BART

STAT8810, Fall 2017

M.T. Pratola

November 1, 2017

Today

BART: Bayesian Additive Regression Trees

BART: Bayesian Additive Regression Trees

- Additive model generalizes the single-tree regression model:

$$Y(\mathbf{x}) = f(\mathbf{x}) + \epsilon, \quad \epsilon \sim N(0, \sigma^2)$$

where

$$f(\mathbf{x}) = \sum_{j=1}^{m} g(\mathbf{x}; \mathcal{T}_j, \mathcal{M}_j).$$

- We viewed each tree as representing a map $g(\mathbf{x}; \mathcal{T}_j, \mathcal{M}_j) : \mathbb{R}^p \to \mathbb{R}$. Can get a richer class of models by considering the sum of many such maps.
- We will see that each individual function $g(\mathbf{x}; \mathcal{T}_j, \mathcal{M}_j)$ is constrained to be a simplistic function that explains only a small portion of the response variability.
  - so-called "sum of weak-learners" assumption.

BART: Bayesian Additive Regression Trees

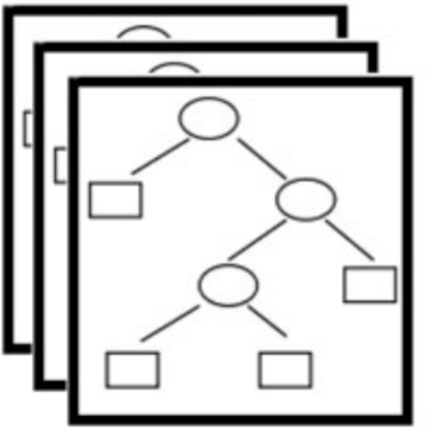


Figure 1: BART uses a sum of many simple trees.

BART: Bayesian Additive Regression Trees

- There is, of course, nothing new about a GAM-like formulation.
- However, one advantage of the tree-based approach is the tree's natural ability to capture interactions, possibly of a high-dimensional form. Or, to not capture such behavior if not present.
- That is, in the tree based approach we are learning the form of predictor functions themselves rather than assuming a fixed class of bases with a particular form.

BART Model

- The data is modeled as

$$Y(\mathbf{x}) | \{(\mathcal{T}_j, \mathcal{M}_j)\}_{j=1}^{m}, \sigma^2 \sim N\left(\sum_{j=1}^{m} g(\mathbf{x}; \mathcal{T}_j, \mathcal{M}_j), \sigma^2\right)$$

  giving our likelihood,

$$L(\{(\mathcal{T}_j, \mathcal{M}_j)\}_{j=1}^{m}, \sigma^2 | \mathbf{Y}) =$$
$$\frac{1}{(2\pi\sigma^2)^{n/2}} exp\left(\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y(\mathbf{x}_i) - \sum_{j=1}^{m} g(\mathbf{x}_i | \mathcal{T}_j, \mathcal{M}_j)\right)^2\right)$$

  where $\mathbf{Y} = (y(\mathbf{x}_1), \ldots, y(\mathbf{x}_n))$.

BART Model

- Default number of trees is $m = 200$, which seems to work well in many problems. Increasing m allows one to have a model with greater fidelity to more complex responses.
- Interpretation is as follows. We can view $g(\mathbf{x}; \mathcal{T}_j, \mathcal{M}_j)$ as a function that assigns a terminal node scalar μ_{jb} for a given input $\mathbf{x}$.
- And so, the expected response $E[Y(\mathbf{x})| \{(\mathcal{T}_j, \mathcal{M}_j)\}_{j=1}^m]$ is simply the sum of all such μ_{jb}'s that is assigned to $\mathbf{x}$ by each tree $(\mathcal{T}_j, \mathcal{M}_j)$.

BART Priors

- Similar to our single-tree model, the BART prior is factored as

$$\pi(\{(\mathcal{T}_j, \mathcal{M}_j)\}_{j=1}^m, \sigma^2) = \pi(\sigma^2) \prod_{j=1}^m \pi(\mathcal{M}_j|\mathcal{T}_j)\pi(\mathcal{T}_j)$$

where for each j,

$$\pi(\mathcal{M}_j|\mathcal{T}_j) = \prod_{b=1}^{B_j} \pi(\mu_{jb}),$$

where $B_j = |\mathcal{M}_j|$ is the number of terminal nodes in tree $\mathcal{T}_j$.

BART Priors: $\pi(\mathcal{T}_j)$

- The interior of the tree $\mathcal{T}_j$ is made up of split rules, $\{(v_{ji}, c_{ji})\}_{i=1}^{|\mathcal{T}_j|}$ with discrete uniform priors as in our single-tree model,

$$\pi(\mathbf{v}_j) = \prod_i \pi(v_{ji})$$

and

$$\pi(\mathbf{c}_j) = \prod_i \pi(c_{ji}|v_{ji}, \mathcal{T}_j \backslash v_{ji}).$$

- And the tree is regularized via the depth-penalizing prior from before as well,

$$\pi(\eta_{ji} \text{ splits}) = \alpha(1 + d(\eta_{ji}, \eta_{j1}))^{-\beta}$$

where $d(\eta_{ji}, \eta_{j1})$ is the depth from node η_{ji} to the root node in tree $\mathcal{T}_j$.

