Bayesian Regression Trees

STAT8810, Fall 2017

M.T. Pratola

October 17, 2017

Today

Bayesian Single-Tree Models

Bayesian Regression Trees

- A more explicitly "divide-and-conquer" approach to the theme of localization.

Bayesian Regression Trees

- A more explicitly "divide-and-conquer" approach to the theme of localization.
  - Fit locally simple models to arrive at a more flexible global model.

Bayesian Regression Trees

- A more explicitly “divide-and-conquer” approach to the theme of localization.
  - Fit locally simple models to arrive at a more flexible global model.
  - Local models depend on subset of the data, increasing computational scalability compared to GP regression.

Bayesian Regression Trees

- A more explicitly “divide-and-conquer” approach to the theme of localization.
  - Fit locally simple models to arrive at a more flexible global model.
  - Local models depend on subset of the data, increasing computational scalability compared to GP regression.
- Tradeoff is model no longer interpolates observations.

Bayesian Regression Trees

- A more explicitly "divide-and-conquer" approach to the theme of localization.
  - Fit locally simple models to arrive at a more flexible global model.
  - Local models depend on subset of the data, increasing computational scalability compared to GP regression.
- Tradeoff is model no longer interpolates observations.
  - Fine for data which is observed with obserational error.

Bayesian Regression Trees

- A more explicitly "divide-and-conquer" approach to the theme of localization.
  - Fit locally simple models to arrive at a more flexible global model.
  - Local models depend on subset of the data, increasing computational scalability compared to GP regression.
- Tradeoff is model no longer interpolates observations.
  - Fine for data which is observed with obserational error.
  - Not ideal for deterministic simulator outputs, but we already know approximations of various sorts are needed for this problem.

Bayesian Single Tree Model

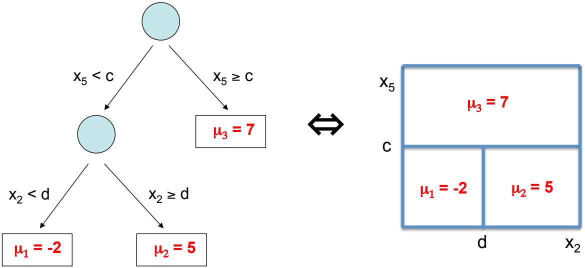


Figure 1: A Single Tree with Scalar Terminal Nodes

Bayesian Single Tree Model

- To take a Bayesian approach, we need to define a stochastic representation of this model.

Bayesian Single Tree Model

- To take a Bayesian approach, we need to define a stochastic representation of this model.
- Let us call $z(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ for $\mathbf{x} \in \mathbb{R}^d$ to be a mapping from the inputs to the (unobserved) response function.

Bayesian Single Tree Model

- To take a Bayesian approach, we need to define a stochastic representation of this model.
- Let us call $z(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ for $\mathbf{x} \in \mathbb{R}^d$ to be a mapping from the inputs to the (unobserved) response function.
- And let us assume that the observed data, $y(\mathbf{x}_i), i = 1, \ldots, n$ is observed with i.i.d. Normally distributed error,

$$y(\mathbf{x}_i) = z(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2).$$

Bayesian Single Tree Model

- Previously, in the GP approach, we would place a GP prior on the $z(\mathbf{x})$ process and write the posterior of the parameters, $\boldsymbol{\rho}$, as

$$\pi(\boldsymbol{\rho}|\mathbf{y}) \propto L(\boldsymbol{\rho}|\mathbf{y})\pi(\boldsymbol{\rho})$$

  and we would predict the response function z using

$$\pi(z(\mathbf{x})|\mathbf{y}) = \int_{\boldsymbol{\rho}} \pi(z(\mathbf{x})|\boldsymbol{\rho}, \mathbf{y})\pi(\boldsymbol{\rho}|\mathbf{y})d\boldsymbol{\rho}.$$

Bayesian Single Tree Model

- Previously, in the GP approach, we would place a GP prior on the $z(\mathbf{x})$ process and write the posterior of the parameters, $\boldsymbol{\rho}$, as

$$\pi(\boldsymbol{\rho}|\mathbf{y}) \propto L(\boldsymbol{\rho}|\mathbf{y})\pi(\boldsymbol{\rho})$$

  and we would predict the response function z using

$$\pi(z(\mathbf{x})|\mathbf{y}) = \int_{\boldsymbol{\rho}} \pi(z(\mathbf{x})|\boldsymbol{\rho}, \mathbf{y})\pi(\boldsymbol{\rho}|\mathbf{y})d\boldsymbol{\rho}.$$

- What is the analogue for a stochastic representation of a tree-process model?

Bayesian Single Tree Model

- Previously, in the GP approach, we would place a GP prior on the $z(\mathbf{x})$ process and write the posterior of the parameters, $\boldsymbol{\rho}$, as

$$\pi(\boldsymbol{\rho}|\mathbf{y}) \propto L(\boldsymbol{\rho}|\mathbf{y})\pi(\boldsymbol{\rho})$$

  and we would predict the response function z using

$$\pi(z(\mathbf{x})|\mathbf{y}) = \int_{\boldsymbol{\rho}} \pi(z(\mathbf{x})|\boldsymbol{\rho}, \mathbf{y})\pi(\boldsymbol{\rho}|\mathbf{y})d\boldsymbol{\rho}.$$

- What is the analogue for a stochastic representation of a tree-process model?
  - We need to identify parameters and specify priors on them.

Bayesian Single Tree Model

- Previously, in the GP approach, we would place a GP prior on the $z(\mathbf{x})$ process and write the posterior of the parameters, $\boldsymbol{\rho}$, as
$$\pi(\boldsymbol{\rho}|\mathbf{y}) \propto L(\boldsymbol{\rho}|\mathbf{y})\pi(\boldsymbol{\rho})$$
and we would predict the response function z using
$$\pi(z(\mathbf{x})|\mathbf{y}) = \int_{\boldsymbol{\rho}} \pi(z(\mathbf{x})|\boldsymbol{\rho}, \mathbf{y})\pi(\boldsymbol{\rho}|\mathbf{y})d\boldsymbol{\rho}.$$
- What is the analogue for a stochastic representation of a tree-process model?
  - We need to identify parameters and specify priors on them.
- For example:

Bayesian Single Tree Model

- Previously, in the GP approach, we would place a GP prior on the $z(\mathbf{x})$ process and write the posterior of the parameters, $\boldsymbol{\rho}$, as
$$\pi(\boldsymbol{\rho}|\mathbf{y}) \propto L(\boldsymbol{\rho}|\mathbf{y})\pi(\boldsymbol{\rho})$$
and we would predict the response function z using
$$\pi(z(\mathbf{x})|\mathbf{y}) = \int_{\boldsymbol{\rho}} \pi(z(\mathbf{x})|\boldsymbol{\rho}, \mathbf{y})\pi(\boldsymbol{\rho}|\mathbf{y})d\boldsymbol{\rho}.$$
- What is the analogue for a stochastic representation of a tree-process model?
    - We need to identify parameters and specify priors on them.
- For example:
    - Internal node parameter variables - what variable is used in a split rule and what value that variable is split at.

Bayesian Single Tree Model

- Previously, in the GP approach, we would place a GP prior on the $z(\mathbf{x})$ process and write the posterior of the parameters, $\boldsymbol{\rho}$, as

$$\pi(\boldsymbol{\rho}|\mathbf{y}) \propto L(\boldsymbol{\rho}|\mathbf{y})\pi(\boldsymbol{\rho})$$

  and we would predict the response function z using

$$\pi(z(\mathbf{x})|\mathbf{y}) = \int_{\boldsymbol{\rho}} \pi(z(\mathbf{x})|\boldsymbol{\rho}, \mathbf{y})\pi(\boldsymbol{\rho}|\mathbf{y})d\boldsymbol{\rho}.$$

- What is the analogue for a stochastic representation of a tree-process model?
  - We need to identify parameters and specify priors on them.
- For example:
  - Internal node parameter variables - what variable is used in a split rule and what value that variable is split at.
  - Some way of defining the connection between nodes to form a stochastic tree.

Bayesian Single Tree Model

- Previously, in the GP approach, we would place a GP prior on the $z(\mathbf{x})$ process and write the posterior of the parameters, $\boldsymbol{\rho}$, as
$$\pi(\boldsymbol{\rho}|\mathbf{y}) \propto L(\boldsymbol{\rho}|\mathbf{y})\pi(\boldsymbol{\rho})$$
and we would predict the response function z using
$$\pi(z(\mathbf{x})|\mathbf{y}) = \int_{\boldsymbol{\rho}} \pi(z(\mathbf{x})|\boldsymbol{\rho},\mathbf{y})\pi(\boldsymbol{\rho}|\mathbf{y})d\boldsymbol{\rho}.$$
- What is the analogue for a stochastic representation of a tree-process model?
  - We need to identify parameters and specify priors on them.
- For example:
  - Internal node parameter variables - what variable is used in a split rule and what value that variable is split at.
  - Some way of definining the connection between nodes to form a stochastic tree.
  - Terminal node parameters for those scalar "μ's".

Bayesian Single Tree Model

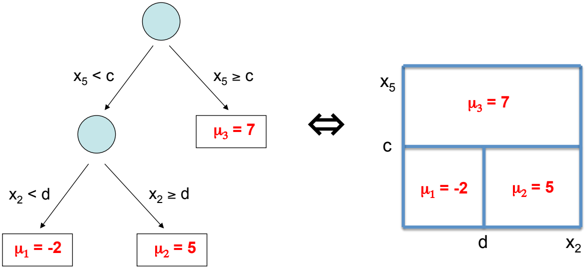
- Previously, in the GP approach, we would place a GP prior on the $z(\mathbf{x})$ process and write the posterior of the parameters, $\boldsymbol{\rho}$, as
$$\pi(\boldsymbol{\rho}|\mathbf{y}) \propto L(\boldsymbol{\rho}|\mathbf{y})\pi(\boldsymbol{\rho})$$
and we would predict the response function z using
$$\pi(z(\mathbf{x})|\mathbf{y}) = \int_{\boldsymbol{\rho}} \pi(z(\mathbf{x})|\boldsymbol{\rho}, \mathbf{y})\pi(\boldsymbol{\rho}|\mathbf{y})d\boldsymbol{\rho}.$$
- What is the analogue for a stochastic representation of a tree-process model?
  - We need to identify parameters and specify priors on them.
- For example:
  - Internal node parameter variables - what variable is used in a split rule and what value that variable is split at.
  - Some way of definining the connection between nodes to form a stochastic tree.
  - Terminal node parameters for those scalar "μ's".
  - Model complexity?

Bayesian Single Tree Model

- So, think $Z(\mathbf{x}) := Z(\mathbf{x}|\mathcal{T}, \mathcal{M})$, where $\mathcal{T}$ are parameters associated with the internal configuration of the tree and $\mathcal{M}$ are parameters associated with the terminal nodes.

