Nonparametric machine learning and efficient computation with Bayesian Additive Regression Trees (BART)

Rodney Sparapani
Associate Professor of Biostatistics
Medical College of Wisconsin

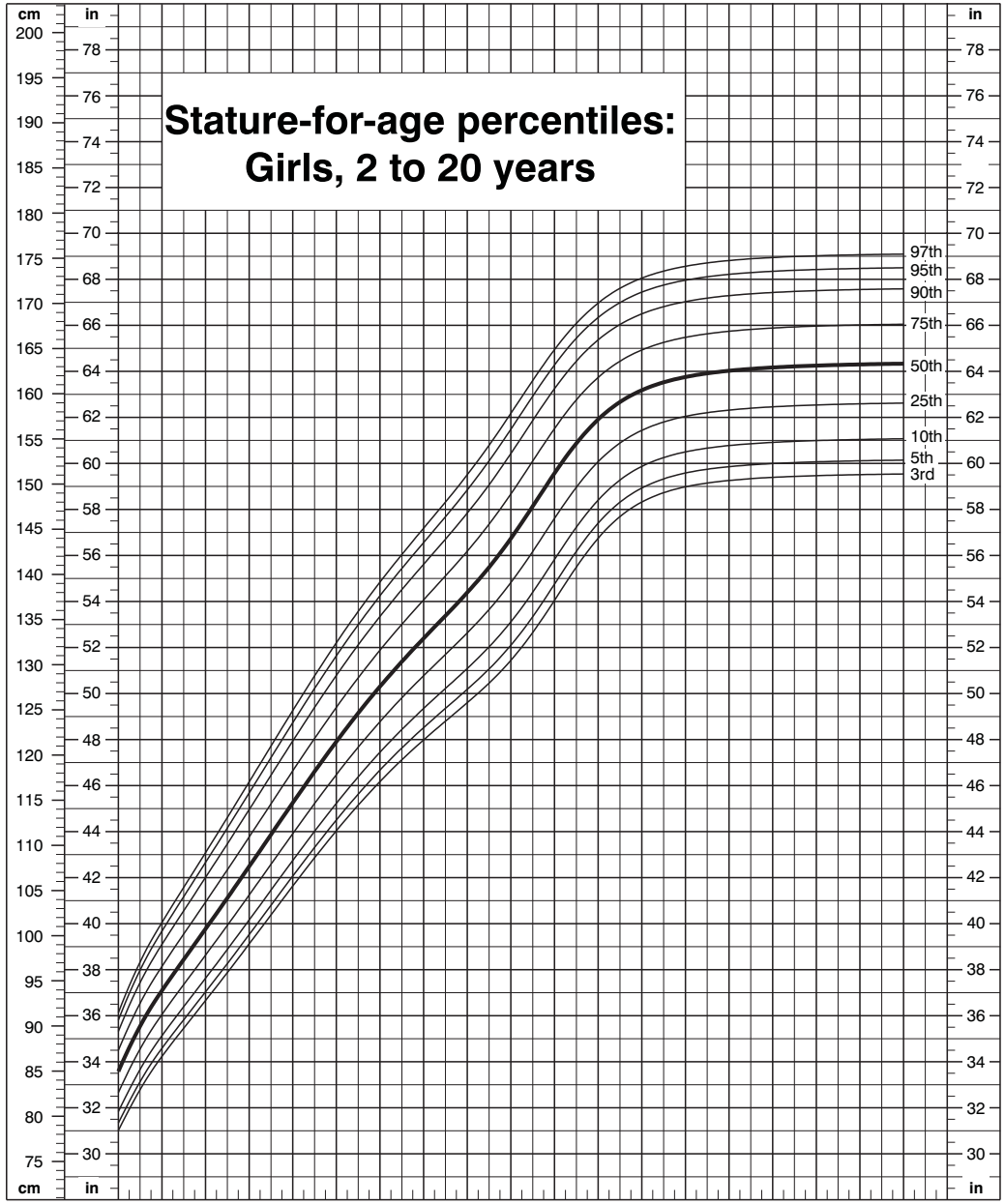
Research Seminar in Statistics and Mathematics at WU Wien
March 16, 2022

Funding for this research was provided, in part, by the Advancing Healthier Wisconsin Research and Education Program under awards 9520277 and 9520364.

Outline of Material

- Part A: Introduction to BART
  - Motivating Example: Growth Charts
  - Bayesian Additive Regression Trees (BART)
  - Heteroskedastic Bayesian Additive Regression Trees (HBART)
  - The **BART** package and other BART software
  - The BART prior
  - Friedman's partial dependence function
  - Returning to growth chart example
  - Posterior MCMC
- Part B: BART computational considerations
  - Installing the **BART** R package
  - The **Rcpp** R package
  - The **BART** package and other BART software
  - A brief overview of multi-processing/-threading
  - Multi-threading with the **BART** R package
  - Missing imputation and multiple imputation
  - Calling BART R functions and `predict`
  - Creating a BART executable with C++ sans R
- Conclusions
- Q & A

CDC Growth Charts: United States



Motivating Example: Growth Charts

- The US Centers for Disease Control and Prevention (CDC) as well as the World Health Organization have developed growth charts for childhood development: height by age, weight by age, body mass index by age and weight by height
- Here we will focus on height, y_t, by age in months, $t = 24, \ldots, 215$ (2 to 17 years old)
- The CDC uses the LMS method via natural cubic splines (Cole and Green 1992 *Statistics in Medicine*)
- Three parameters estimated by penalized maximum likelihood the Box-Cox power transformation, L_t; the mean, M_t; and the coefficient of variation, S_t

$$z_t = \left\{ \begin{array}{ll} \frac{-1+(y_t/M_t)^{L_t}}{L_t S_t} & L_t \neq 0 \\ \frac{\log(y_t/M_t)}{S_t} & L_t = 0 \end{array} \right\} \sim \mathrm{N}(0, 1)$$

- But, this only uses part of the data: just males or just females
- What if we wanted to use all of the data?
- Or include more information like weight or race/ethnicity?