- The default prior is the same as our single-tree model: $\alpha = 0.95, \beta = 2$.

BART Priors: $\pi(\mu_{ji}|\mathcal{T}_j)$

- The prior on the scalar terminal node parameters is the conjugate Normal,

$$\mu_{ji} \sim N(\mu_\mu, \sigma_\mu^2).$$

- Note that this prior implies a priori that the prior on $E[Y(\mathbf{x})|\{(\mathcal{T}_j, \mathcal{M}_j)\}_{j=1}^m]$ is

$$N(m\mu_\mu, m\sigma_\mu^2).$$

- In practice, data is usually centered to have mean zero and scaled so $(y_{min}, y_{max}) = (-0.5, 0.5)$ and the prior used is

$$\mu_{ji} \sim N(0, \sigma_\mu^2).$$

BART Priors: $\pi(\mu_{ji}|\mathcal{T}_j)$

- The induced prior on $E[Y(\mathbf{x})|\{(\mathcal{T}_j, \mathcal{M}_j)\}_{j=1}^m]$ is

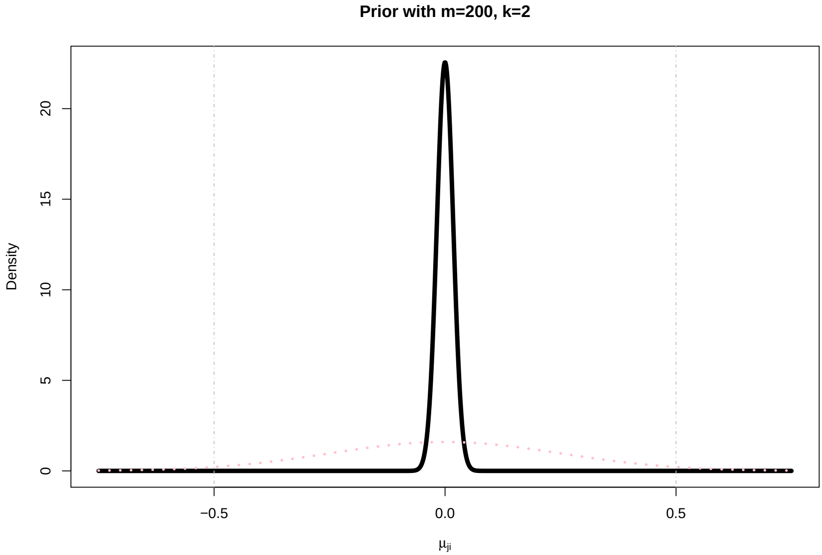
$$N(0, m\sigma_\mu^2).$$

- The strategy to calibrate this prior is the same as the single-tree model: choose a value k such that $k\sqrt{m}\sigma_\mu = 0.5$ which implies that the prior is

$$\mu_{ji} \sim N(0, \frac{0.5}{k\sqrt{m}}).$$

- As in the single-tree model, the default value recommended is $k = 2$.

BART Priors: $\pi(\mu_{ji}|\mathcal{T}_j)$



BART Priors: $\pi(\mu_{ji}|\mathcal{T}_j)$

- This means that we induce further shrinkage of the μ_{ji}'s towards zero by increasing k, **or** by increasing m.
- **However**, for m fixed, increasing k implies more of the response variability ends up in σ^2.
- While for k fixed, increasing m implies more of the response variability ends up in $f(\mathbf{x})$.
- Besides the default choice of $k = 2$, one might try tuning this prior hyperparameter using cross-validation.

BART Priors: $\pi(\sigma^2)$

- The variance prior is again

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

  and is calibrated similarly as in the single-tree model.
- ν is selected to get an "appropriate shape." Typical values are between 3 and 10, with $\nu = 3$ being the default.

BART Priors: $\pi(\sigma^2)$

- 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.

Sampling BART's Posterior

- The posterior is

$$\pi(\{(\mathcal{T}_j, \mathcal{M}_j)\}_{j=1}^m, \sigma^2|\mathbf{Y}) \propto$$

$$L(\{(\mathcal{T}_j, \mathcal{M}_j)\}_{j=1}^m, \sigma^2|\mathbf{Y})\pi(\sigma^2)\prod_{j=1}^{m}\pi(\mathcal{M}_j|\mathcal{T}_j)\pi(\mathcal{T}_j)$$

Sampling BART's Posterior

- First, note the following:

$$
\begin{aligned}
& L(\{(\mathcal{T}_j, \mathcal{M}_j)\}_{j=1}^{m}, \sigma^2 | \mathbf{Y}) \\
= \quad & \frac{1}{(2\pi\sigma^2)^{n/2}} exp\left(\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y(\mathbf{x}_i) - \sum_{j=1}^{m} g(\mathbf{x}_i | \mathcal{T}_j, \mathcal{M}_j)\right)^2\right) \\
= \quad & \frac{1}{(2\pi\sigma^2)^{n/2}} exp\left(\frac{1}{2\sigma^2}\sum_{i=1}^{n}(r_j(\mathbf{x}_i) - g(\mathbf{x}_i | \mathcal{T}_j, \mathcal{M}_j))^2\right)
\end{aligned}
$$

where $r_j(\mathbf{x}_i) = y(\mathbf{x}_i) - \sum_{k \neq j}^{m} g(\mathbf{x}_i | \mathcal{T}_k, \mathcal{M}_k)$.