Bayesian Single Tree Model

- So, think $Z(\mathbf{x}) := Z(\mathbf{x}|\mathcal{T}, \mathcal{M})$, where $\mathcal{T}$ are parameters associated with the internal configuration of the tree and $\mathcal{M}$ are parameters associated with the terminal nodes.
- A realization of $Z(\mathbf{x}|\mathcal{T}, \mathcal{M})$ is this:



Bayesian Single Tree Model

- Given a $(\mathcal{T}, \mathcal{M})$, we can think of $Z(\mathbf{x})$ as a random function assigning a response value given a particular input, $\mathbf{x}$.

Bayesian Single Tree Model

- Given a $(\mathcal{T}, \mathcal{M})$, we can think of $Z(\mathbf{x})$ as a random function assigning a response value given a particular input, $\mathbf{x}$.
- For instance, in the previous tree, conditional on $\mathcal{T}, \mathcal{M}$ that gave us that picture, an input $\mathbf{x}$ such that $x_5 < c$ and $x_2 > d$ would have predicted response $\hat{y}(\mathbf{x}) \equiv \mu_2 = 5$.

Bayesian Single Tree Model

- Given a $(\mathcal{T}, \mathcal{M})$, we can think of $Z(\mathbf{x})$ as a random function assigning a response value given a particular input, $\mathbf{x}$.
- For instance, in the previous tree, conditional on $\mathcal{T}, \mathcal{M}$ that gave us that picture, an input $\mathbf{x}$ such that $x_5 < c$ and $x_2 > d$ would have predicted response $\hat{y}(\mathbf{x}) \equiv \mu_2 = 5$.
- Our task then is to specify priors on $\mathcal{T}, \mathcal{M}$ and derive an algorithm for sampling the posterior distribution of these parameters given data.

Bayesian Single Tree Model

- Given a $(\mathcal{T}, \mathcal{M})$, we can think of $Z(\mathbf{x})$ as a random function assigning a response value given a particular input, $\mathbf{x}$.
- For instance, in the previous tree, conditional on $\mathcal{T}, \mathcal{M}$ that gave us that picture, an input $\mathbf{x}$ such that $x_5 < c$ and $x_2 > d$ would have predicted response $\hat{y}(\mathbf{x}) \equiv \mu_2 = 5$.
- Our task then is to specify priors on $\mathcal{T}, \mathcal{M}$ and derive an algorithm for sampling the posterior distribution of these parameters given data.
    - Presumably, if our model definition is useful, we will be able to predict our observations fairly well.

Model Variables

- What parameters are associated with the abstract representation $\mathcal{T}$?

† H.A. Chipman, E.I. George and R.E. McCulloch: *Bayesian CART Model Search*, Journal of the American Statistical Association, vol.93, pp.935–948 (1998).

H.A. Chipman, E.I. George and R.E. McCulloch: *BART: Bayesian Additive Regression Trees*, The Annals of Applied Statistics, vol.4, pp.266–298 (2010).

Model Variables

- What parameters are associated with the abstract representation $\mathcal{T}$?
  - Nodes $\eta_1, \eta_2, \ldots$. These nodes are either internal or terminal.

† H.A. Chipman, E.I. George and R.E. McCulloch: *Bayesian CART Model Search*, Journal of the American Statistical Association, vol.93, pp.935–948 (1998).

H.A. Chipman, E.I. George and R.E. McCulloch: *BART: Bayesian Additive Regression Trees*, The Annals of Applied Statistics, vol.4, pp.266–298 (2010).

Model Variables

- What parameters are associated with the abstract representation $\mathcal{T}$?
  - Nodes $\eta_1, \eta_2, \ldots$. These nodes are either internal or terminal.
  - For each internal node η_i, there is an associated tuple v_i, c_i which define the split rule $x_{v_i} < c_i$.

† H.A. Chipman, E.I. George and R.E. McCulloch: *Bayesian CART Model Search*, Journal of the American Statistical Association, vol.93, pp.935–948 (1998).

H.A. Chipman, E.I. George and R.E. McCulloch: *BART: Bayesian Additive Regression Trees*, The Annals of Applied Statistics, vol.4, pp.266–298 (2010).

Model Variables

- What parameters are associated with the abstract representation $\mathcal{T}$?
  - Nodes $\eta_1, \eta_2, \ldots$. These nodes are either internal or terminal.
  - For each internal node η_i, there is an associated tuple v_i, c_i which define the split rule $x_{v_i} < c_i$.
  - For each terminal node η_j, there is an associated scalar parameter μ_j.

† H.A. Chipman, E.I. George and R.E. McCulloch: *Bayesian CART Model Search*, Journal of the American Statistical Association, vol.93, pp.935–948 (1998).

H.A. Chipman, E.I. George and R.E. McCulloch: *BART: Bayesian Additive Regression Trees*, The Annals of Applied Statistics, vol.4, pp.266–298 (2010).

Model Variables

- What parameters are associated with the abstract representation $\mathcal{T}$?
  - Nodes $\eta_1, \eta_2, \ldots$. These nodes are either internal or terminal.
  - For each internal node η_i, there is an associated tuple v_i, c_i which define the split rule $x_{v_i} < c_i$.
  - For each terminal node η_j, there is an associated scalar parameter μ_j.
- There are many ways one might specify a stochastic tree model using these variables. We follow the generative process described in a series of papers by Chipman, George and McCulloch (CGM)†.

† H.A. Chipman, E.I. George and R.E. McCulloch: *Bayesian CART Model Search*, Journal of the American Statistical Association, vol.93, pp.935–948 (1998).

H.A. Chipman, E.I. George and R.E. McCulloch: *BART: Bayesian Additive Regression Trees*, The Annals of Applied Statistics, vol.4, pp.266–298 (2010).

Model Variables

- Note that CGM do not specify edges, say e_{ij} for an edge between η_i and η_j in their model.

Model Variables

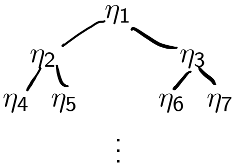
- Note that CGM do not specify edges, say e_{ij} for an edge between η_i and η_j in their model.
- This is because for such binary tree models, the presense of an edge e_{ij} is deterministic given that η_i, η_j are in the tree.

Model Variables

- Note that CGM do not specify edges, say e_{ij} for an edge between η_i and η_j in their model.
- This is because for such binary tree models, the presense of an edge e_{ij} is deterministic given that η_i, η_j are in the tree.
- Another way of saying this is that tree models are not arbitrary graphical models where one might learn both the η_i's and the e_{ij}'s.

Model Variables

- Note that CGM do not specify edges, say e_{ij} for an edge between η_i and η_j in their model.
- This is because for such binary tree models, the presense of an edge e_{ij} is deterministic given that η_i, η_j are in the tree.
- Another way of saying this is that tree models are not arbitrary graphical models where one might learn both the η_i's and the e_{ij}'s.
- For simplicity, a unique numbering system for nodes is employed. η_1 is the root node, and the expansion looks like:



Priors

- Let $\mathcal{I}$ represent the collection of indices of internal nodes η_i, and $\mathcal{B}$ represent the collection of indices of terminal nodes η_i.

† H.A. Chipman, E.I. George and R.E. McCulloch: *BART: Bayesian Additive Regression Trees*, The Annals of Applied Statistics, vol.4, pp.266–298 (2010).

M.T. Pratola: *Efficient Metropolis–Hastings Proposal Mechanisms for Bayesian*

Priors

- Let $\mathcal{I}$ represent the collection of indices of internal nodes η_i, and $\mathcal{B}$ represent the collection of indices of terminal nodes η_i.
- The CGM prior† is as follows:

$$
\begin{aligned}
\pi(\sigma^2, \mathcal{T}, \mathcal{M}) &= \pi(\sigma^2)\pi(\mathcal{M}|\mathcal{T})\pi(\mathcal{T}) \\
&= \pi(\sigma^2) \prod_{j \in \mathcal{B}} \pi(\mu_j|\eta_j)\pi(\eta_j \text{ is terminal}) \\
&\quad \times \prod_{k \in \mathcal{I}} \pi(v_k, c_k|\mathcal{T} \setminus \eta_k)\pi(\eta_k \text{ is internal}) \\
&= \pi(\sigma^2) \prod_{j \in \mathcal{B}} \pi(\mu_j|\eta_j)\pi(\eta_j \text{ is terminal}) \\
&\quad \times \prod_{k \in \mathcal{I}} \pi(c_k|v_k, \mathcal{T} \setminus \eta_k)\pi(v_k|\mathcal{T} \setminus \eta_k)\pi(\eta_k \text{ is interna}
\end{aligned}
$$

† H.A. Chipman, E.I. George and R.E. McCulloch: *BART: Bayesian Additive Regression Trees*, The Annals of Applied Statistics, vol.4, pp.266–298 (2010).

M.T. Pratola: *Efficient Metropolis–Hastings Proposal Mechanisms for Bayesian*

Priors

- The prior on a node being internal/terminal is given by the so-called depth penalizing prior,

$$\pi(\eta_j \text{ is internal}) = \alpha(1 + d(\eta_j, \eta_1))^{-\beta}$$

where $d(\eta_j, \eta_1)$ is the depth of node η_j, $\alpha \in (0, 1)$ and $\beta \in [0, \infty)$, and correspondingly,

$$\pi(\eta_j \text{ is terminal}) = 1 - \pi(\eta_j \text{ is internal}).$$

Priors

- The prior on a node being internal/terminal is given by the so-called depth penalizing prior,

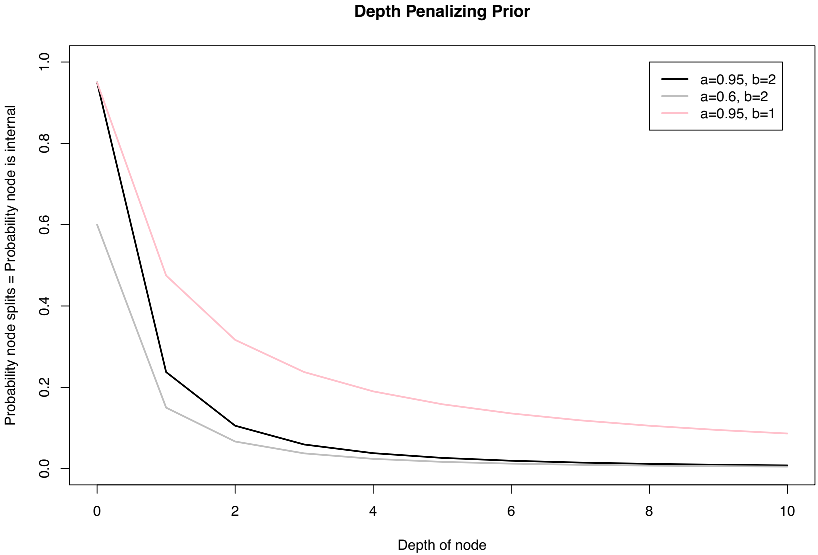
$$\pi(\eta_j \text{ is internal}) = \alpha(1 + d(\eta_j, \eta_1))^{-\beta}$$

  where $d(\eta_j, \eta_1)$ is the depth of node η_j, $\alpha \in (0, 1)$ and $\beta \in [0, \infty)$, and correspondingly,

$$\pi(\eta_j \text{ is terminal}) = 1 - \pi(\eta_j \text{ is internal}).$$

- Interpretation is probability a node splits (and is hence internal) decreases the deeper that node is in the tree. In other words, this prior favors shallower, sparser trees.