What is Machine Learning Regression?

- Machine Learning Regression (MLR) is within the paradigm of Artificial (or Computational) Intelligence
- MLR is extensible, but for the moment consider the general regression case of a continuous outcome with Normal errors

$$y_i = \mu + f(x_i) + \epsilon_i \qquad \text{where } \epsilon_i \overset{\text{iid}}{\sim} \mathrm{N}(0, \sigma^2)$$

- f is an unspecified function whose form is to be *learned* from the data and x_i is a vector of covariates for $i = 1, \ldots, N$
- A common extension in MLR

$$y_i = \mu + f(x_i) + \sigma s(x_i)\epsilon_i \qquad \text{where } \epsilon_i \overset{\text{iid}}{\sim} F_\epsilon$$

- And f and s will both be *learned*, but how?
- *Ideally* in a *nonparametric* manner without resorting to *precarious restrictive assumptions*

What is Bayesian Additive Regression Trees?

- a supervised MLR with nice properties: automated learning of the functional relationship and interactions without requiring covariate transformations for continuous, binary, categorical and time-to-event outcomes
- tree-based ensemble predictive model
- Bayesian nonparametric method with robust defaults for the prior parameter settings
- computationally efficient posterior inference via MCMC estimates naturally computed from summaries of the posterior along with the quantification of their uncertainty
- seamless extension to variable selection in high dimensions

Selected BART references with URLs

Overview	Chipman, George and McCulloch 2010 *AOAS* Sparapani, Spanbauer and McCulloch 2021 *JSS*
Survival Analysis	Sparapani, Logan et al. 2016 *Statistics in Medicine* Henderson, Louis et al. 2020 *Biostatistics* Sparapani, Rein et al. 2020 *Biostatistics* Sparapani, Logan et al. 2020 *SMMR* Linero, Basak et al. 2021 *Bayesian Analysis*
Big Data (Big N)	Pratola, Chipman et al. 2014 *JCGS* Entezari, Craiu et al. 2017 *Canadian J of Stat*
Variable Selection (Big P)	Linero 2018 *JASA* Liu, Rockova 2021 *JASA*
Efficient MCMC	Pratola 2016 *Bayesian Analysis*
Nonparametric Theory	Rockova and Saha 2019 *PMLR* Rockova and van der Pas 2020 *AOS*
Heteroskedastic	Pratola, Chipman et al. 2020 *JCGS*
Propensity Scores	Hahn, Murray et al. 2020 *Bayesian Analysis*
Monotonic	Chipman, George et al. 2021 *Bayesian Analysis*

BART software with a `predict` function

		R packages		
Debut	Language	Stable (CRAN)	Development	Multi-threading
2006	C++	**BayesTree**	None	None
2013	Java	**bartMachine**		Java
2014	C++	**dbarts**		forking
2014	C++	MPI BART source code		MPI
2017	C++	**BART** 2.9*	**BART3***	OpenMP/forking
2019	C++	**rbart** 1.0*	**hbart***	OpenMP
2019	C++	None	**mxBART***	OpenMP/forking
2021	C++	None	**mBART***	OpenMP/forking
2021	C++	**nftbart** 1.2*	**nftbart***	OpenMP
		*Descendents of MPI BART		

Development on `github.com` by users `rsparapa` (me),
`cspanbauer` (Charley Spanbauer) and `remcc` (Rob McCulloch)
Special thanks to Rob (**BART**), Matt Pratola for **rbart**
Hugh Chipman, Robert Gramacy, the R Core team,
the Rcpp Core team and so many others in the FOSS community!

BART software features: descendents of MPI BART

Stable Development	**BART** **BART3**	**nftbart** **nftbart**	**rbart** **hbart**	**mBART**
`github.com` user		`rsparapa`		`remcc`
`predict` function	Yes	Yes	Yes	**BART**
heteroskedastic	No	Yes	Yes	No
monotonic	No	No	No	Yes
continuous	Yes	Yes	Yes	Yes
binary/categorical	Yes	No	No	No
right censoring	Yes	Yes	No	No
left censoring	No	Yes	No	No
competing risks	Yes	No	No	No
recurrent events	Yes	No	No	No
sparse prior	Yes	No	No	No
marginal effects	**BART3**	Yes	No	No
missing imputation	Yes	Yes	No	No
advanced tree proposals	No	Yes	Yes	No
nonparametric error	No	Yes	No	No
C++ header-only	**BART3**	No	**hbart**	No

Bayesian Additive Regression Trees (BART)

Chipman, George & McCulloch 2010 *Annals of Applied Stat*

$$y_i = \mu + f(x_i) + \epsilon_i \qquad \epsilon_i \overset{\text{iid}}{\sim} \mathrm{N}\left(0,\ w_i^2\sigma^2\right)$$

$$f \overset{\text{prior}}{\sim} \mathbf{BART}\ (\alpha, \beta, H, \kappa, \mu, \tau)$$

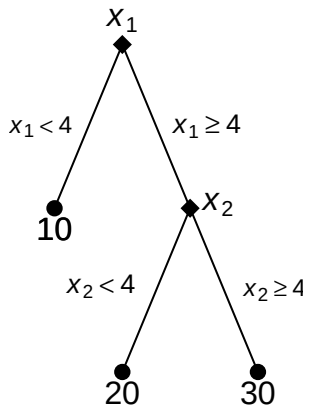
$$f(x_i) \equiv \sum_{h=1}^{H} g(x_i; \mathcal{T}_h, \mathcal{M}_h) \qquad H \in \{50, 200, 500\}$$