Sampling BART's Posterior

- Our MCMC algorithm will perform the following steps:

1. For $j = 1, \ldots, m$:
   1.1 Draw $\mathcal{T}_j | \sigma^2, \mathbf{R}_j$ where $\mathbf{R}_j = (r_j(\mathbf{x}_1), \ldots, r_j(\mathbf{x}_n))$
      - Metropolis-Hastings step via proposal distribution

   1.2 Draw $\mathcal{M}_j | \mathcal{T}_j, \sigma^2, \mathbf{R}_j$
      - Gibbs step using conjugate prior
2. Draw $\sigma^2 | \left\{ (\mathcal{T}_j, \mathcal{M}_j) \right\}_{j=1}^m, \mathbf{Y}$
   - Gibbs step using conjugate prior

- So once we have the $\mathbf{R}_j$'s, the algorithm proceeds similarly as the single-tree algorithm.

Draw $\sigma^2 | \{(\mathcal{T}_j, \mathcal{M}_j)\}_{j=1}^{m}, \mathbf{Y}$

- 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,

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

where $s^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^2$ and $f(\mathbf{x}_i) = \sum_{j=1}^{m} g(\mathbf{x}_i; \mathcal{T}_j, \mathcal{M}_j)$.

Draw $\sigma^2 | \{(\mathcal{T}_j, \mathcal{M}_j)\}_{j=1}^m , \mathbf{Y}$

- 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

$$\sigma^2 | \{(\mathcal{T}_j, \mathcal{M}_j)\}_{j=1}^m , \mathbf{Y} \sim \chi^{-2}\left(\nu + n, \frac{\nu\tau^2 + ns^2}{\nu + n}\right)$$

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

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

- Suppose there are B terminal nodes in tree $\mathcal{T}_j, \eta^b_{j1}, \ldots, \eta^b_{jB}$. Using the same factorization as the single-tree case:

$$
\begin{aligned}
L(\sigma^2, \mathcal{T}_j, \mathcal{M}_j|\mathbf{R}_j) &\propto exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(r_j(\mathbf{x}_i) - g(\mathbf{x}_i|\mathcal{T}_j, \mathcal{M}_j))^2\right) \\
&= exp\left(-\frac{1}{2\sigma^2}\sum_{k=1}^{B}\sum_{i:r_j(\mathbf{x}_i)\in\eta^b_k}^{n_k}(r_j(\mathbf{x}_i) - \mu_{jk})^2\right) \\
&= \prod_{k=1}^{B} exp\left(-\frac{1}{2\sigma^2}\sum_{i:r_j(\mathbf{x}_i)\in\eta^b_k}^{n_k}(r_j(\mathbf{x}_i) - \mu_{jk})^2\right)
\end{aligned}
$$

where n_k is the number of observations mapping to terminal nodes η^b_{jk} and $\sum_k n_k = n$.

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

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

Draw $\mu_{jk} | \mathcal{T}_j, \sigma^2, \mathbf{R}_j$

- Assuming mean-centered observations, our prior is

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

- Based on our Normal-Normal conjugacy results, the full conditional is

$$\pi(\mu_{jk} | \sigma^2, \mathcal{T}_j, \mathbf{R}_j) \sim N\left(\left(\frac{n_k}{\sigma^2} + \frac{1}{\sigma_\mu^2}\right)^{-1}\left(\frac{n_k \bar{r}_{jk}}{\sigma^2}\right), \left(\frac{n_k}{\sigma^2} + \frac{1}{\sigma_\mu^2}\right)^{-1}\right)$$

where $\bar{r}_{jk} = \frac{1}{n_k}\sum_{i: r_j(\mathbf{x}_i) \in \eta_k^b} r_j(\mathbf{x}_i)$.

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

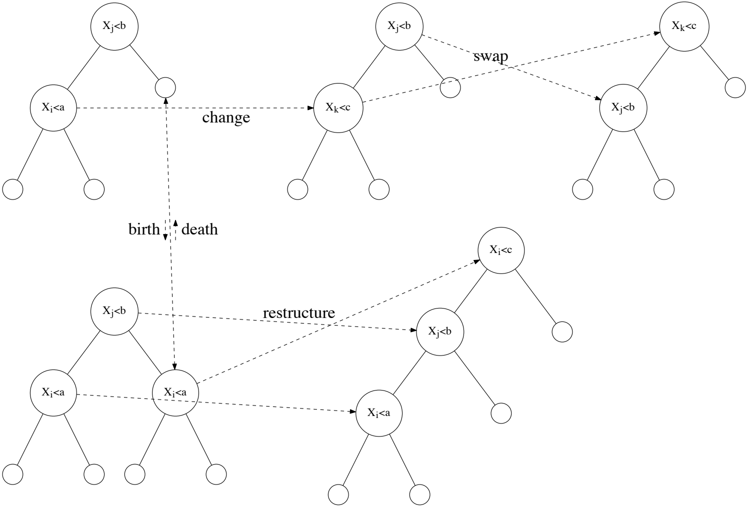


Figure 2: Tree Moves

Marginal Likelihood

- We will again need to marginalize our likelihood over the μ parameters.
- Marginalizing the portion of the likelihood associated with terminal node η_k^b, we have

$$L(\eta_{jk}^b|\sigma^2, \mathbf{R}_j) = \int_{\mu_{jk}} L(\eta_{jk}^b|\mu_{jk}, \sigma^2, \mathbf{R}_j)\pi(\mu_{jk})d\mu_{jk}$$

Birth Proposal

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

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

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

- As mentioned in the single-tree model, death proposals work similarly.