Priors


Depth Penalizing Prior
a=0.95, b=2
a=0.6, b=2
a=0.95, b=1
Probability node splits = Probability node is internal
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
Depth of node

Priors

- The prior on cutpoints c_i is typically a discrete uniform distribution over the cutpoints

$$\left\{0, \frac{1}{n_v - 1}, \ldots, \frac{n_v - 2}{n_v - 1}, 1\right\}$$

where n_v is a fixed, user-specified discretization resolution for variable v.

Priors

- The prior on cutpoints c_i is typically a discrete uniform distribution over the cutpoints

$$\left\{0, \frac{1}{n_v - 1}, \ldots, \frac{n_v - 2}{n_v - 1}, 1\right\}$$

  where n_v is a fixed, user-specified discretization resolution for variable v.
- The prior on variables v_i is typically a discrete uniform distribution over the variable indices

$$\{1, 2, \ldots, d\}.$$

Priors

- The prior on the terminal node scalar parameters are i.i.d. conjugate normal,

$$\mu_j | \mathcal{T} \sim N(\mu_\mu, \sigma_\mu^2) \text{ for all } j \in \mathcal{B}$$

Priors

- The prior on the terminal node scalar parameters are i.i.d. conjugate normal,

$$\mu_j | \mathcal{T} \sim N(\mu_\mu, \sigma_\mu^2) \text{ for all } j \in \mathcal{B}$$

  - typically, we operate on mean-centered data and hence the prior will have assumed mean $\mu_\mu = 0$.

Priors

- The prior on the terminal node scalar parameters are i.i.d. conjugate normal,

$$\mu_j | \mathcal{T} \sim N(\mu_\mu, \sigma_\mu^2) \text{ for all } j \in \mathcal{B}$$

  - typically, we operate on mean-centered data and hence the prior will have assumed mean $\mu_\mu = 0$.
- The prior on the variance is conjugate scaled-inverse-chisquared,

$$\sigma^2 \sim \chi^{-2}_{\nu,\tau^2}$$

Priors

- The prior on the terminal node scalar parameters are i.i.d. conjugate normal,

$$\mu_j | \mathcal{T} \sim N(\mu_\mu, \sigma^2_\mu) \text{ for all } j \in \mathcal{B}$$

  - typically, we operate on mean-centered data and hence the prior will have assumed mean $\mu_\mu = 0$.
- The prior on the variance is conjugate scaled-inverse-chisquared,

$$\sigma^2 \sim \chi^{-2}_{\nu,\tau^2}$$

  - this is a different, but still conjugate, prior than what we had used in our GP model (where we used precision $\lambda \sim$ Gamma).

Unconditional Realizations

- We could draw unconditional realizations of our stochastic regression tree process:

† Note that the variables and cutpoints available at non-root nodes may (very likely) depend on the ancestral part of the tree.

Unconditional Realizations

- We could draw unconditional realizations of our stochastic regression tree process:

1. Calculate prior probability the root node splits

† Note that the variables and cutpoints available at non-root nodes may (very likely) depend on the ancestral part of the tree.

Unconditional Realizations

- We could draw unconditional realizations of our stochastic regression tree process:

1. Calculate prior probability the root node splits
   - If root node is terminal, draw μ_1 from Normal prior.

† Note that the variables and cutpoints available at non-root nodes may (very likely) depend on the ancestral part of the tree.

Unconditional Realizations

- We could draw unconditional realizations of our stochastic regression tree process:

1. Calculate prior probability the root node splits
   - If root node is terminal, draw μ_1 from Normal prior.
   - If root node is internal, draw v_1 and c_1 from Uniform priors.

† Note that the variables and cutpoints available at non-root nodes may (very likely) depend on the ancestral part of the tree.

Unconditional Realizations

- We could draw unconditional realizations of our stochastic regression tree process:

1. Calculate prior probability the root node splits
   - If root node is terminal, draw μ_1 from Normal prior.
   - If root node is internal, draw v_1 and c_1 from Uniform priors.
2. Calculate prior probability node 2 splits

† Note that the variables and cutpoints available at non-root nodes may (very likely) depend on the ancestral part of the tree.

Unconditional Realizations

- We could draw unconditional realizations of our stochastic regression tree process:

1. Calculate prior probability the root node splits
   - If root node is terminal, draw μ_1 from Normal prior.
   - If root node is internal, draw v_1 and c_1 from Uniform priors.
2. Calculate prior probability node 2 splits
   - If node 2 is terminal, draw μ_2 from Normal prior.

† Note that the variables and cutpoints available at non-root nodes may (very likely) depend on the ancestral part of the tree.

Unconditional Realizations

- We could draw unconditional realizations of our stochastic regression tree process:

1. Calculate prior probability the root node splits
   - If root node is terminal, draw μ_1 from Normal prior.
   - If root node is internal, draw v_1 and c_1 from Uniform priors.
2. Calculate prior probability node 2 splits
   - If node 2 is terminal, draw μ_2 from Normal prior.
   - If node 2 is internal, draw v_2 and c_2 from Uniform priors†.

† Note that the variables and cutpoints available at non-root nodes may (very likely) depend on the ancestral part of the tree.

Unconditional Realizations

- We could draw unconditional realizations of our stochastic regression tree process:

1. Calculate prior probability the root node splits
   - If root node is terminal, draw μ_1 from Normal prior.
   - If root node is internal, draw v_1 and c_1 from Uniform priors.
2. Calculate prior probability node 2 splits
   - If node 2 is terminal, draw μ_2 from Normal prior.
   - If node 2 is internal, draw v_2 and c_2 from Uniform priors†.
3. etc.

† Note that the variables and cutpoints available at non-root nodes may (very likely) depend on the ancestral part of the tree.

Example: Unconditional Realization

```
set.seed(88)
cuts=seq(0.1,0.9,length=9)
nonterms=c()
terms=c()
stop=FALSE
alpha=0.95
beta=2

# Node 1
d=0
psplit=alpha*(1+d)^(-beta)
runif(1)<psplit
```

```
## [1] TRUE
```

```
nonterms=c(1)
```

Example: Unconditional Realization

```
# Nodes 2,3
d=1
# Node 2
psplit=alpha*(1+d)^(-beta)
runif(1)<psplit
```

```
## [1] TRUE
```

```
nonterms=c(nonterms,2)
# Node 3
psplit=alpha*(1+d)^(-beta)
runif(1)<psplit
```

```
## [1] FALSE
```

```
terms=c(3)
```

Example: Unconditional Realization

```
# Nodes 4,5
d=2
# Node 4
psplit=alpha*(1+d)^(-beta)
runif(1)<psplit
```

```
## [1] FALSE
```

```
terms=c(terms,4)
# Node 5
psplit=alpha*(1+d)^(-beta)
runif(1)<psplit
```

```
## [1] FALSE
```

```
terms=c(terms,5)
# Nowhere left to grow.
```

Example: Unconditional Realization

```
# Now select variable, cutpoints for internal nodes
# Since we have only 1 variable, its always used in splits
variables=rep(0,length(nonterms))

# Now get cuts
cutpoints=rep(0,length(nonterms))
cutpoints[1]=sample(cuts,1)
cutpoints[1]
```

```
## [1] 0.9
```

```
# Now get cut for node 2
cuts=cuts[cuts<cutpoints[1]]
cutpoints[2]=sample(cuts,1)
cutpoints[2]
```

```
## [1] 0.1
```

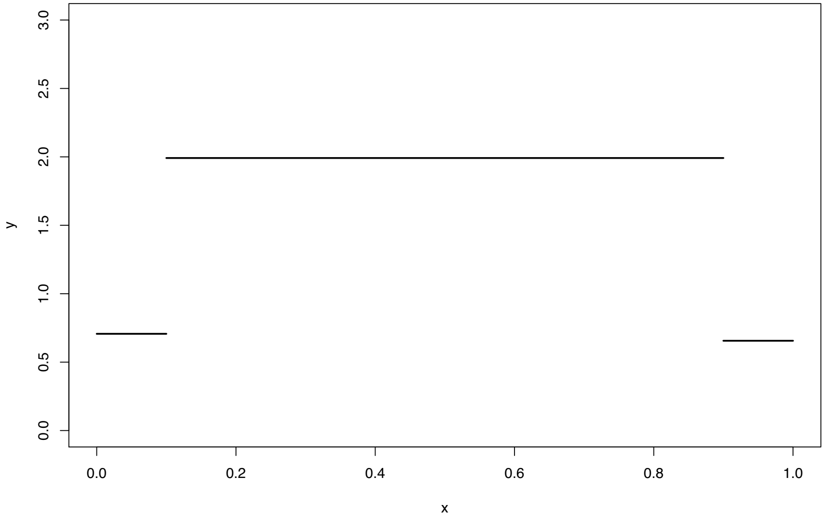
Example: Unconditional Realization

```r
# Now draw terminal node parameters from N(0,tau^2)
tau2=1
mu=rep(0,length(terms))
for(i in 1:length(terms))
  mu[i]=rnorm(1,mean=0,sd=sqrt(tau2))
```

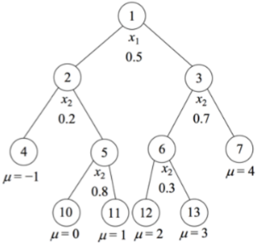
Example: Unconditional Realization

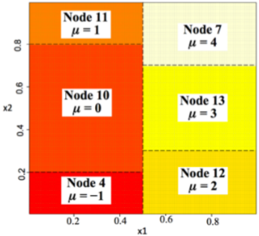
```
# Now plot the function represented by our tree
plot(c(0,cutpoints[2]),rep(mu[1],2),type='l',
    lwd=2,xlim=c(0,1),ylim=c(0,3),xlab="x",ylab="y")
lines(c(cutpoints[2],cutpoints[1]),rep(mu[2],2),lwd=2)
lines(c(cutpoints[1],1),rep(mu[3],2),lwd=2)
```

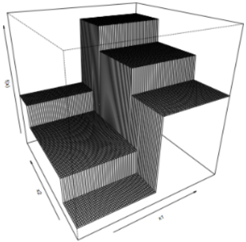
Example: Unconditional Realization



Example Realization with 2 predictors†



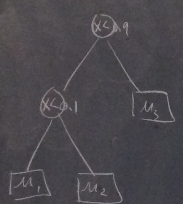




Three different views of a bivariate single tree.