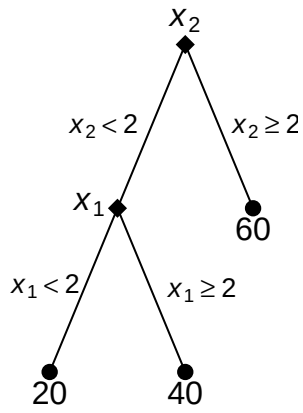
$$\mu_{hl} | \mathcal{T}_h \overset{\text{prior}}{\sim} \mathrm{N}\left(0,\ \frac{\tau^2}{4H\kappa^2}\right) \text{ leaves of } \mathcal{T}_h$$

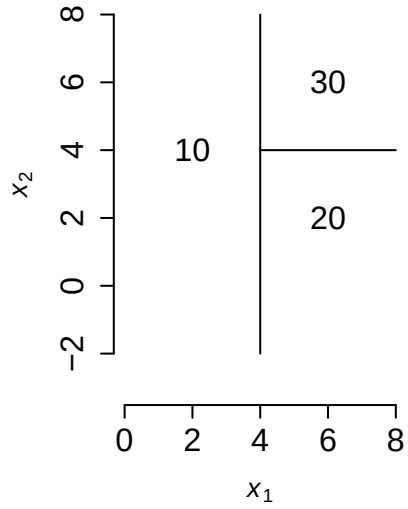
$$\in \mathcal{M}_h$$

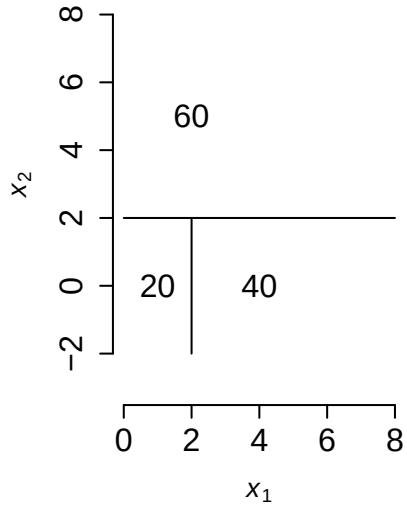
$$\sigma^2 \overset{\text{prior}}{\sim} \lambda\nu\chi^{-2}(\nu)$$

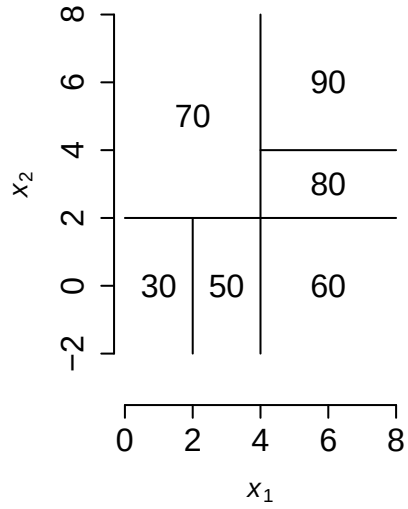
Bayesian Additive Regression Trees (BART)











Heteroskedastic BART (HBART)

Pratola, Chipman, George & McCulloch 2020 *JCGS*

$$y_i = \mu + f(x_i) + s(x_i)\epsilon_i \qquad \epsilon_i \overset{\text{iid}}{\sim} \mathrm{N}\left(0,\ w_i^2\sigma^2\right)$$

$$f \overset{\text{prior}}{\sim} \text{BART}\ (\alpha, \beta, H, \kappa, \mu, \tau)$$

$$s^2 \overset{\text{prior}}{\sim} \textbf{HBART}\ (\tilde{\alpha}, \tilde{\beta}, \widetilde{H}, \tilde{\lambda}, \tilde{\nu})$$

$$s^2(x_i) \equiv \prod_{h=1}^{\widetilde{H}} g(x_i; \widetilde{\mathcal{T}}_h, \widetilde{\mathcal{M}}_h) \qquad \widetilde{H} \approx H/5$$

$$\sigma^2_{hl} | \widetilde{\mathcal{T}}_h \overset{\text{prior}}{\sim} \lambda\nu\chi^{-2}(\nu) \text{ leaves of } \widetilde{\mathcal{T}}_h \qquad \lambda = \tilde{\lambda}^{1/\widetilde{H}}$$

$$\in \widetilde{\mathcal{M}}_h \qquad \nu = 2\left[1 - \left(1 - \frac{2}{\tilde{\nu}}\right)^{1/\widetilde{H}}\right]^{-1}$$

BART, ensembles and prediction error

- mean squared error = bias^2 + variance
- There is a trade-off between the bias and variance
- Consider the spectrum of trade-offs

  Linear regression is on the high bias/low variance end

  Single-tree regression is on the low bias/high variance end
- Ensembles are in the middle: medium bias/medium variance
- BART is in the class of ensemble models which both theoretically, and in practice, have excellent out-of-sample predictive performance

Krogh & Solich 1997 *Physical Review E*
Baldi & Brunak 2001 "Bioinformatics: machine learning approach"
Kuhn & Johnson 2013 "Applied Predictive Modeling"

Binary trees and Bayesian Additive Regression Trees

- BART relies on an ensemble of $\boldsymbol{H}$ binary trees which are a type of a directed acyclic graph
- We exploit the wooden tree metaphor to its fullest except binary trees grow downward by tradition
- Each of these trees grows down starting out as a root node
- The root node is generally a branch decision rule, but it doesn't have to be; occasionally, there are trees in the ensemble which are only a root terminal node consisting of a single leaf output value
- If the root is a branch decision rule, then it spawns a left and a right node which each can be either a branch decision rule or a terminal leaf value and so on
- In binary tree, $\mathcal{T}$, there are $\boldsymbol{C}$ nodes which are made of $\boldsymbol{B}$ branches and $\boldsymbol{L}$ leaves: $\boldsymbol{C = B + L}$
- There is an algebraic relationship between the number of branches and leaves which we express as $\boldsymbol{B = L - 1}$.