MCMC Algorithm

- Let's recap our sampling algorithm.

1. For $j = 1, \ldots, m$:
   1.1 Draw $\mathcal{T}_j | \sigma^2, \mathbf{R}_j$
      - With probability π_b do a birth proposal, otherwise a death proposal.
      - More complex moves possible, such as changing variable/cutpoints of existing tree.

   1.2 Draw $\mathcal{M}_j | \mathcal{T}_j, \sigma^2, \mathbf{R}_j$
      - For $k = 1, \ldots, B_j$, perform our Gibbs steps by drawing

$$\mu_{jk} | \sigma^2, \mathcal{T}_j, \mathbf{R}_j \sim N\left(\left(\frac{n_k}{\sigma^2} + \frac{1}{\sigma_\mu^2}\right)^{-1}\left(\frac{n_k \bar{r}_{jk}}{\sigma^2}\right), \left(\frac{n_k}{\sigma^2} + \frac{1}{\sigma_\mu^2}\right)^{-1}\right)$$

2. Draw $\sigma^2 | \left\{(\mathcal{T}_j, \mathcal{M}_j)\right\}_{j=1}^m, \mathbf{Y}$
   - Perform our Gibbs step by drawing

$$\sigma^2 | \left\{(\mathcal{T}_j, \mathcal{M}_j)\right\}_{j=1}^m, \mathbf{Y} \sim \chi^{-2}\left(\nu + n, \frac{\nu\tau^2 + ns^2}{\nu + n}\right)$$

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
m=1
alpha=0.95
beta=2
nc=100
# MCMC settings
N=1000
burn=1000
```

Example

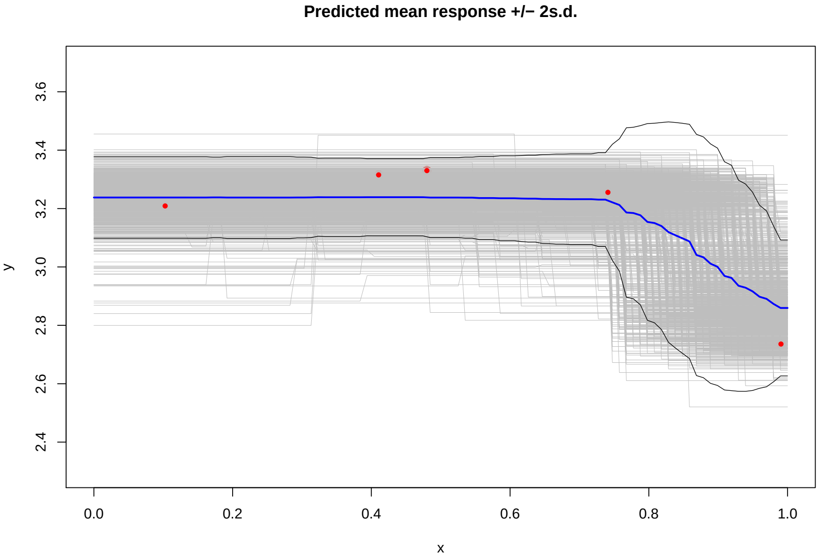
```
fit=bart(design,y,preds,sigest=shat,sigdf=nu,sigquant=q,
         k=k,power=beta,base=alpha,ntree=m,numcut=nc,
         ndpost=N,nskip=burn)
```

```
##
##
## 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



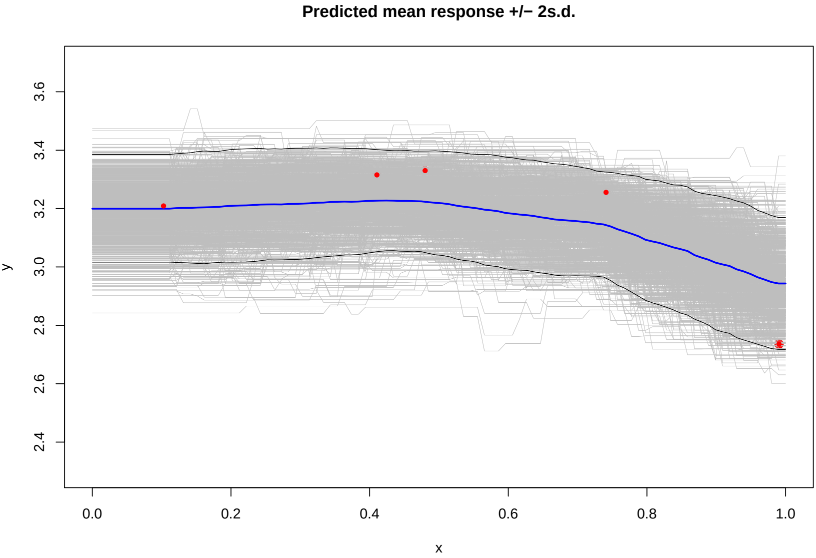
Predicted mean response +/− 2s.d.
3.6
3.4
3.2
3.0
2.8
2.6
2.4
y
0.0
0.2
0.4
0.6
0.8
1.0
x