† Source: E.I. George, BNPSKi (2014).

X,



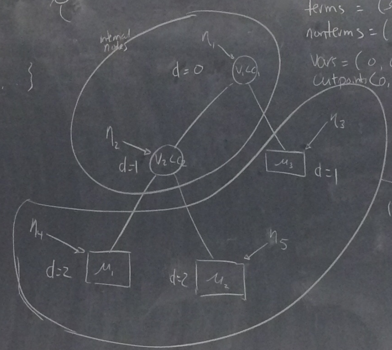
$\mu_j \sim N(0,1)$

- nodes η_j's
- terminal node params $M = \{\mu_1, \mu_2, \ldots\}$
- variable v, cutpoint c for internal node decision rule.
- $d(\eta_j, \eta_1) \equiv$ depth of node η_j
- $\pi(\eta_j \text{ splits}) \propto \alpha(1+d)^{-\beta}$ (internal node)

$\alpha \in (0,1)$

$\beta > 1$.

$\mathcal{T} =$



terms = (3, 4, 5)

nonterms = (1, 2)

vars = (0, 0)

cutpoints (0.9, 0.1)

cuts = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8)

$I = \{1, 2\}$

$B = \{3, 4, 5\}$

$X_1, \ldots, X_p$

$V_1 < C_1$

$X_5 < C_1$

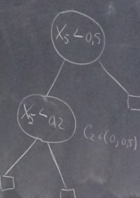
$\pi(c|v)\,\pi(v)$

Model

$y(x_i) = z(x_i) + \varepsilon_i$

$\varepsilon_i \sim N(0, \sigma^2)$

$z(x_i) = z(x_i; \mathcal{T}, M)$



$X_5 \in (0,1)$

$c_2 \in (0, 0.5)$

Sampling the Posterior Distribution

- Recall, our observation model was

$$y(x_i) = z(x_i) + \epsilon_i$$

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

Sampling the Posterior Distribution

- Recall, our observation model was

$$y(x_i) = z(x_i) + \epsilon_i$$

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

- Given observations $\mathbf{y} = (y_1, \ldots, y_n)$, we are interested in sampling the posterior distribution

$$\pi(\sigma^2, \mathcal{T}, \mathcal{M}|\mathbf{y}) \propto L(\sigma^2, \mathcal{T}, \mathcal{M}|\mathbf{y})\pi(\sigma^2)\pi(\mathcal{M}|\mathcal{T})\pi(\mathcal{T})$$

Sampling the Posterior Distribution

- Recall, our observation model was

$$y(x_i) = z(x_i) + \epsilon_i$$

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

- Given observations $\mathbf{y} = (y_1, \ldots, y_n)$, we are interested in sampling the posterior distribution

$$\pi(\sigma^2, \mathcal{T}, \mathcal{M}|\mathbf{y}) \propto L(\sigma^2, \mathcal{T}, \mathcal{M}|\mathbf{y})\pi(\sigma^2)\pi(\mathcal{M}|\mathcal{T})\pi(\mathcal{T})$$

- Conditional on a realization of our stochastic tree process, our likelihood function is

$$L(\sigma^2, \mathcal{T}, \mathcal{M}|\mathbf{y}) = \frac{1}{\sqrt{2\pi}\sigma^n} exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - z(\mathbf{x}_i))^2\right)$$

Sampling the Posterior Distribution

- Our MCMC algorithm will perform the following steps:

Sampling the Posterior Distribution

- Our MCMC algorithm will perform the following steps:

1. Draw $\mathcal{T}|\sigma^2, \mathbf{y}$

Sampling the Posterior Distribution

- Our MCMC algorithm will perform the following steps:

1. Draw $\mathcal{T}|\sigma^2, \mathbf{y}$
    - Metropolis-Hastings step via proposal distribution

Sampling the Posterior Distribution

- Our MCMC algorithm will perform the following steps:

1. Draw $\mathcal{T}|\sigma^2, \mathbf{y}$
    - Metropolis-Hastings step via proposal distribution
2. Draw $\mathcal{M}|\mathcal{T}, \sigma^2, \mathbf{y}$

Sampling the Posterior Distribution

- Our MCMC algorithm will perform the following steps:

1. Draw $\mathcal{T}|\sigma^2, \mathbf{y}$
   - Metropolis-Hastings step via proposal distribution
2. Draw $\mathcal{M}|\mathcal{T}, \sigma^2, \mathbf{y}$
   - Gibbs step using conjugate prior

Sampling the Posterior Distribution

- Our MCMC algorithm will perform the following steps:

1. Draw $\mathcal{T}|\sigma^2, \mathbf{y}$
   - Metropolis-Hastings step via proposal distribution
2. Draw $\mathcal{M}|\mathcal{T}, \sigma^2, \mathbf{y}$
   - Gibbs step using conjugate prior
3. Draw $\sigma^2|\mathcal{T}, \mathcal{M}, \mathbf{y}$

Sampling the Posterior Distribution

- Our MCMC algorithm will perform the following steps:

1. Draw $\mathcal{T}|\sigma^2, \mathbf{y}$
   - Metropolis-Hastings step via proposal distribution
2. Draw $\mathcal{M}|\mathcal{T}, \sigma^2, \mathbf{y}$
   - Gibbs step using conjugate prior
3. Draw $\sigma^2|\mathcal{T}, \mathcal{M}, \mathbf{y}$
   - Gibbs step using conjugate prior

Sampling the Posterior Distribution

- Our MCMC algorithm will perform the following steps:

1. Draw $\mathcal{T}|\sigma^2, \mathbf{y}$
   - Metropolis-Hastings step via proposal distribution
2. Draw $\mathcal{M}|\mathcal{T}, \sigma^2, \mathbf{y}$
   - Gibbs step using conjugate prior
3. Draw $\sigma^2|\mathcal{T}, \mathcal{M}, \mathbf{y}$
   - Gibbs step using conjugate prior

- We'll go in reverse order...

Draw $\sigma^2|\mathcal{T}, \mathcal{M}, \mathbf{y}$

- We have

$$\pi(\sigma^2|\nu,\tau^2) = \frac{\left(\frac{\nu\tau^2}{2}\right)^{\nu/2}}{\Gamma\left(\frac{\nu}{2}\right)\sigma^{\nu+2}} exp\left(-\frac{\nu\tau^2}{2\sigma^2}\right) \propto \frac{1}{\sigma^{\nu+2}} exp\left(-\frac{\nu\tau^2}{2\sigma^2}\right)$$

Draw $\sigma^2|\mathcal{T},\mathcal{M},\mathbf{y}$

- We have

$$\pi(\sigma^2|\nu,\tau^2) = \frac{\left(\frac{\nu\tau^2}{2}\right)^{\nu/2}}{\Gamma\left(\frac{\nu}{2}\right)\sigma^{\nu+2}} exp\left(-\frac{\nu\tau^2}{2\sigma^2}\right) \propto \frac{1}{\sigma^{\nu+2}} exp\left(-\frac{\nu\tau^2}{2\sigma^2}\right)$$

- So,

$$\begin{aligned}
\pi(\sigma^2|\mathcal{T},\mathcal{M},\mathbf{y}) &\propto \frac{1}{\sigma^n} exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - z(\mathbf{x}_i))^2\right) \\
&\quad \times \frac{1}{\sigma^{\nu+2}} exp\left(-\frac{\nu\tau^2}{2\sigma^2}\right) \\
&= \frac{1}{\sigma^{(\nu+n)+2}} exp\left(-\frac{(\nu+n)}{2\sigma^2}\left(\frac{\nu\tau^2+ns^2}{\nu+n}\right)\right)
\end{aligned}$$

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

Draw $\sigma^2|\mathcal{T}, \mathcal{M}, \mathbf{y}$

- And we recognize $\frac{1}{\sigma^{(\nu+n)+2}} exp\left(-\frac{(\nu+n)}{2\sigma^2}\left(\frac{\nu\tau^2+ns^2}{\nu+n}\right)\right)$ as the kernel of a scaled-inverse-chisquared distribution, so

$$\sigma^2|\mathcal{T}, \mathcal{M}, \mathbf{y} \sim \chi^{-2}\left(\nu+n, \frac{\nu\tau^2+ns^2}{\nu+n}\right)$$

Draw $\sigma^2|\mathcal{T}, \mathcal{M}, \mathbf{y}$

- And we recognize $\frac{1}{\sigma^{(\nu+n)+2}} exp\left(-\frac{(\nu+n)}{2\sigma^2}\left(\frac{\nu\tau^2+ns^2}{\nu+n}\right)\right)$ as the kernel of a scaled-inverse-chisquared distribution, so

$$\sigma^2|\mathcal{T}, \mathcal{M}, \mathbf{y} \sim \chi^{-2}\left(\nu+n, \frac{\nu\tau^2+ns^2}{\nu+n}\right)$$

- So we know how to perform the Gibbs step for σ^2.

Draw $\mathcal{M}|\mathcal{T}, \sigma^2, \mathbf{y}$

- What about the terminal node scalar mean parameters?

Draw $\mathcal{M}|\mathcal{T}, \sigma^2, \mathbf{y}$

- What about the terminal node scalar mean parameters?
- Suppose there are B terminal nodes in tree $\mathcal{T}, \eta_1^b, \ldots, \eta_B^b$. It is important to note the following factorization of the likelihood:

$$
\begin{aligned}
L(\sigma^2, \mathcal{T}, \mathcal{M}|\mathbf{y}) &\propto exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - z(\mathbf{x}_i))^2\right) \\
&= exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{B}\sum_{i:y_i\in\eta_j^b}^{n_j}(y_i - \mu_j)^2\right) \\
&= \prod_{j=1}^{B} exp\left(-\frac{1}{2\sigma^2}\sum_{i:y_i\in\eta_j^b}^{n_j}(y_i - \mu_j)^2\right)
\end{aligned}
$$

where n_j is the number of observations mapping to terminal nodes η_j^b and $\sum_j n_j = n$.

Draw $\mathcal{M}|\mathcal{T}, \sigma^2, \mathbf{y}$

- In other words, conditional on $\mathcal{T}$, the scalar terminal node parameters are independent!

Draw $\mathcal{M}|\mathcal{T}, \sigma^2, \mathbf{y}$

- In other words, conditional on $\mathcal{T}$, the scalar terminal node parameters are independent!
- So, we can simply write down the full conditional for each μ_j and draw them sequentially using Gibbs steps.

Draw $\mu_j|\mathcal{T}, \sigma^2, \mathbf{y}$

- Assuming mean-centered observations, our prior is

$$\pi(\mu_j|\mathcal{T}) = N(0, \sigma_\mu^2).$$

Draw $\mu_j|\mathcal{T}, \sigma^2, \mathbf{y}$

- Assuming mean-centered observations, our prior is

$$\pi(\mu_j|\mathcal{T}) = N(0, \sigma_\mu^2).$$

- Based on our results from awhile ago (slides 9), the full conditional is

$$\pi(\mu_j|\sigma^2, \mathcal{T}, \mathbf{y}) \sim N\left(\left(\frac{n_j}{\sigma^2} + \frac{1}{\sigma_\mu^2}\right)^{-1}\left(\frac{n_j \bar{y}_j}{\sigma^2}\right), \left(\frac{n_j}{\sigma^2} + \frac{1}{\sigma_\mu^2}\right)^{-1}\right)$$

where $\bar{y}_j = \frac{1}{n_j}\sum_{i:y_i \in \eta_j^b} y_i$.