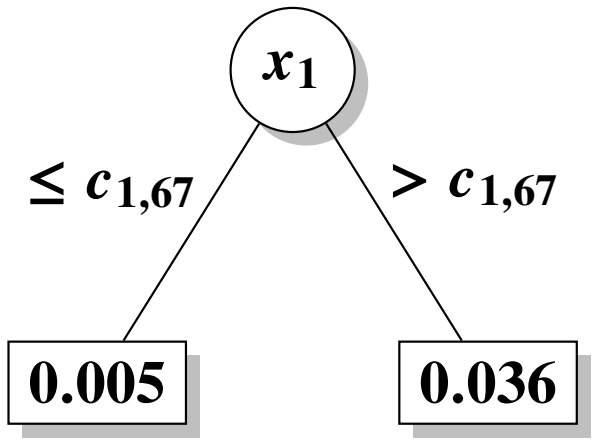
The BART R package

- to facilitate the `predict` function, BART fits can be stored as R objects to be reloaded later
- the ensemble of trees is encoded in an ASCII string which is returned in the `treedraws$trees` list item
- This string can be read by R
- Encoded with C/C++ indexing starting with 0 is used rather than R object indexing starting with 1
- Since the `predict` function calls C/C++ code for speed

The BART R package and trees

Sparapani, Spanbauer and McCulloch 2021
Journal of Statistical Software

```
R> write(post$treedraws$trees, "trees.txt")
R> tc <- textConnection(post$treedraws$tree)
R> trees <- read.table(file=tc, fill=TRUE, row.names=NULL,
+    col.names=c("node", "var", "cut", "leaf"))
R> close(tc)
R> head(trees)
  node var cut          leaf
1 1000 200   1            NA
2    3  NA  NA            NA
3    1   0  66 -0.001032108
4    2   0   0  0.004806880
5    3   0   0  0.035709372
6    3  NA  NA            NA
```



The BART R package and trees

- The string is encoded as follows
- The first line is an exception which has the number of MCMC samples, M, in the field `node`; the number of trees, H, in the field `var`; and the number of variables, P, in the field `cut`
- For the rest of the file, the field `node` is used for the number of nodes in the tree when all other fields are `NA`; or for a specific node when the other fields are present
- The nodes are numbered in relation to the tree's tier level, $t(n) = \lfloor \log_2 n \rfloor$ or `t=floor(log2(node))`

Tier							
0			1				
1		2				3	
2	4		5		6		7
⋮							
t		2^t		…	$2^{t+1}-1$		

The BART R package and trees

- The `var` field is the variable in the branch decision rule which is encoded $\mathbf{0}, \ldots, \boldsymbol{P-1}$ as a C/C++ array index (rather than an R index)
- Similarly, the `cut` field is the cut-point of the variable in the branch decision rule which is encoded $\mathbf{0}, \ldots, \boldsymbol{c_j - 1}$ for variable $\boldsymbol{j}$; note that the cut-points are returned in the `treedraws$cutpoints` list item
- The terminal leaf output value is contained in the field `leaf`
- It is not immediately obvious which nodes are branches vs. leaves since, at first, it would appear that the `leaf` field is given for both branches and leaves
- Leaves are always associated with `var=0` and `cut=0`; however, note that this is also a valid branch variable/cut-point since these are C/C++ indices

The BART R package and trees

- The key to discriminating between branches and leaves is via the algebraic relationship between a branch, $\boldsymbol{n}$, at tree tier $\boldsymbol{t(n)}$ leading to its left, $\boldsymbol{l = 2n}$, and right, $\boldsymbol{r = 2n + 1}$, nodes at tier $\boldsymbol{t(n) + 1}$
- for each node, besides root, you can determine from which branch it arose and those nodes that are not a branch (since they have no leaves) are necessarily leaves

The BART prior

- The BART prior specifies a flexible class of unknown functions, f, from which we can gather randomly generated fits to the given data via the posterior
- We define f as returning a scalar value, but BART extensions which return multivariate values are conceivable
- Let function $g(x; \mathcal{T}, \mathcal{M})$ assign a value based on the input x
- The binary tree $\mathcal{T}$ is represented by a collection of C four-tuples $(n, \psi_n; j, k)$: n for node number;
  $\psi_n = 1$ for a branch and 0 for a leaf;
  and, if a leaf, j for covariate x_j with k for the cut-point c_{jk}
- Suppose the collection of branches is denoted by
  $\mathcal{B} = \{n : \psi_n = 1\}$
- The branch decision rules are of the form $x_j \leq c_{jk}$ which means branch left and $x_j > c_{jk}$, branch right; or terminal leaves where it stops. $\mathcal{M}$ represents leaves and is a set of ordered pairs,
  (n, μ_n): $n \in \mathcal{L}$ where $\mathcal{L}$ is the set of leaves
  ($\mathcal{L}$ is the complement of $\mathcal{B}$) and μ_n for the outcome value

The BART prior

- The function, $f(x)$, is a sum of H trees:

$$f(x) = \sum_{h=1}^{H} g(x; \mathcal{T}_h, \mathcal{M}_h)$$

where H is "large", let's say, 50, **200** or 500

- For a continuous outcome, y_i, we have the following BART regression on the vector of covariates, x_i:

$$y_i = \mu + f(x_i) + \epsilon_i \text{ where } \epsilon_i \overset{\text{iid}}{\sim} \mathrm{N}\left(0,\ w_i^2 \sigma^2\right)$$

with i indexing subjects $i = 1, \ldots, N$

- The BART priors for the unknown random function, f, and the error variance, σ^2, are as follows

$$f \overset{\text{prior}}{\sim} \text{BART}\ (\alpha, \beta, H, \mu, \kappa, \tau) \qquad \sigma^2 \overset{\text{prior}}{\sim} \nu\lambda\chi^{-2}\ (\nu)$$

where H is the number of trees, μ is a known constant which centers y and the rest of the parameters will be explained later in this section (for brevity, we often use $f \overset{\text{prior}}{\sim} \text{BART}$)