Example

```
# Try m=10 trees
m=10
fit=bart(design,y,preds,sigest=shat,sigdf=nu,sigquant=q,
        k=k,power=beta,base=alpha,ntree=m,numcut=nc,
        ndpost=N,nskip=burn)
```

```
##
##
## Running BART with numeric y
##
## number of trees: 10
## 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
```

Example



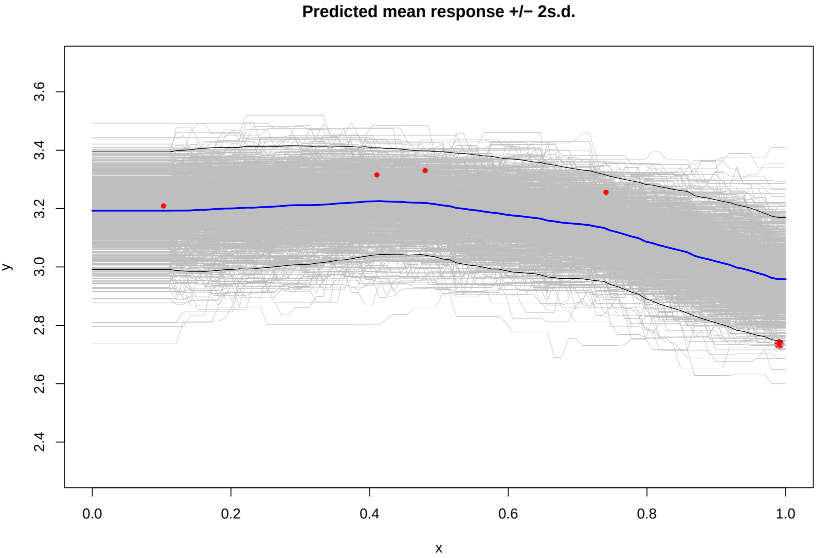
Predicted mean response +/− 2s.d.
3.6
3.4
3.2
3.0
2.8
2.6
2.4
y
0.0
0.2
0.4
0.6
0.8
1.0
x

Example

```
# Try m=20 trees
m=20
fit=bart(design,y,preds,sigest=shat,sigdf=nu,sigquant=q,
        k=k,power=beta,base=alpha,ntree=m,numcut=nc,
        ndpost=N,nskip=burn)
```

```
##
##
## Running BART with numeric y
##
## number of trees: 20
## 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
```

Example



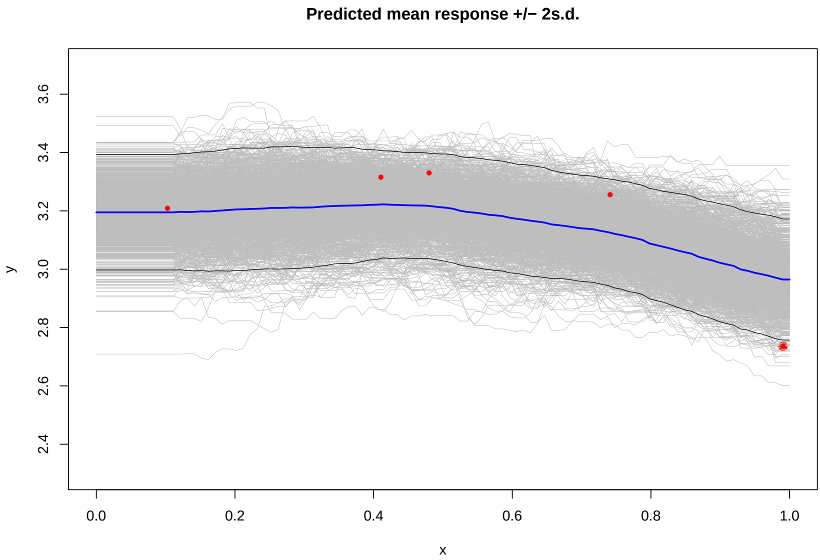
Predicted mean response +/− 2s.d.
y
x
3.6
3.4
3.2
3.0
2.8
2.6
2.4
0.0
0.2
0.4
0.6
0.8
1.0

Example

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

```
##
##
## Running BART with numeric y
##
## number of trees: 100
## 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
```