Draw $\mathcal{T}|\sigma^2, \mathbf{y}$

- Sampling the posterior distributions of trees is more complicated.

† H.A. Chipman, E.I. George and R.E. McCulloch: *Bayesian CART Model Search*, Journal of the American Statistical Association, vol.93, pp.935–948 (1998).

Draw $\mathcal{T}|\sigma^2, \mathbf{y}$

- Sampling the posterior distributions of trees is more complicated.
  - discrete, infinite-dimensional space

† H.A. Chipman, E.I. George and R.E. McCulloch: *Bayesian CART Model Search*, Journal of the American Statistical Association, vol.93, pp.935–948 (1998).

Draw $\mathcal{T}|\sigma^2, \mathbf{y}$

- Sampling the posterior distributions of trees is more complicated.
  - discrete, infinite-dimensional space
  - need clever(?) Metropolis-Hastings proposals

† H.A. Chipman, E.I. George and R.E. McCulloch: *Bayesian CART Model Search*, Journal of the American Statistical Association, vol.93, pp.935–948 (1998).

Draw $\mathcal{T}|\sigma^2, \mathbf{y}$

- Sampling the posterior distributions of trees is more complicated.
  - discrete, infinite-dimensional space
  - need clever(?) Metropolis-Hastings proposals
  - if $q(\mathcal{T} \to \mathcal{T}')$ changes the number of terminal nodes in the tree, what happens to the terminal node parameters, $\mathcal{M}$?

† H.A. Chipman, E.I. George and R.E. McCulloch: *Bayesian CART Model Search*, Journal of the American Statistical Association, vol.93, pp.935–948 (1998).

Draw $\mathcal{T}|\sigma^2, \mathbf{y}$

- Sampling the posterior distributions of trees is more complicated.
  - discrete, infinite-dimensional space
  - need clever(?) Metropolis-Hastings proposals
  - if $q(\mathcal{T} \to \mathcal{T}')$ changes the number of terminal nodes in the tree, what happens to the terminal node parameters, $\mathcal{M}$?
- Chipman et al.† propose four basic proposals for mixing over tree-space: Birth, Death, Change and Swap.

† H.A. Chipman, E.I. George and R.E. McCulloch: *Bayesian CART Model Search*, Journal of the American Statistical Association, vol.93, pp.935–948 (1998).

Draw $\mathcal{T}|\sigma^2, \mathbf{y}$

- Sampling the posterior distributions of trees is more complicated.
  - discrete, infinite-dimensional space
  - need clever(?) Metropolis-Hastings proposals
  - if $q(\mathcal{T} \to \mathcal{T}')$ changes the number of terminal nodes in the tree, what happens to the terminal node parameters, $\mathcal{M}$?
- Chipman et al.† propose four basic proposals for mixing over tree-space: Birth, Death, Change and Swap.
  - We'll look at Birth and Death only for now.

† H.A. Chipman, E.I. George and R.E. McCulloch: *Bayesian CART Model Search*, Journal of the American Statistical Association, vol.93, pp.935–948 (1998).

Draw $\mathcal{T}|\sigma^2, \mathbf{y}$

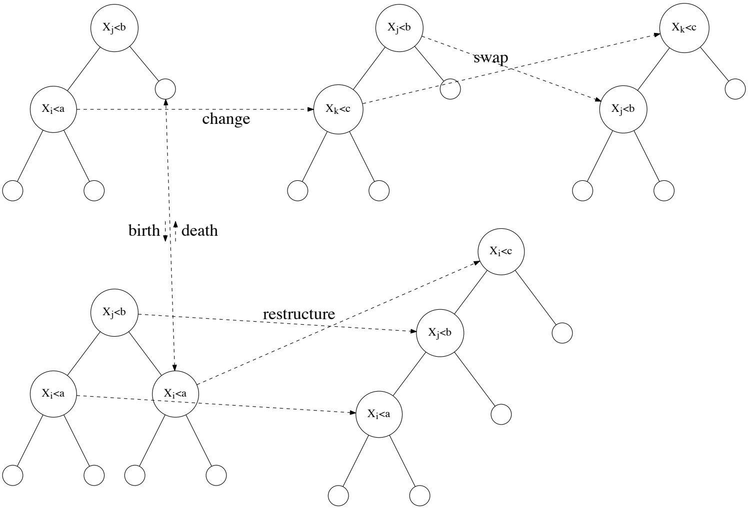


Figure 2: Tree Moves

Draw $\mathcal{T} | \sigma^2, \mathbf{y}$

- A birth replaces an existing terminal node with a new decision rule of the form "$v < c$" and introduces (births) two new terminal nodes below the new decision rule.

Draw $\mathcal{T}|\sigma^2, \mathbf{y}$

- A birth replaces an existing terminal node with a new decision rule of the form "$v < c$" and introduces (births) two new terminal nodes below the new decision rule.
- Conversely, a death selects an existing next-to-terminal node and removes (deaths) its two children terminal nodes, and the selected node loses it's decision rule and instead becomes a terminal node with a new parameter μ.

Draw $\mathcal{T}|\sigma^2, \mathbf{y}$

- A birth replaces an existing terminal node with a new decision rule of the form "$v < c$" and introduces (births) two new terminal nodes below the new decision rule.
- Conversely, a death selects an existing next-to-terminal node and removes (deaths) its two children terminal nodes, and the selected node loses it's decision rule and instead becomes a terminal node with a new parameter μ.
- Note that if we transition from $\mathcal{T} \to \mathcal{T}'$ via birth, then $B' = B + 1$ where $B = |\mathcal{M}|$.

Draw $\mathcal{T}|\sigma^2, \mathbf{y}$

- A birth replaces an existing terminal node with a new decision rule of the form "$v < c$" and introduces (births) two new terminal nodes below the new decision rule.
- Conversely, a death selects an existing next-to-terminal node and removes (deaths) its two children terminal nodes, and the selected node loses it's decision rule and instead becomes a terminal node with a new parameter μ.
- Note that if we transition from $\mathcal{T} \to \mathcal{T}'$ via birth, then $B' = B + 1$ where $B = |\mathcal{M}|$.
- This means that when we birth, a previous terminal node parameter μ disappears and two new parameters, say $\mu_{(l)}$ and $\mu_{(r)}$ are born.

Draw $\mathcal{T}|\sigma^2, \mathbf{y}$

- A birth replaces an existing terminal node with a new decision rule of the form "$v < c$" and introduces (births) two new terminal nodes below the new decision rule.
- Conversely, a death selects an existing next-to-terminal node and removes (deaths) its two children terminal nodes, and the selected node loses it's decision rule and instead becomes a terminal node with a new parameter μ.
- Note that if we transition from $\mathcal{T} \to \mathcal{T}'$ via birth, then $B' = B + 1$ where $B = |\mathcal{M}|$.
- This means that when we birth, a previous terminal node parameter μ disappears and two new parameters, say $\mu_{(l)}$ and $\mu_{(r)}$ are born.
- And when we death, two previous terminal node parameters, $\mu_{(l)}$ and $\mu_{(r)}$, dissappear and a new parameter μ is born.

Draw $\mathcal{T}|\sigma^2, \mathbf{y}$

- The dimension-changing nature of tree proposals (at least birth-death proposals) would appear to introduce some challenges.

† P.J. Green: *Reversible jump Markov chain Monte Carlo computation and Bayesian model determination*, Biometrika, vol.82, pp.711–732 (1995).

Draw $\mathcal{T}|\sigma^2, \mathbf{y}$

- The dimension-changing nature of tree proposals (at least birth-death proposals) would appear to introduce some challenges.
- How to construct such dimension-changing proposals is explored in the Reversible-Jump Markov Chain Monte Carlo (RJMCMC) work of Green (1995)†.

† P.J. Green: *Reversible jump Markov chain Monte Carlo computation and Bayesian model determination*, Biometrika, vol.82, pp.711–732 (1995).

Draw $\mathcal{T}|\sigma^2, \mathbf{y}$

- The dimension-changing nature of tree proposals (at least birth-death proposals) would appear to introduce some challenges.
- How to construct such dimension-changing proposals is explored in the Reversible-Jump Markov Chain Monte Carlo (RJMCMC) work of Green (1995)†.
- Fortunately, Green (1995) shows that when the dimension-changing parameter can be marginalized out, one can proceed with the usual MH algorithm but using the marginalized likelihood.

† P.J. Green: *Reversible jump Markov chain Monte Carlo computation and Bayesian model determination*, Biometrika, vol.82, pp.711–732 (1995).

Draw $\mathcal{T}|\sigma^2, \mathbf{y}$

- The dimension-changing nature of tree proposals (at least birth-death proposals) would appear to introduce some challenges.
- How to construct such dimension-changing proposals is explored in the Reversible-Jump Markov Chain Monte Carlo (RJMCMC) work of Green (1995)†.
- Fortunately, Green (1995) shows that when the dimension-changing parameter can be marginalized out, one can proceed with the usual MH algorithm but using the marginalized likelihood.
- For our conjugate Normal prior on the μ's, this marginal likelihood is available.

† P.J. Green: *Reversible jump Markov chain Monte Carlo computation and Bayesian model determination*, Biometrika, vol.82, pp.711–732 (1995).

Marginal Likelihood

- Marginalizing the portion of the likelihood associated with terminal node η_j^b, we have

$$L(\eta_j^b|\sigma^2, \mathbf{y}) = \int_{\mu_j} L(\eta_j^b|\mu_j, \sigma^2, \mathbf{y})\pi(\mu_j)d\mu_j$$

(I will leave this as an excercise).

Birth Proposal

- We randomly generate $\mathcal{T}'$ as follows:

Birth Proposal

- We randomly generate $\mathcal{T}'$ as follows:

1. Randomly select a terminal node $b \in \{1, \ldots, B\}$ with probability $\frac{1}{B}$ where $B = |\mathcal{M}|$.

Birth Proposal

- We randomly generate $\mathcal{T}'$ as follows:

1. Randomly select a terminal node $b \in \{1, \ldots, B\}$ with probability $\frac{1}{B}$ where $B = |\mathcal{M}|$.
2. Introduce a new rule $v_b \sim \pi_v(v_b)$ and cutpoint $c_b \sim \pi_c(c_b)$ where π_v, π_c are typically discrete Uniform on the available variable, cutpoints.