The BART prior

- The w_i are known standard deviation weight multiples which you can supply with the argument `w` that is only available for continuous outcomes, hence, the weighted BART name; the unit weight vector is the default
- The centering parameter, μ, can be specified via the `fmean` argument where the default is taken to be $\bar{y}$
- x_i: matrices or data frames can be supplied
- unlike matrices, data frames can contain categorical factors when `x.train` is a data frame
- Factors with multiple levels are transformed into dummy variables with each level as their own binary indicator; factors with only two levels are a binary indicator with a single dummy variable

The BART prior

- BART is a Bayesian nonparametric prior
- we represent the BART prior in terms of the collection of all trees, $\mathcal{T}$; collection of all leaves, $\boldsymbol{M}$; and the error variance, $\boldsymbol{\sigma^2}$, as: $\left[\mathcal{T}, \boldsymbol{M}, \sigma^2\right] = \left[\sigma^2\right]\left[\mathcal{T}, \boldsymbol{M}\right] = \left[\sigma^2\right]\left[\mathcal{T}\right]\left[\boldsymbol{M}|\mathcal{T}\right]$
- the individual trees themselves are independent: $\left[\mathcal{T}, \boldsymbol{M}\right] = \prod_h \left[\mathcal{T}_h\right]\left[\boldsymbol{M}_h|\mathcal{T}_h\right]$ where $\left[\mathcal{T}_h\right]$ is the prior for the hth tree and $\left[\boldsymbol{M}_h|\mathcal{T}_h\right]$ is the collection of leaves for the hth tree
- the collection of leaves for the hth tree are independent: $\left[\boldsymbol{M}_h|\mathcal{T}_h\right] = \prod_n \left[\mu_{hn}|\mathcal{T}_h\right]$ where n indexes the leaf nodes

The BART prior

- The tree prior: $[\mathcal{T}_h]$. There are three prior components of $\mathcal{T}_h$ which govern whether the tree branches grow or are pruned
- The first tree prior regularizes the probability of a branch at leaf node n in tree tier $t(n) = \lfloor \log_2 n \rfloor$ as follows.

$$\psi_n \overset{\text{prior}}{\sim} \mathrm{B}(p(t(n))) \text{ where } p(t(n)) = \alpha(t(n)+1)^{-\beta} \qquad (1)$$

  $\psi_n = 1$ represents a branch while $\psi_n = 0$ is a leaf
  $0 < \alpha < 1$ and $\beta \geq 0$
- You can specify these prior parameters with arguments, but the following defaults are recommended: α is set by the parameter `base=0.95` and β by `power=2`
- The expected number of branches (leaves) is 1 (2) with probability $\mathrm{P}[\psi_1 = 1, \psi_2 = \psi_3 = 0] = p(0)q(1)^2 \approx 0.55$
- Or 2 (3) with $2\mathrm{P}[\psi_1 = \psi_2 = 1, \psi_3 = \psi_4 = \psi_5 = 0] = 2p(0)p(1)q(1)q(2)^2 \approx 0.27$ (doubled due to symmetry)
- Trees with only 1 or 2 branches (2 or 3 leaves) would dominate with a probability of about **0.82**

BART and Bayesian nonparametric theory

- frequentist theoretical justification for BART's performance: asymptotically consistent with a near optimal learning rate
- the BART posterior distribution concentrates around the truth at a near optimal minimax rate
- the default BART Branching penalty is near optimal:
  $\psi_n \overset{\text{prior}}{\sim} \mathbf{B}\left(\alpha(1+t(n))^{-\beta}\right)$ **where** $t(n) = 0, \ldots$
- the optimal BART Branching penalty is now shown to be:
  $\psi_n \overset{\text{prior}}{\sim} \mathbf{B}\left(\gamma^{t(n)}\right)$ **where** $0 < \gamma < 0.5$

Branches (Leaves)	0 (1)	1 (2)	2 (3)	3+ (4+)
Prior probability	0.00	$(1-\gamma)^2$	$2\gamma(1-\gamma)(1-\gamma^2)^2$	$\ldots$
$\gamma = 0.25$	0.00	0.56	0.33	0.11
$\alpha = 0.95, \beta = 2$	0.05	0.55	0.27	0.13

Chipman, George & McCulloch 1998 *JASA*
Rockova & Saha 2018 *PMLR*
Rockova & van der Pas 2020 *Annals of Statistics*

The BART prior

- The leaf prior: $[\mu_{hn}|\mathcal{T}_h]$
- Given a tree, $\mathcal{T}_h$, there is a prior on its leaf values, $\mu_{hn}|\mathcal{T}_h$ and we denote the collection of all leaves in $\mathcal{T}_h$ by $\mathcal{M}_h = \{(n, \mu_{hn}) : n \in \mathcal{L}_h\}$
- Suppose that $y \in [y_{\min}, y_{\max}]$ where $y_{\min}$ and $y_{\max}$ might be elicited as $\mathbf{0.025}$ and $\mathbf{0.975}$ quantiles otherwise, the observed min and max are used (the default)
- Denote $[\tilde{\mu}_1, \ldots, \tilde{\mu}_H]$ as the leaf output values from each tree corresponding to the vector of covariates, x
- If $\tilde{\mu}_h|\mathcal{T}_h \overset{\text{iid}}{\sim} \mathrm{N}\left(0, \sigma_\mu^2\right)$, then the model estimate is $\hat{y} = \mathrm{E}[y|x] = \mu + \sum_h \tilde{\mu}_h$ where $\hat{y} \sim \mathrm{N}\left(\mu, H\sigma_\mu^2\right)$
- Solve for σ_μ: $y_{\min} = \mu - \kappa\sqrt{H}\sigma_\mu$ and $y_{\max} = \mu + \kappa\sqrt{H}\sigma_\mu$ $\sigma_\mu = \frac{y_{\max} - y_{\min}}{2\kappa\sqrt{H}}$
- Therefore, we arrive at $\mu_{hn} \overset{\text{prior}}{\sim} \mathrm{N}\left(0, \frac{\tau^2}{4H\kappa^2}\right)$ where $\tau = y_{\max} - y_{\min}$