Example

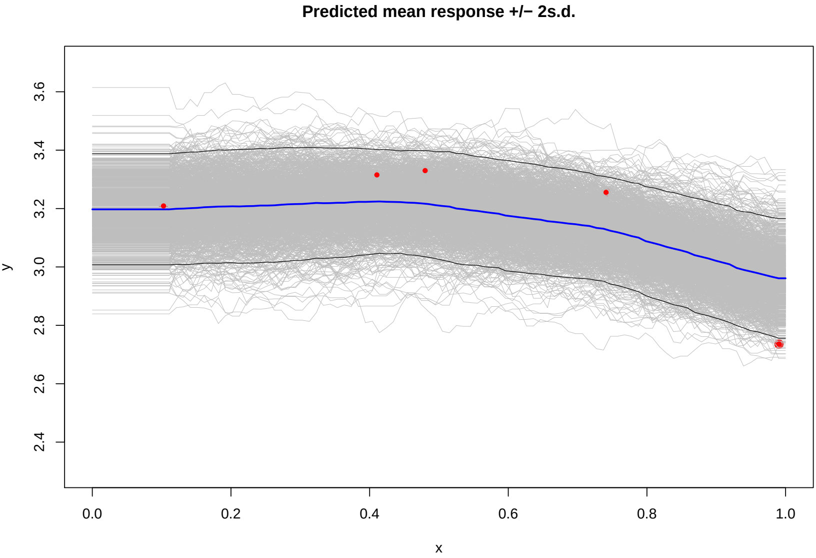

Predicted mean response +/− 2s.d.
y
3.6
3.4
3.2
3.0
2.8
2.6
2.4
x
0.0
0.2
0.4
0.6
0.8
1.0

Example

```
# Try m=200 trees, the recommended default
m=200
fit=bart(design,y,preds,sigest=shat,sigdf=nu,sigquant=q,
        k=k,power=beta,base=alpha,ntree=m,numcut=nc,
        ndpost=N,nskip=burn)
```

```
##
##
## Running BART with numeric y
##
## number of trees: 200
## 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
```

Example



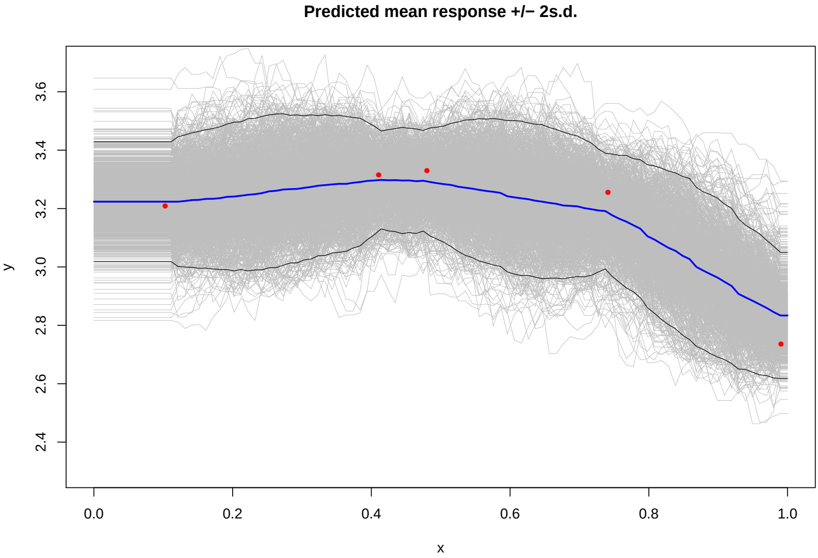
Predicted mean response +/− 2s.d.
3.6
3.4
3.2
3.0
2.8
2.6
2.4
y
0.0
0.2
0.4
0.6
0.8
1.0
x

Example

```
# Try m=200 trees, the recommended default
m=200
# And k=1
k=1
fit=bart(design,y,preds,sigest=shat,sigdf=nu,sigquant=q,
         k=k,power=beta,base=alpha,ntree=m,numcut=nc,
         ndpost=N,nskip=burn)
```

```
##
##
## Running BART with numeric y
##
## number of trees: 200
## Prior:
##  k: 1.000000
##  degrees of freedom in sigma prior: 3
##  quantile in sigma prior: 0.900000
```

Example


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

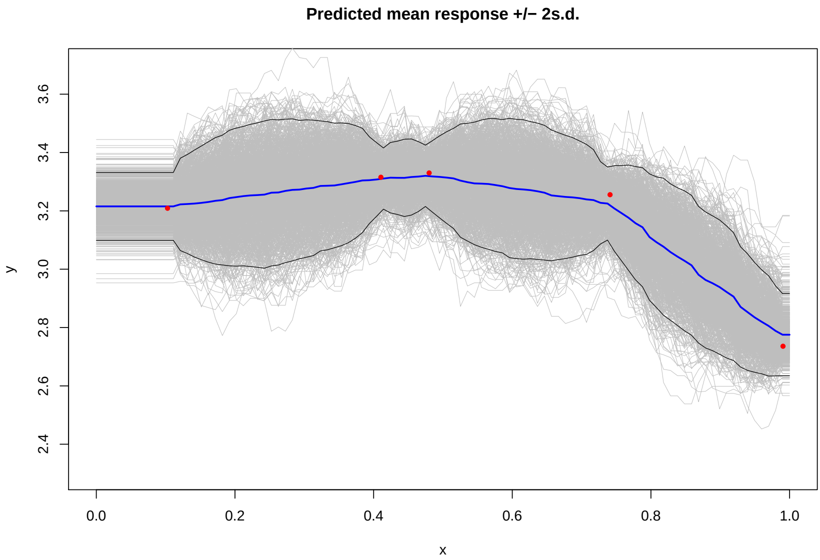
Example

```
# Try m=200 trees, the recommended default
m=200
# And k=1
k=1
# And nu=3, q=.99
nu=3
q=0.99

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