Birth Proposal

- We randomly generate $\mathcal{T}'$ as follows:

1. Randomly select a terminal node $b \in \{1, \ldots, B\}$ with probability $\frac{1}{B}$ where $B = |\mathcal{M}|$.
2. Introduce a new rule $v_b \sim \pi_v(v_b)$ and cutpoint $c_b \sim \pi_c(c_b)$ where π_v, π_c are typically discrete Uniform on the available variable, cutpoints.
3. Calculate

$$\alpha = min\left\{1, \frac{\pi(\mathcal{T}'|\sigma^2, \mathbf{y})q(\mathcal{T}|\mathcal{T}')}{\pi(\mathcal{T}|\sigma^2, \mathbf{y})q(\mathcal{T}'|\mathcal{T})}\right\}$$

Birth Proposal

- We randomly generate $\mathcal{T}'$ as follows:

1. Randomly select a terminal node $b \in \{1, \ldots, B\}$ with probability $\frac{1}{B}$ where $B = |\mathcal{M}|$.
2. Introduce a new rule $v_b \sim \pi_v(v_b)$ and cutpoint $c_b \sim \pi_c(c_b)$ where π_v, π_c are typically discrete Uniform on the available variable, cutpoints.
3. Calculate

$$\alpha = min\left\{1, \frac{\pi(\mathcal{T}'|\sigma^2, \mathbf{y})q(\mathcal{T}|\mathcal{T}')}{\pi(\mathcal{T}|\sigma^2, \mathbf{y})q(\mathcal{T}'|\mathcal{T})}\right\}$$

4. Generate $u \sim \text{Uniform}(0, 1)$. If $u < \alpha$ then accept $\mathcal{T}'$ otherwise reject.

Birth Proposal

- In Step 3, note that

$$\begin{aligned}\pi(\mathcal{T}'|\sigma^2,\mathbf{y}) &= L(\eta^b_{j(l)}|\sigma^2,\mathbf{y})L(\eta^b_{j(r)}|\sigma^2,\mathbf{y})\pi(\eta^b_j \text{is internal})\\ &\quad\times\pi(\eta^b_{j(l)}\text{is terminal})\pi(\eta^b_{j(r)}\text{is terminal})\\ &\quad\times\pi_v(v^b_j = v_b)\pi_c(c^b_j = c_b)\end{aligned}$$

and

$$\begin{aligned}q(\mathcal{T}|\mathcal{T}') = q(\mathcal{T}' \to \mathcal{T}) &= \pi(\text{death proposal})\\ &\quad\times\pi(\text{kill } \eta^b_{j(l)},\eta^b_{j(r)}|\text{death proposal})\\ &= (1-\pi_b)\pi_{d,\eta^b_j}\end{aligned}$$

Birth Proposal

- Typically the probability of doing a birth proposal is $\pi_b = \frac{1}{2}$.

Birth Proposal

- Typically the probability of doing a birth proposal is $\pi_b = \frac{1}{2}$.
- And π_{d,η_j^b} is the probability of selecting node η_j^b to perform the death.

Birth Proposal

- Typically the probability of doing a birth proposal is $\pi_b = \frac{1}{2}$.
- And π_{d,η_j^b} is the probability of selecting node η_j^b to perform the death.
  - Usually this will be $\frac{1}{D'}$ where D' is the number of next-to-terminal-nodes in tree $\mathcal{T}'$.

Birth Proposal

- Typically the probability of doing a birth proposal is $\pi_b = \frac{1}{2}$.
- And π_{d,η_j^b} is the probability of selecting node η_j^b to perform the death.
    - Usually this will be $\frac{1}{D'}$ where D' is the number of next-to-terminal-nodes in tree $\mathcal{T}'$.
    - An exception is when we have the root node as our tree (obviously we can't perform a death). In this case $\pi_{d,\eta_j^b} = 0$.

Birth Proposal

- Analogously, for Step 3 note that

$$\pi(\mathcal{T}|\sigma^2, \mathbf{y}) \quad = \quad L(\eta_j^b|\sigma^2, \mathbf{y})\pi(\eta_j^b \text{ is terminal})$$

and

$$\begin{aligned} q(\mathcal{T}'|\mathcal{T}) = q(\mathcal{T} \to \mathcal{T}') &= \pi(\text{birth proposal}) \\ &\quad \times \pi(\text{birth at } \eta_j^b|\text{birth proposal}) \\ &\quad \times \pi_v(v_j^b = v_b)\pi_c(c_j^b = c_b) \\ &= \pi_b \pi_{b,\eta_j^b} \pi_v(v_j^b = v_b)\pi_c(c_j^b = c_b) \end{aligned}$$

Birth Proposal

- Typically $\pi_{b,\eta_j^b} = \frac{1}{B}$ where B is the number of terminal nodes in tree $\mathcal{T}$.

Birth Proposal

- Typically $\pi_{b,\eta_j^b} = \frac{1}{B}$ where B is the number of terminal nodes in tree $\mathcal{T}$.
  - An exception is when, for example, there is no variable or cutpoint available to birth at η_j^b. In this case $\pi_{b,\eta_j^b} = 0$.

Death Proposals

- As you might imagine, it works similarly to Birth proposals.

Death Proposals

- As you might imagine, it works similarly to Birth proposals.
- I will spare you the details.

Algorithm

- Let's recap our sampling algorithm.

† We might return to discussing more complex proposals for $\mathcal{T}$ later on...

Algorithm

- Let's recap our sampling algorithm.

1. †Draw $\mathcal{T}|\sigma^2, \mathbf{y}$

† We might return to discussing more complex proposals for $\mathcal{T}$ later on. . .

Algorithm

- Let's recap our sampling algorithm.

1. †Draw $\mathcal{T}|\sigma^2, \mathbf{y}$
    - With probability π_b do a birth proposal, otherwise a death proposal.

† We might return to discussing more complex proposals for $\mathcal{T}$ later on...

Algorithm

- Let's recap our sampling algorithm.

1. †Draw $\mathcal{T}|\sigma^2, \mathbf{y}$
   - With probability π_b do a birth proposal, otherwise a death proposal.
2. Draw $\mathcal{M}|\mathcal{T}, \sigma^2, \mathbf{y}$

† We might return to discussing more complex proposals for $\mathcal{T}$ later on...

Algorithm

- Let's recap our sampling algorithm.

1. †Draw $\mathcal{T}|\sigma^2, \mathbf{y}$
   - With probability π_b do a birth proposal, otherwise a death proposal.
2. Draw $\mathcal{M}|\mathcal{T}, \sigma^2, \mathbf{y}$
   - For $j = 1, \ldots, B$, perform our Gibbs steps by drawing

$$\mu_j|\sigma^2, \mathcal{T}, \mathbf{y} \sim N\left(\left(\frac{n_j}{\sigma^2} + \frac{1}{\sigma_\mu^2}\right)^{-1}\left(\frac{n_j\bar{y}_j}{\sigma^2}\right), \left(\frac{n_j}{\sigma^2} + \frac{1}{\sigma_\mu^2}\right)^{-1}\right)$$

† We might return to discussing more complex proposals for $\mathcal{T}$ later on...

Algorithm

- Let's recap our sampling algorithm.

1. †Draw $\mathcal{T}|\sigma^2, \mathbf{y}$
   - With probability π_b do a birth proposal, otherwise a death proposal.
2. Draw $\mathcal{M}|\mathcal{T}, \sigma^2, \mathbf{y}$
   - For $j = 1, \ldots, B$, perform our Gibbs steps by drawing

$$\mu_j|\sigma^2, \mathcal{T}, \mathbf{y} \sim N\left(\left(\frac{n_j}{\sigma^2} + \frac{1}{\sigma_\mu^2}\right)^{-1}\left(\frac{n_j\bar{y}_j}{\sigma^2}\right), \left(\frac{n_j}{\sigma^2} + \frac{1}{\sigma_\mu^2}\right)^{-1}\right)$$

3. Draw $\sigma^2|\mathcal{T}, \mathcal{M}, \mathbf{y}$

† We might return to discussing more complex proposals for $\mathcal{T}$ later on...

Algorithm

- Let's recap our sampling algorithm.

1. †Draw $\mathcal{T}|\sigma^2, \mathbf{y}$
   - With probability π_b do a birth proposal, otherwise a death proposal.
2. Draw $\mathcal{M}|\mathcal{T}, \sigma^2, \mathbf{y}$
   - For $j = 1, \ldots, B$, perform our Gibbs steps by drawing
   $$\mu_j|\sigma^2, \mathcal{T}, \mathbf{y} \sim N\left(\left(\frac{n_j}{\sigma^2} + \frac{1}{\sigma_\mu^2}\right)^{-1}\left(\frac{n_j\bar{y}_j}{\sigma^2}\right), \left(\frac{n_j}{\sigma^2} + \frac{1}{\sigma_\mu^2}\right)^{-1}\right)$$
3. Draw $\sigma^2|\mathcal{T}, \mathcal{M}, \mathbf{y}$
   - Perform our Gibbs step by drawing
   $$\sigma^2|\mathcal{T}, \mathcal{M}, \mathbf{y} \sim \chi^{-2}\left(\nu + n, \frac{\nu\tau^2 + ns^2}{\nu + n}\right)$$

† We might return to discussing more complex proposals for $\mathcal{T}$ later on. . .

Calibrating the Tree Prior

- For the depth penalizing prior,

$$\alpha(1 - d)^{-\beta}$$

typical values are $\alpha = 0.95$ and $\beta = 3$.

Calibrating the Tree Prior

- For the depth penalizing prior,

$$\alpha(1 - d)^{-\beta}$$

  typical values are $\alpha = 0.95$ and $\beta = 3$.
- The variables are selected with a discrete uniform prior.

Calibrating the Tree Prior

- For the depth penalizing prior,

$$\alpha(1-d)^{-\beta}$$

  typical values are $\alpha = 0.95$ and $\beta = 3$.
- The variables are selected with a discrete uniform prior.
- The cutpoints are selected with a discrete uniform prior.

Calibrating the Tree Prior

- For the depth penalizing prior,

$$\alpha(1 - d)^{-\beta}$$

  typical values are $\alpha = 0.95$ and $\beta = 3$.
- The variables are selected with a discrete uniform prior.
- The cutpoints are selected with a discrete uniform prior.
  - The number of cutpoints is hyperparameter we can choose. Default is numcuts $= 100$. This works well in general, sometimes we might like a more refined grid, say numcuts $= 1,000$.

Calibrating the variance prior, $\pi(\sigma^2|\nu, \tau^2)$

- ν is selected to get an "appropriate shape." Typical values are between 3 and 10, with $\nu = 3$ being the default.

Calibrating the variance prior, $\pi(\sigma^2|\nu, \tau^2)$

- ν is selected to get an "appropriate shape." Typical values are between 3 and 10, with $\nu = 3$ being the default.
- The scale parameter τ^2 is selected in the following way.