The BART prior

- The parameter κ calibrates this prior as follows
- The default value, $\kappa = 2$, corresponds to $\hat{y}$ falling within the extrema with approximately 0.95 probability
- Alternative choices of κ can be supplied via the `k` argument
- We have found that values of $\kappa \in [1, 3]$ generally yield good results
- Note that κ is a potential candidate parameter for choice via cross-validation

The BART prior

- We fix the number of trees at $\boldsymbol{H}$ which corresponds to the argument `ntree`
- The default number of trees is 200 for continuous outcomes; but for computational convenience, 50 is also a reasonable choice which is the default for all other outcomes
- cross-validation could be considered

The BART prior

- The number of cut-points is provided by the argument `numcut` and the default is 100
- The default number of cut-points is achieved for continuous covariates
- For continuous covariates, the cut-points are uniformly distributed by default, or generated via uniform quantiles if the argument `usequants=TRUE` is provided
- By default, discrete covariates which have fewer than 100 values will necessarily have fewer cut-points
- However, if you want a single discrete covariate to be represented by a group of binary dummy variables, one for each category, then pass the variable as a factor within a data frame

The BART prior

- Next, there is a prior dictating the choice of a splitting variable j conditional on a branch event ψ_n which defaults to uniform probability $s_j = P^{-1}$ where P is the number of covariates
- Given a branch event, ψ_n, and a variable chosen, x_j, the last tree prior selects a cut point, c_{jk}, within the range of observed values for x_j; this prior is uniform

The BART error variance prior: $[\sigma^2]$

- The prior for σ^2 is the conjugate scaled inverse Chi-square distribution, i.e., $\lambda\nu\chi^{-2}(\nu)$
- We recommend that the degrees of freedom, ν, be from 3 to 10 and the default is 3 (can be over-ridden by the argument `sigdf`)
- The λ parameter can be specified by the `lambda` argument (defaults to `NA`)
- If `lambda` is unspecified, then we determine a reasonable value for λ based on an estimate, $\widehat{\sigma}$, (which can be specified by the argument `sigest` and defaults to `NA`)

The BART error variance prior

- If `sigest` is unspecified, the default value of `sigest` is determined via linear regression or the sample standard deviation: if $P < N$, then $y_i \sim \mathrm{N}\left(x_i'\widehat{\beta},\ \widehat{\sigma}^2\right)$; otherwise, $\widehat{\sigma} = s_y$
- Now we solve for λ such that $\mathrm{P}\left[\sigma^2 \leq \widehat{\sigma}^2\right] = q$
- This quantity, q, can be specified by the argument `sigquant` and the default is 0.9 whereas we also recommend considering 0.75 and 0.99
- Note that the pair (ν, q) are potential candidate parameters for choice via cross-validation.

Friedman's partial dependence function and Marginal Effects of Independent Variables

Friedman 2001 *Annals of Statistics*

$f(x) = f(x_S, x_C)$ a complex function like BART where $x = [x_S, x_C]$

$$f_S(x_S) = \mathrm{E}_{x_C}\left[f(x_S, x_C)|x_S\right]$$

$$\approx N^{-1}\sum_i f(x_S, x_{iC}) \quad \text{partial dependence function}$$

$$f_{Sm}(x_S) \equiv N^{-1}\sum_i f_m(x_S, x_{iC})$$

$$\hat{f}_S(x_S) \equiv M^{-1}\sum_m f_m(x_S)$$

Friedman's partial dependence function and Marginal Effects of Dependent Variables

- Consider our growth chart for height example
- Age and weight obviously co-vary
- t for age, u for sex, v for race/ethnicity and w for weight
  $f_{t,u}(t,u) = \mathbf{E}_{v,w}\left[f(t,u,v,w)|t,u\right]$ assuming Independence
- To do this right, first consider the likely strong relationship between age, gender and weight among children
  $\mathbf{E}\left[w|t,u\right] = \tilde{w} = \tilde{f}(t,u)$
- So estimate the relationship with a BART model
  $w_i = \tilde{f}(t_i,u_i) + \tilde{\epsilon}_i$ **where** $\tilde{f} \overset{\text{prior}}{\sim}$ **BART**
- A marginal effect more appropriate for dependent variables

$$f_{t,u}(t,u) = \mathbf{E}_v\left[f(t,u,v,\tilde{w})|t,u,\tilde{w} = \mathbf{E}[w|t,u]\right] \quad \text{assuming}$$
$$= \mathbf{E}_v\left[f(t,u,v,\tilde{f}(t,u))|t,u\right] \quad \text{Dependence}$$