```
##
##
## Running BART with numeric y
##
## number of trees: 200
```

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

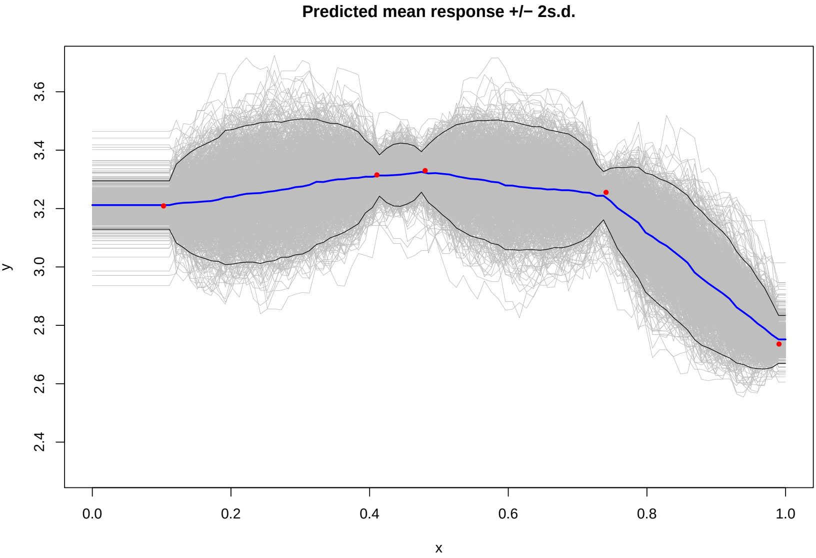
Example

```
# Try m=200 trees, the recommended default
m=200
# And k=1
k=1
# And nu=2, q=.99
nu=2
q=0.99

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

```
##
##
## Running BART with numeric y
##
## number of trees: 200
```

Example



Predicted mean response +/− 2s.d.
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

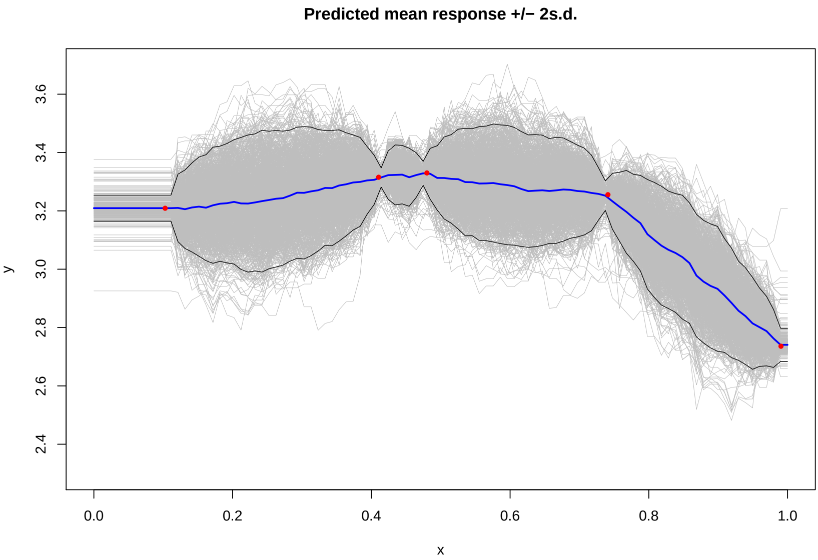
Example

```
# Try m=200 trees, the recommended default
m=200
# And k=1
k=1
# And nu=1, q=.99
nu=1
q=0.99

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

```
##
##
## Running BART with numeric y
##
## number of trees: 200
```

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

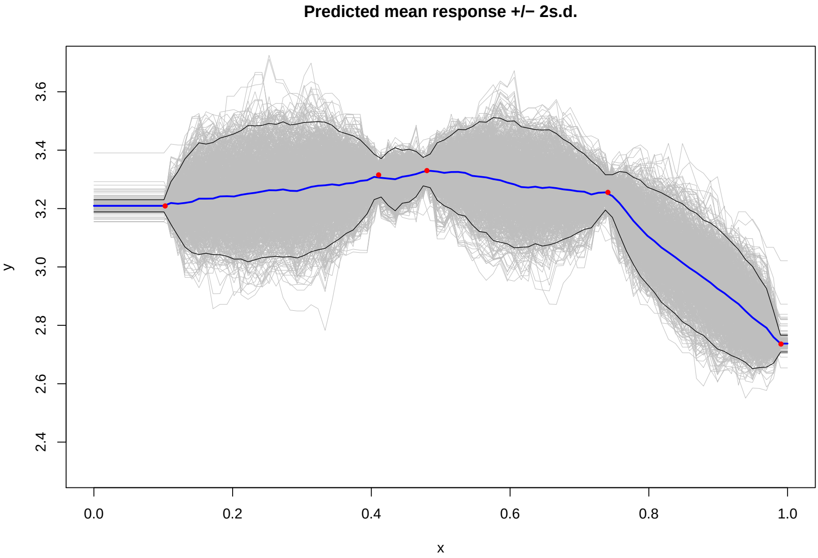
Example

```
# Try m=200 trees, the recommended default
m=200
# And k=1
k=1
# And nu=1, q=.99
nu=1
q=0.99
# And numcuts=1000
nc=1000

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

```
##
##
## Running BART with numeric y
```