Calibrating the variance prior, $\pi(\sigma^2|\nu, \tau^2)$

- ν is selected to get an "appropriate shape." Typical values are between 3 and 10, with $\nu = 3$ being the default.
- The scale parameter τ^2 is selected in the following way.
  - Provide an initial estimate of the standard deviation of your data, $\hat{\sigma}$. Typically the sample standard deviation.

Calibrating the variance prior, $\pi(\sigma^2|\nu, \tau^2)$

- ν is selected to get an "appropriate shape." Typical values are between 3 and 10, with $\nu = 3$ being the default.
- The scale parameter τ^2 is selected in the following way.
  - Provide an initial estimate of the standard deviation of your data, $\hat{\sigma}$. Typically the sample standard deviation.
  - Provide an upper quantile q, with $q = 0.90$ being the default.

Calibrating the variance prior, $\pi(\sigma^2|\nu,\tau^2)$

- ν is selected to get an "appropriate shape." Typical values are between 3 and 10, with $\nu = 3$ being the default.
- The scale parameter τ^2 is selected in the following way.
  - Provide an initial estimate of the standard deviation of your data, $\hat{\sigma}$. Typically the sample standard deviation.
  - Provide an upper quantile q, with $q = 0.90$ being the default.
  - τ^2 is selected so that, a priori, $P(\sigma < \hat{\sigma}) = q$.

Calibrating the variance prior, $\pi(\sigma^2|\nu, \tau^2)$

- ν is selected to get an "appropriate shape." Typical values are between 3 and 10, with $\nu = 3$ being the default.
- The scale parameter τ^2 is selected in the following way.
  - Provide an initial estimate of the standard deviation of your data, $\hat{\sigma}$. Typically the sample standard deviation.
  - Provide an upper quantile q, with $q = 0.90$ being the default.
  - τ^2 is selected so that, a priori, $P(\sigma < \hat{\sigma}) = q$.
- The idea is that our data is unlikely all noise, so a conservative approach is to setup the prior such that it is very unlikely to estimate the variance to be greater than the sample variance of our data.

Calibrating the variance prior, $\pi(\sigma^2|\nu, \tau^2)$

- ν is selected to get an "appropriate shape." Typical values are between 3 and 10, with $\nu = 3$ being the default.
- The scale parameter τ^2 is selected in the following way.
  - Provide an initial estimate of the standard deviation of your data, $\hat{\sigma}$. Typically the sample standard deviation.
  - Provide an upper quantile q, with $q = 0.90$ being the default.
  - τ^2 is selected so that, a priori, $P(\sigma < \hat{\sigma}) = q$.
- The idea is that our data is unlikely all noise, so a conservative approach is to setup the prior such that it is very unlikely to estimate the variance to be greater than the sample variance of our data.
- The smaller ν the more concentrated on small σ the prior becomes.

Calibrating the mean prior, $\pi(\mu_j|\mathcal{T})$

- Assume the data is already mean-centered and scaled to $[-0.5, 0.5]$.

Calibrating the mean prior, $\pi(\mu_j|\mathcal{T})$

- Assume the data is already mean-centered and scaled to $[-0.5, 0.5]$.
- Then the prior for $\mu_j \sim N(\mu_\mu, \sigma_\mu^2)$ is set to have mean $\mu_\mu = 0$ and the variance is chosen such that

$$k\sigma_\mu = 0.5.$$

Calibrating the mean prior, $\pi(\mu_j|\mathcal{T})$

- Assume the data is already mean-centered and scaled to $[-0.5, 0.5]$.
- Then the prior for $\mu_j \sim N(\mu_\mu, \sigma_\mu^2)$ is set to have mean $\mu_\mu = 0$ and the variance is chosen such that

$$k\sigma_\mu = 0.5.$$

- In other words, this sets the prior up such that k standard deviations cover the range of the observed data.

Calibrating the mean prior, $\pi(\mu_j|\mathcal{T})$

- Assume the data is already mean-centered and scaled to $[-0.5, 0.5]$.
- Then the prior for $\mu_j \sim N(\mu_\mu, \sigma_\mu^2)$ is set to have mean $\mu_\mu = 0$ and the variance is chosen such that

$$k\sigma_\mu = 0.5.$$

- In other words, this sets the prior up such that k standard deviations cover the range of the observed data.
- The greater is k, the more shrinkage a priori is applied to the mean parameters. The default is $k = 2$.

Example

```
source("dace.sim.r")

# Generate response:
set.seed(88)
n=5; k=1; rhotrue=0.2; lambdatrue=1
design=as.matrix(runif(n))
l1=list(m1=outer(design[,1],design[,1],"-"))
l.dez=list(l1=l1)
R=rhogeodacecormat(l.dez,c(rhotrue))$R
L=t(chol(R))
u=rnorm(nrow(R))
z=L%*%u

# Our observed data:
y=as.vector(z)
```

Example

```
library(BayesTree)
preds=matrix(seq(0,1,length=100),ncol=1)

# Variance prior
shat=sd(y)
nu=3
q=0.90
# Mean prior
k=2
# Tree prior
alpha=0.95
beta=2
nc=100
# MCMC settings
N=1000
burn=1000
```

Example

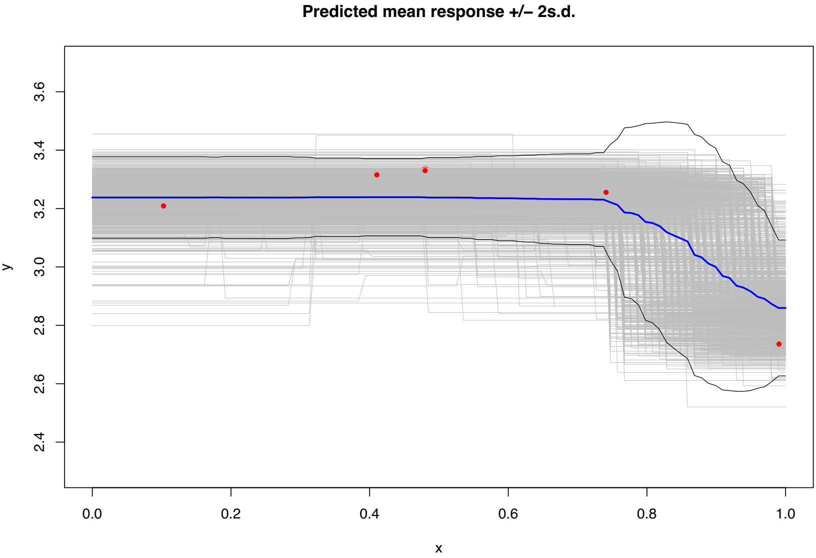
```
fit=bart(design,y,preds,sigest=shat,sigdf=nu,sigquant=q,
         k=k,power=beta,base=alpha,ntree=1,numcut=nc,
         ndpost=N,nskip=burn)
```

```
##
##
## Running BART with numeric y
##
## number of trees: 1
## Prior:
##  k: 2.000000
##  degrees of freedom in sigma prior: 3
##  quantile in sigma prior: 0.900000
##  power and base for tree prior: 2.000000 0.950000
##  use quantiles for rule cut points: 0
## data:
##  number of training observations: 5
```

Example

```
plot(design,y,pch=20,col="red",cex=2,xlim=c(0,1),
    ylim=c(2.3,3.7),xlab="x",
    main="Predicted mean response +/- 2s.d.")
for(i in 1:nrow(fit$yhat.test))
  lines(preds,fit$yhat.test[i,],col="grey",lwd=0.25)
mean=apply(fit$yhat.test,2,mean)
sd=apply(fit$yhat.test,2,sd)
lines(preds,mean-1.96*sd,lwd=0.75,col="black")
lines(preds,mean+1.96*sd,lwd=0.75,col="black")
lines(preds,mean,lwd=2,col="blue")
points(design,y,pch=20,col="red")
```

Example

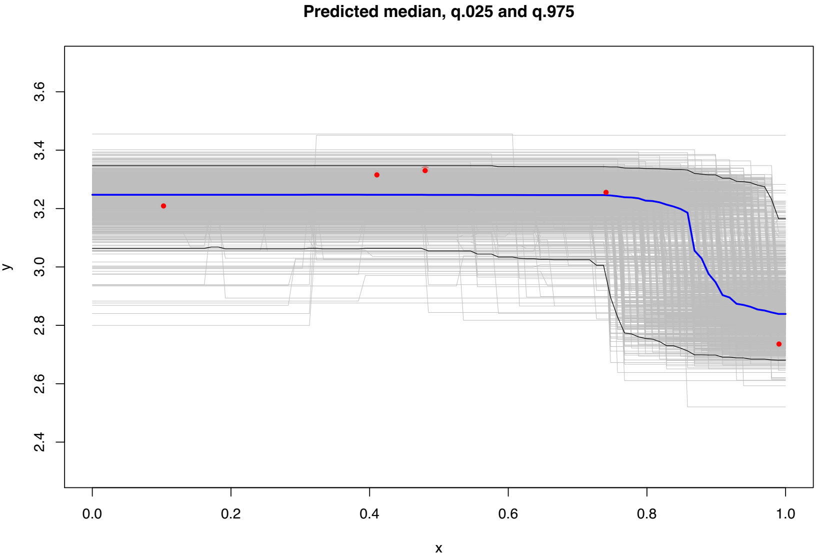


Predicted mean response +/– 2s.d.
3.6
3.4
3.2
3.0
2.8
2.6
2.4
y
0.0
0.2
0.4
0.6
0.8
1.0
x

Example

```
plot(design,y,pch=20,col="red",cex=2,xlim=c(0,1),ylim=c(2.3
    xlab="x",main="Predicted median, q.025 and q.975")
for(i in 1:nrow(fit$yhat.test))
  lines(preds,fit$yhat.test[i,],col="grey",lwd=0.25)
med=apply(fit$yhat.test,2,quantile,0.5)
q.025=apply(fit$yhat.test,2,quantile,0.025)
q.975=apply(fit$yhat.test,2,quantile,0.975)
lines(preds,q.025,lwd=0.75,col="black")
lines(preds,q.975,lwd=0.75,col="black")
lines(preds,med,lwd=2,col="blue")
points(design,y,pch=20,col="red")
```

Example



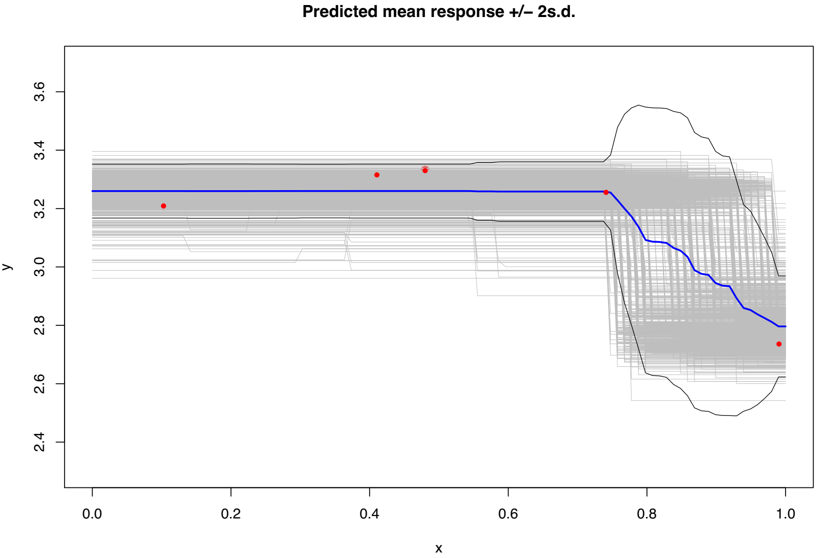
Predicted median, q.025 and q.975
3.6
3.4
3.2
3.0
2.8
2.6
2.4
y
0.0
0.2
0.4
0.6
0.8
1.0
x