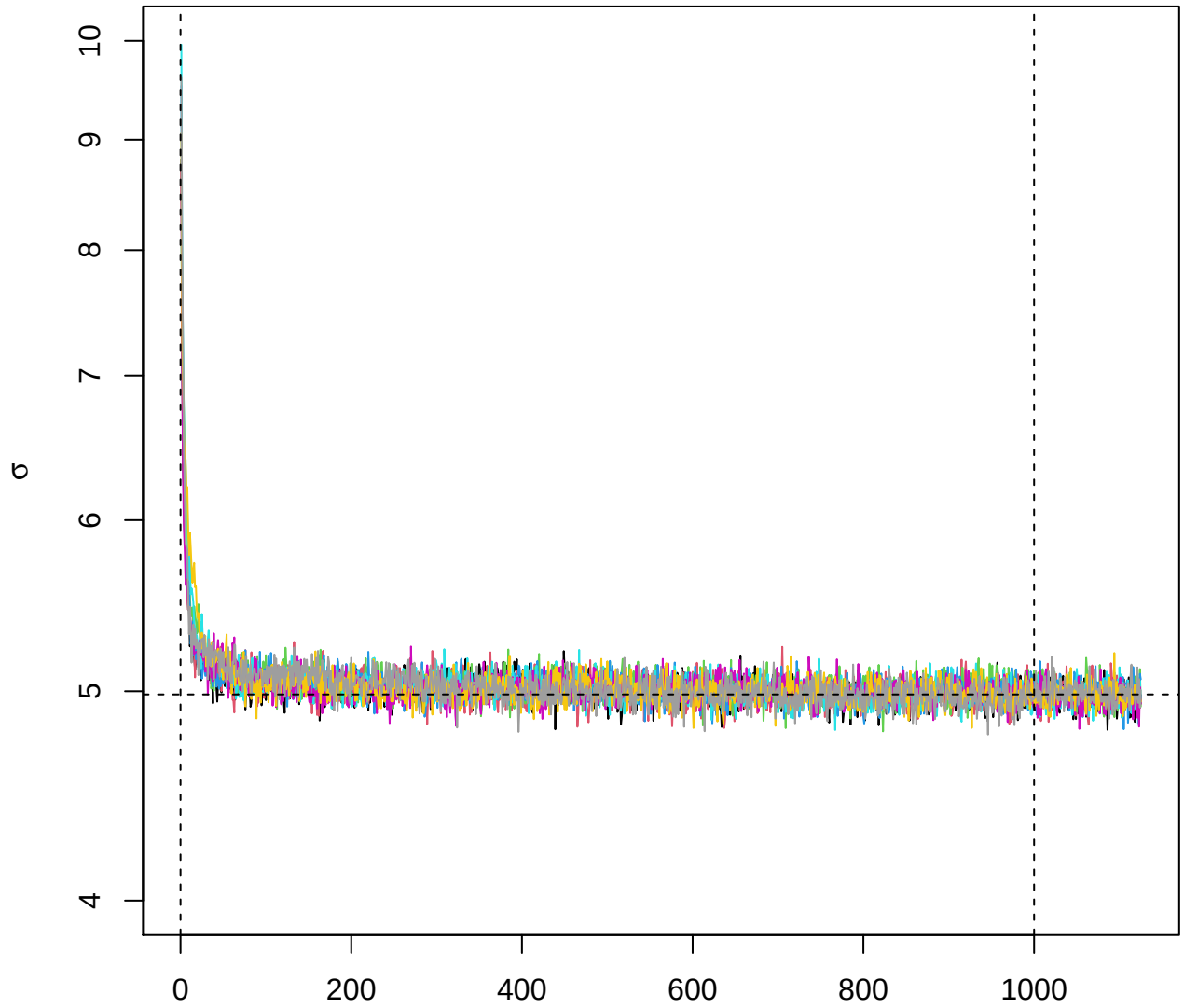
Returning to the real data example

- The CDC mainly used the US National Health and Nutrition Examination Survey (NHANES): waves I-III $\boldsymbol{n = 12677}$
- For simplicity, I used NHANES 1999-2000 annual/continuous
- The data set is in the BART3 package: `bmx`
  and see the `height3.R` example in `demo`
- 2-17 years (fractional age for months)
- each child only measured once
- height (cm) and weight (kg) collected
- Check MCMC convergence with $\boldsymbol{\max \widehat{R} < 1.1}$ for $\boldsymbol{\sigma}$:
  Vehtari, Gelman et al. 2021 *Bayesian Analysis*

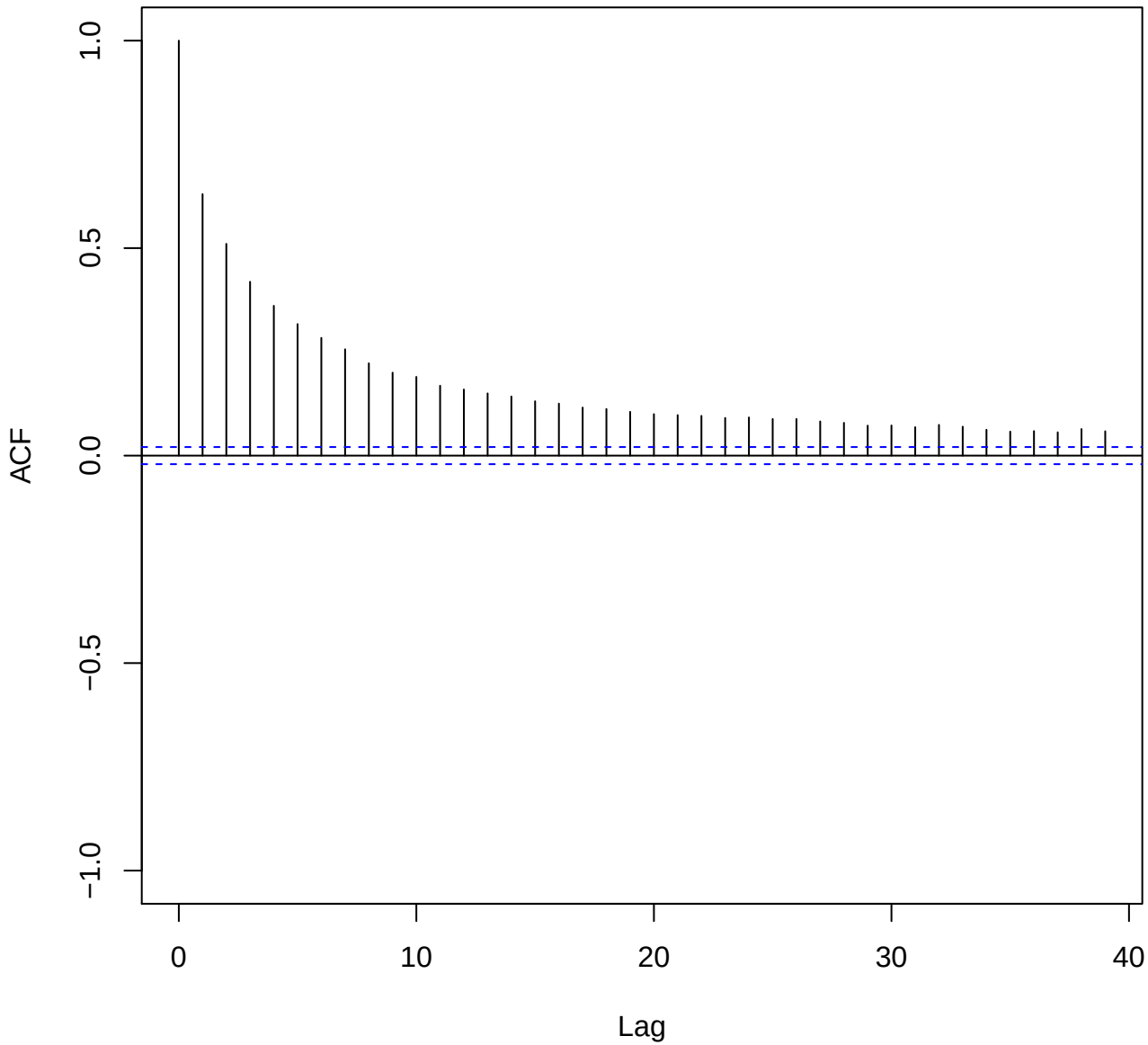
	n	%
Total	3435	
Males	1768	51.5
Females	1667	48.5
White	800	23.3
Black	1035	30.1
Hispanic	1600	46.6

