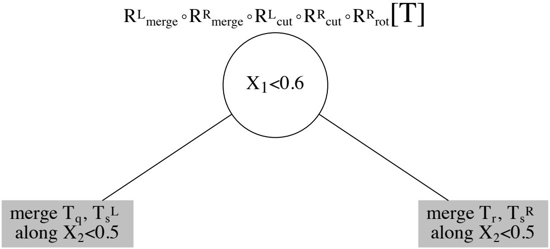

$R^L{}_{merge} \circ R^R{}_{merge} \circ R^L{}_{cut} \circ R^R{}_{cut} \circ R^R{}_{rot}[T]$
X_1<0.6
merge T_q, $T_s{}^L$ along X_2<0.5
merge T_r, $T_s{}^R$ along X_2<0.5

Figure 29: $\mathcal{R}^L_{merge}, \mathcal{R}^R_{merge}\mathcal{R}^L_{cut}\mathcal{R}^R_{cut}\mathcal{R}^R_{rot}[T]$

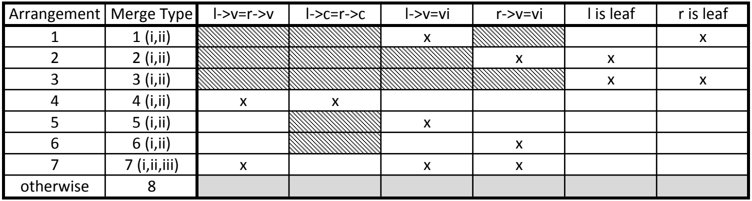
Tree Rotation Proposals

- The merge operation is a recursive procedure that takes two subtrees and fuses them along a v, c merging rule.
- Essentially, it is needed so that one could "reconstruct" T_s from T_s^L, T_s^R if we wanted to invert the rotation operation.
- Why would we want to invert the rotation operation?
  - We need reversibility for our MCMC.

Tree Rotation Proposals

- Merge starts by considering the root nodes of both subtrees T_s^L, T_s^R and attempts to merge them.
- Depending on the configuration of these root nodes, there may be multiple possible merges (select one at random).
- Once the merge has been performed at the root node level, we may be done.
- Or, a subsequent recursive merge is required which would involve comparing two children nodes further down the subtree.
- These complicated operations are performed by matching the configuration of the current subtree nodes being merged to the table on the next page. This table describes various arrangements, each of which has a certain set of possible merged configurations.

Tree Rotation Proposals



Arrangement	Merge Type	l->v=r->v	l->c=r->c	l->v=vi	r->v=vi	l is leaf	r is leaf
1	1 (i,ii)	▨	▨	x	▨		x
2	2 (i,ii)	▨	▨	▨	x	x	
3	3 (i,ii)	▨	▨	▨	▨	x	x
4	4 (i,ii)	x	x				
5	5 (i,ii)		▨	x			
6	6 (i,ii)		▨		x		
7	7 (i,ii,iii)	x		x	x		
otherwise	8						

Figure 30: Merge possibilities

Tree Rotation Proposals

- For example, let's look at the possible merges for arrangements 4 and arrangement 7.

Tree Rotation Proposals

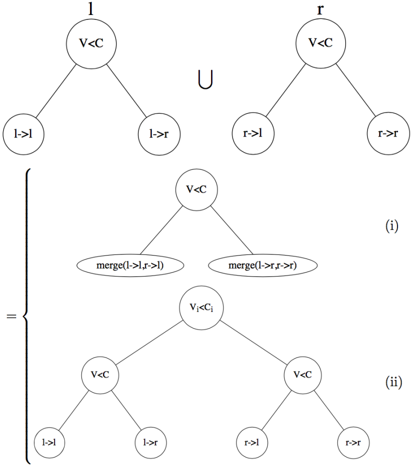


Figure 31: Merge type 4: Both nodes split on the same variable at same cutpoint.

Tree Rotation Proposals

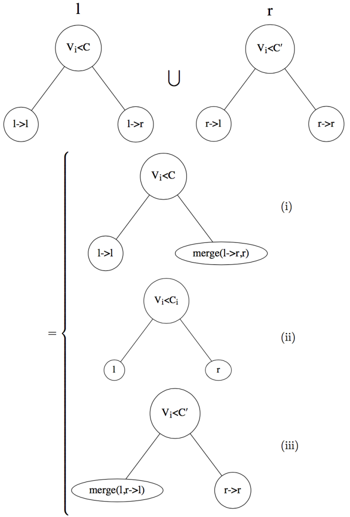


Figure 32: Merge type 7: Both nodes split on the same variable which is also the merging variable.

Tree Rotation Proposals

- We could look at all the possible results of applying the merge operations of our example case.
- But for simplicity, let's suppose the trees don't select any of the fancy merges, that is they stay as in the following diagram.

Tree Rotation Proposals

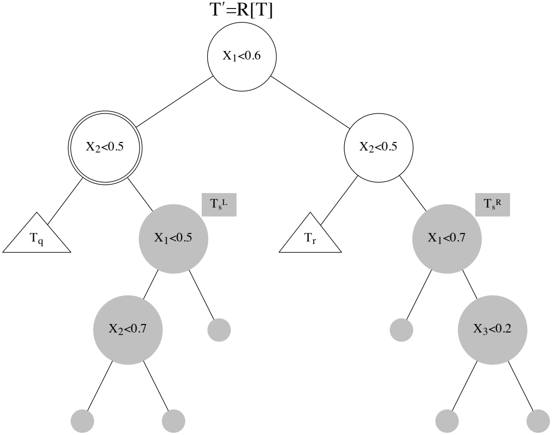


Figure 33: Here is one possible result, T', after applying the merges.