Example

```
nu=1
fit=bart(design,y,preds,sigest=shat,sigdf=nu,sigquant=q,
        k=k,power=beta,base=alpha,ntree=1,numcut=nc,
        ndpost=N,nskip=burn)
```

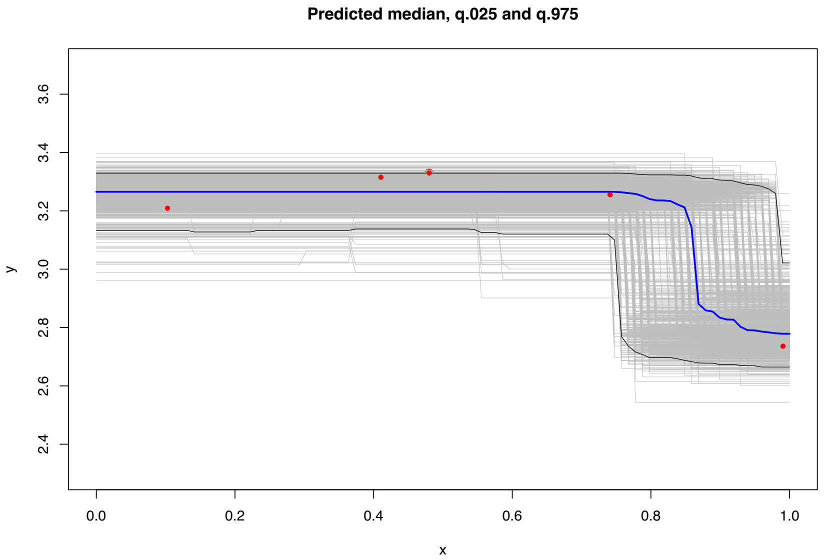
```
##
##
## Running BART with numeric y
##
## number of trees: 1
## Prior:
##  k: 2.000000
##  degrees of freedom in sigma prior: 1
##  quantile in sigma prior: 0.900000
##  power and base for tree prior: 2.000000 0.950000
##  use quantiles for rule cut points: 0
## data:
```

Example



Predicted mean response +/– 2s.d.
3.6
3.4
3.2
3.0
2.8
2.6
2.4
y
0.0
0.2
0.4
0.6
0.8
1.0
x

Example

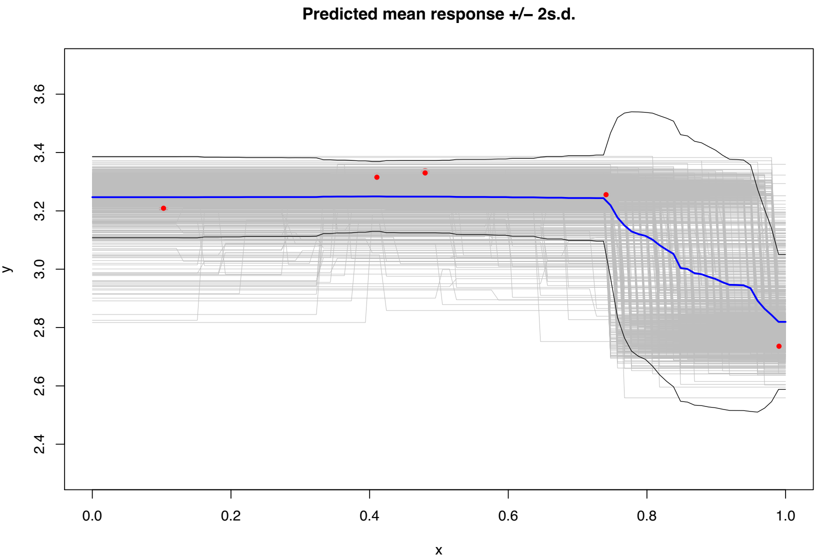

Predicted median, q.025 and q.975
y
x
3.6
3.4
3.2
3.0
2.8
2.6
2.4
0.0
0.2
0.4
0.6
0.8
1.0

Example

```
nu=1
nc=1000
fit=bart(design,y,preds,sigest=shat,sigdf=nu,sigquant=q,
        k=k,power=beta,base=alpha,ntree=1,numcut=nc,
        ndpost=N,nskip=burn)
```

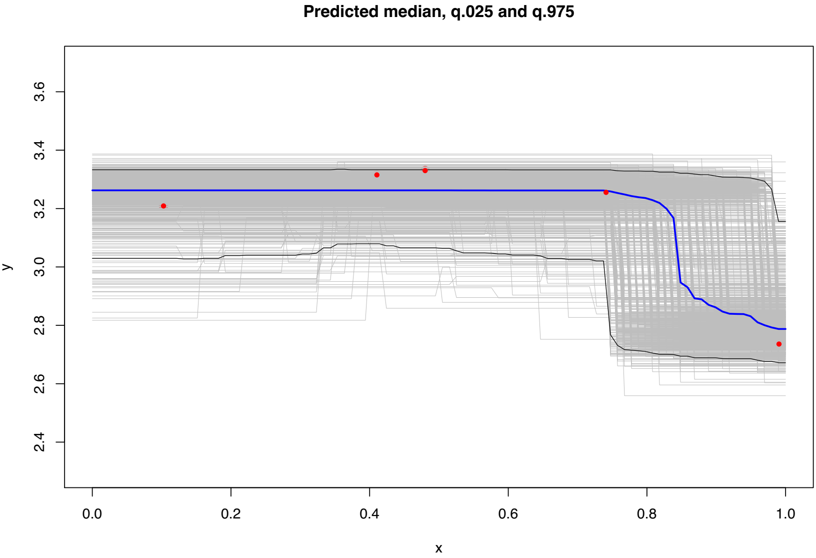
```
##
##
## Running BART with numeric y
##
## number of trees: 1
## Prior:
##  k: 2.000000
##  degrees of freedom in sigma prior: 1
##  quantile in sigma prior: 0.900000
##  power and base for tree prior: 2.000000 0.950000
##  use quantiles for rule cut points: 0
```

Example



Predicted mean response +/– 2s.d.
3.6
3.4
3.2
3.0
2.8
2.6
2.4
y
0.0
0.2
0.4
0.6
0.8
1.0
x

Example



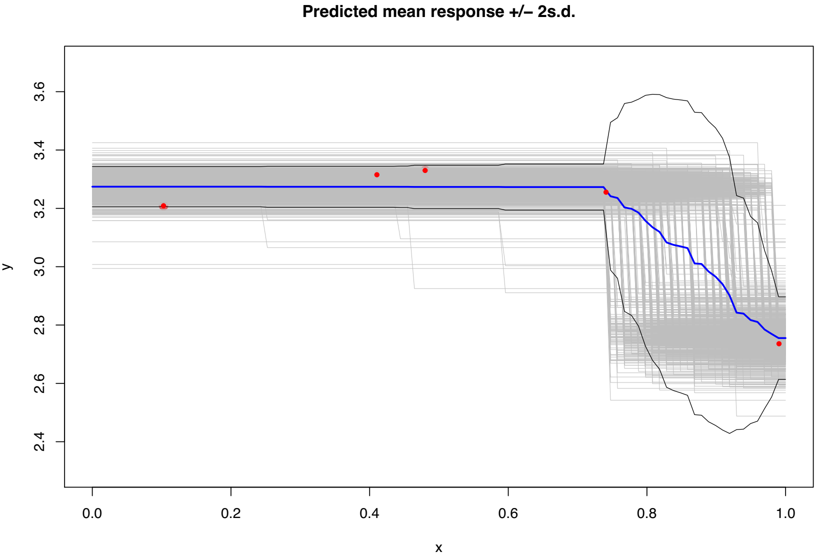
Predicted median, q.025 and q.975
3.6
3.4
3.2
3.0
2.8
2.6
2.4
y
0.0
0.2
0.4
0.6
0.8
1.0
x

Example

```
nu=1
k=1
nc=100
fit=bart(design,y,preds,sigest=shat,sigdf=nu,sigquant=q,
        k=k,power=beta,base=alpha,ntree=1,numcut=nc,
        ndpost=N,nskip=burn)
```

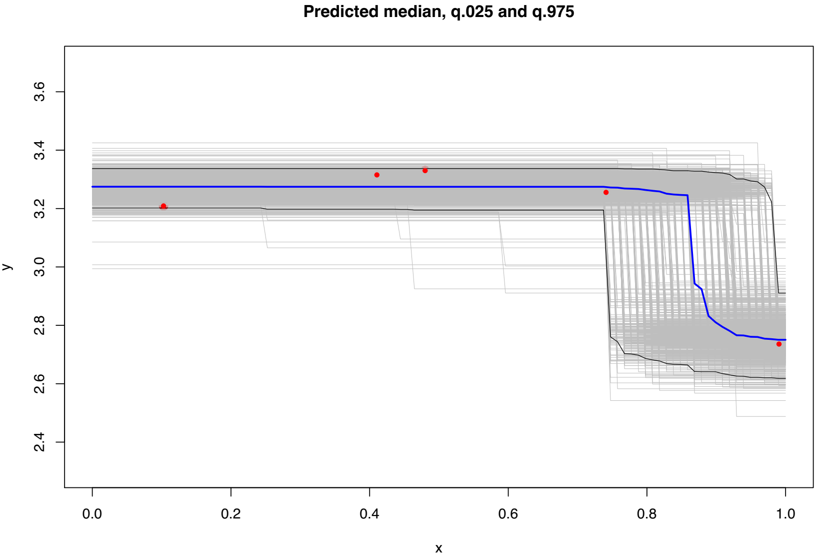
```
##
##
## Running BART with numeric y
##
## number of trees: 1
## Prior:
##  k: 1.000000
##  degrees of freedom in sigma prior: 1
##  quantile in sigma prior: 0.900000
##  power and base for tree prior: 2.000000 0.950000
```

Example



Predicted mean response +/– 2s.d.
3.6
3.4
3.2
3.0
2.8
2.6
2.4
y
0.0
0.2
0.4
0.6
0.8
1.0
x

Example


Predicted median, q.025 and q.975
y
3.6
3.4
3.2
3.0
2.8
2.6
2.4
0.0
0.2
0.4
0.6
0.8
1.0
x