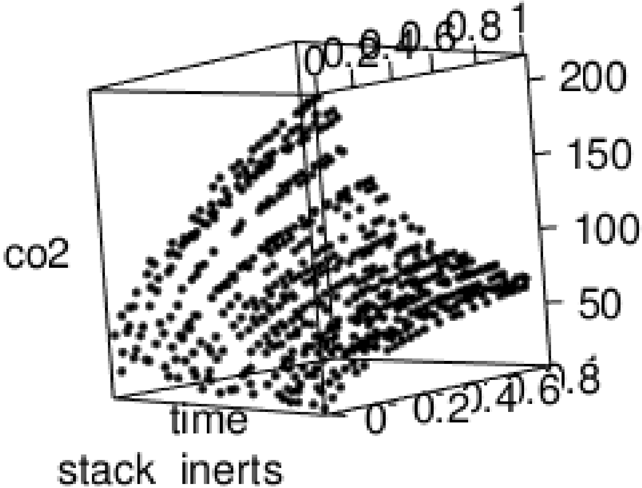
Example


Predicted mean response +/− 2s.d.
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

```
library(rgl)
load("co2plume.dat")
plot3d(co2plume)
rgl.snapshot("co2a.png")
```

Example



Example

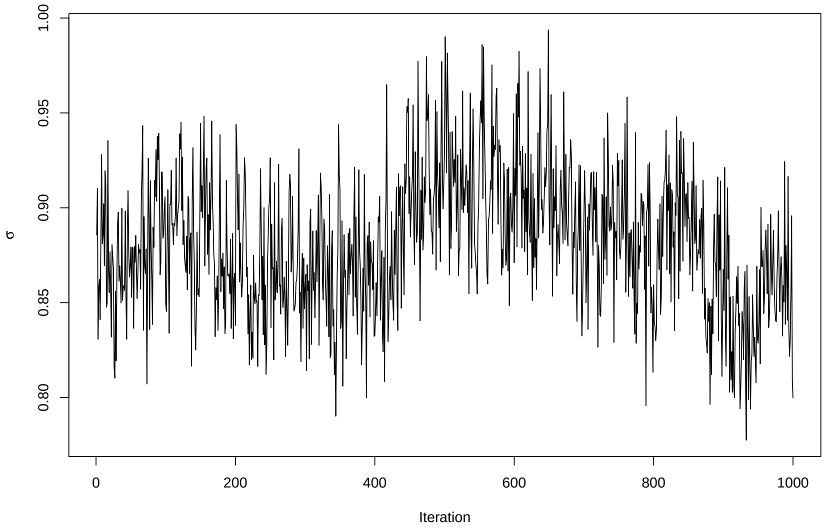
```
y=co2plume$co2
x=co2plume[,1:2]
preds=as.data.frame(expand.grid(seq(0,1,length=20),
        seq(0,1,length=20)))
colnames(preds)=colnames(x)
shat=sd(y)
# Try m=200 trees, the recommended default
m=200
# And k=1
k=1
# And nu=1, q=.99
nu=1
q=0.99
# And numcuts=1000
nc=1000

fit=bart(x,y,preds,sigest=shat,sigdf=nu,sigquant=q,
```

Example

```
plot(fit$sigma,type='l',xlab="Iteration",
    ylab=expression(sigma))
```

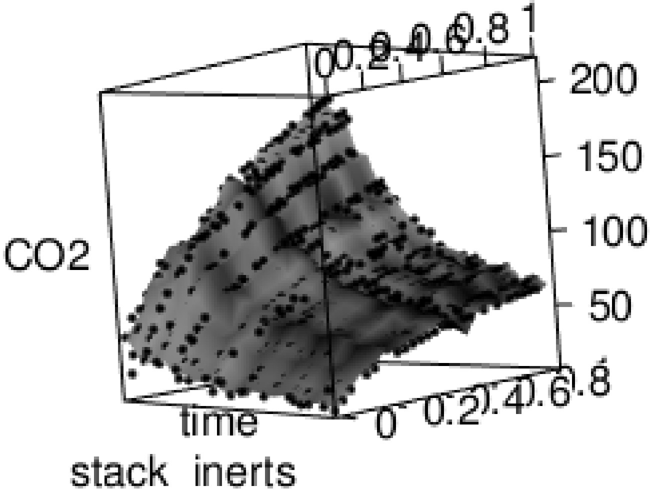
Example



Example

```
ym=fit$yhat.test.mean
ysd=apply(fit$yhat.test,2,sd)
persp3d(x=seq(0,1,length=20),y=seq(0,1,length=20),z=matrix(
        col="grey",xlab="stack_inerts",ylab="time",zlab="CO
plot3d(co2plume,add=TRUE)
rgl.snapshot("co2b.png")
```

Example



Example

```
persp3d(x=seq(0,1,length=20),y=seq(0,1,length=20),z=matrix(
        col="grey",xlab="stack_inerts",ylab="time",zlab="CO
persp3d(x=seq(0,1,length=20),y=seq(0,1,length=20),
        z=matrix(ym+2*ysd,20,20),col="green",add=TRUE)
persp3d(x=seq(0,1,length=20),y=seq(0,1,length=20),
        z=matrix(ym-2*ysd,20,20),col="green",add=TRUE)
plot3d(co2plume,add=TRUE)
rgl.snapshot("co2c.png")
```

Example