Tree Rotation Proposals

- What happens if we apply the rotate again at node η_2 of the new tree, T'?
- The rule $X_2 < 0.5$ will go back up a level and $X_1 < 0.6$ will now appear below that rule on both left and right sides of the tree.
- A new "T_s" will be copied to both sides as before.
- The (determinisitc) split operations will be performed.
- And we will end up needing to perform two merges along the rule $X_1 < 0.6$.

Tree Rotation Proposals

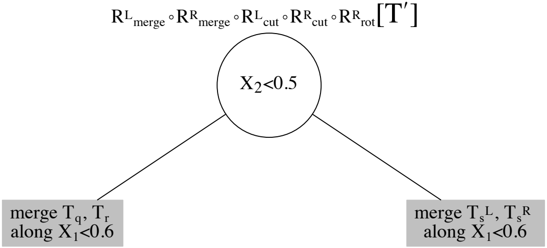

$R^{L}_{merge} \circ R^{R}_{merge} \circ R^{L}_{cut} \circ R^{R}_{cut} \circ R^{R}_{rot}[T']$
$X_2<0.5$
merge T_q, T_r along $X_1<0.6$
merge $T_s{}^L$, $T_s{}^R$ along $X_1<0.6$

Figure 34: Subsequent merges after applying rotate a second time to T'.

Tree Rotation Proposals

- Let's suppose the left side will stay as is after the merge.
- What are all the possible merges on the right side?

Tree Rotation Proposals

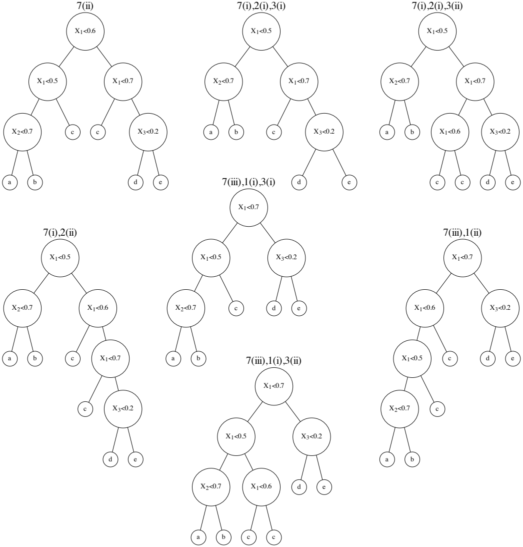


Figure 35: All possible merges of T_s^L, T_s^R after the second rotation.

Tree Rotation Proposals

- We could choose any one of these seven possibilities at random.
- If we choose the merge shown in the middle, created by application of merge types 7(iii), 1(i), 3(i), we arrive back at our original tree.
- We have demonstrated reversibility!
- Merge never destroys information, it just changes the tree-based representation of the same information.

Tree Rotation Proposals

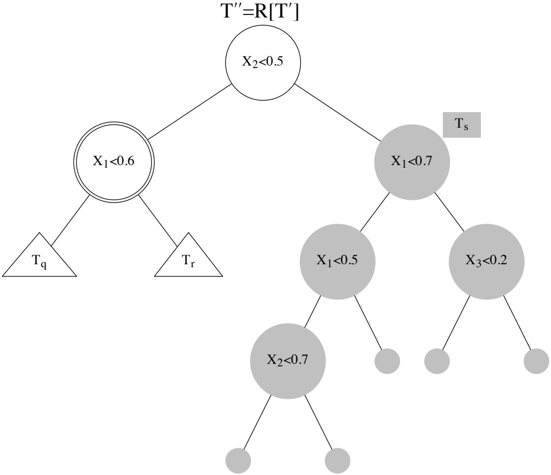


Figure 36: $T'' = T$ after subsequent rotation with non-zero probability.

Tree Rotation Proposals

- Let $L(T)$ be the likelihood of tree T and $\pi(T)$ be the prior probability of T.
- Let $p_r(T)$ be the probability of rotating at a rotatable node in tree T.
- Let n_m^l, n_m^r be the number of possible left/right side merges in the transition $T \to T'$ and let n_s^l, n_s^r be the number of possible left/right side merges in the transition $T' \to T$.
- Let $p_m^l = 1/n_m^l, p_m^r = 1/n_m^r$ and $p_s^l = 1/n_s^l, p_s^r = 1/n_s^r$.
- The acceptance probability for a rotate of $T \to T'$ can be calculated as

$$\alpha = min\left(1, \frac{L(T')\pi(T')p_r(T')p_s^l p_s^r}{L(T)\pi(T)p_r(T)p_m^l p_m^r}\right)$$

Improved BART MCMC Algorithm

1. For $j = 1, \ldots, m$
    1.1 With probability p_{bd} draw τ_j via birth/death; otherwise draw τ_j via rotation.
    1.2 Draw $(v_{j,i}, c_{j,i})|\tau_j, \{(v_{j,-i}, c_{j,-i})\}, \sigma^2, \mathbf{Y}$ for all $i = 1, \ldots, |T_j|$.
    1.3 Draw $\mu_{jk}|\tau_j, \{(v_{j,i}, c_{j,i})\}, \sigma^2, \mathbf{Y}$ for all $k = 1, \ldots, |M_j|$ (Gibbs step).
2. Draw $\sigma^2|\{\tau_j, \{(v_{j,i}, c_{j,i})\}, \{\mu_{jk}\}\}, \mathbf{Y}$ (Gibbs step)

- Note: for (1.2) sometimes just sampling the cutpoints might be reasonable, otherwise an approach that samples variable and cutpoints with probability p_{cv} otherwise just sample cutpoints.