MCMC Convergence `fit$sigma`: $\max \widehat{R} = 1.08$

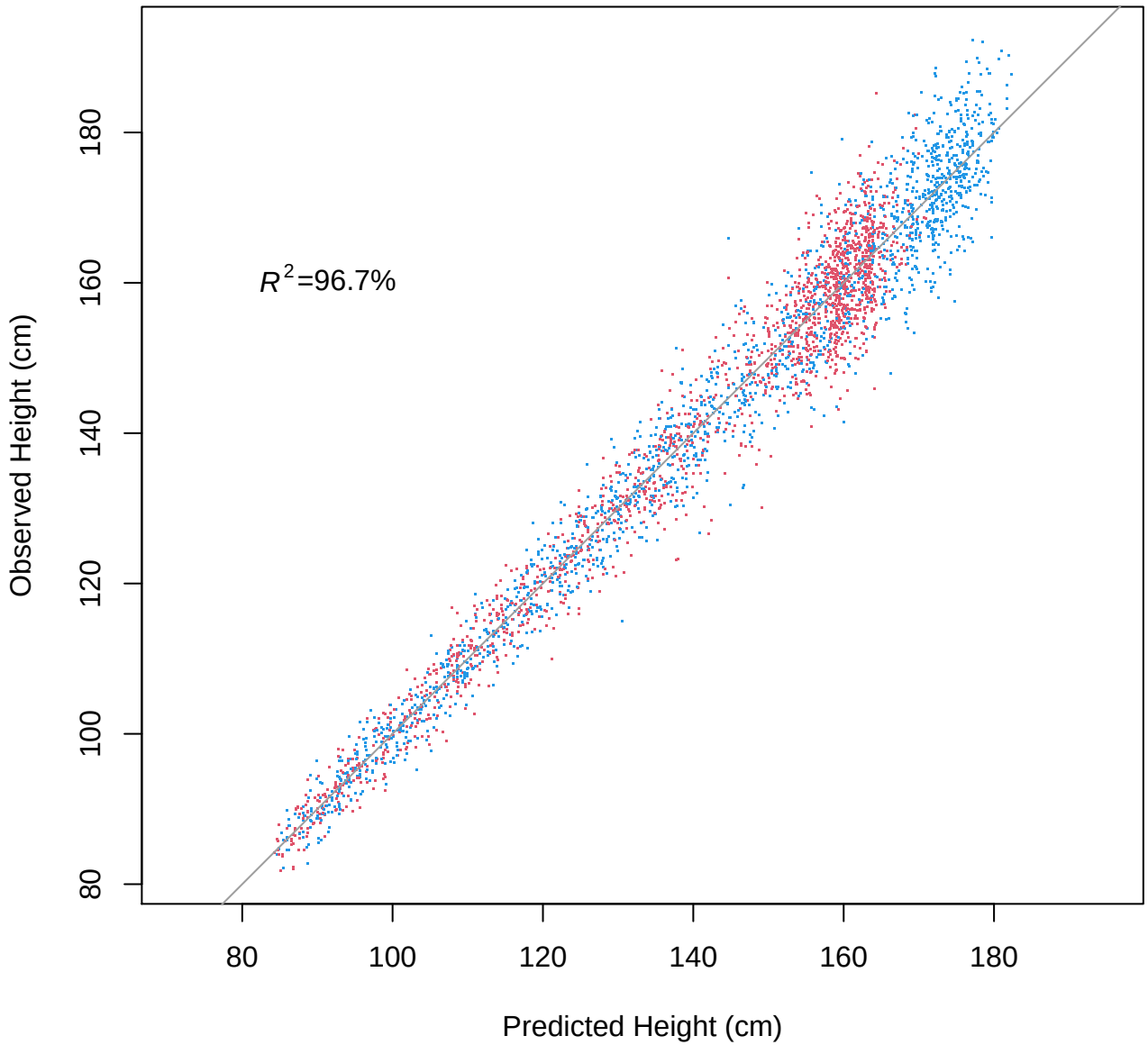
Burn-in 1000, Thinning 10, Chains 8, Posterior 1000



MCMC Convergence `fit$sigma`: Auto-correlation

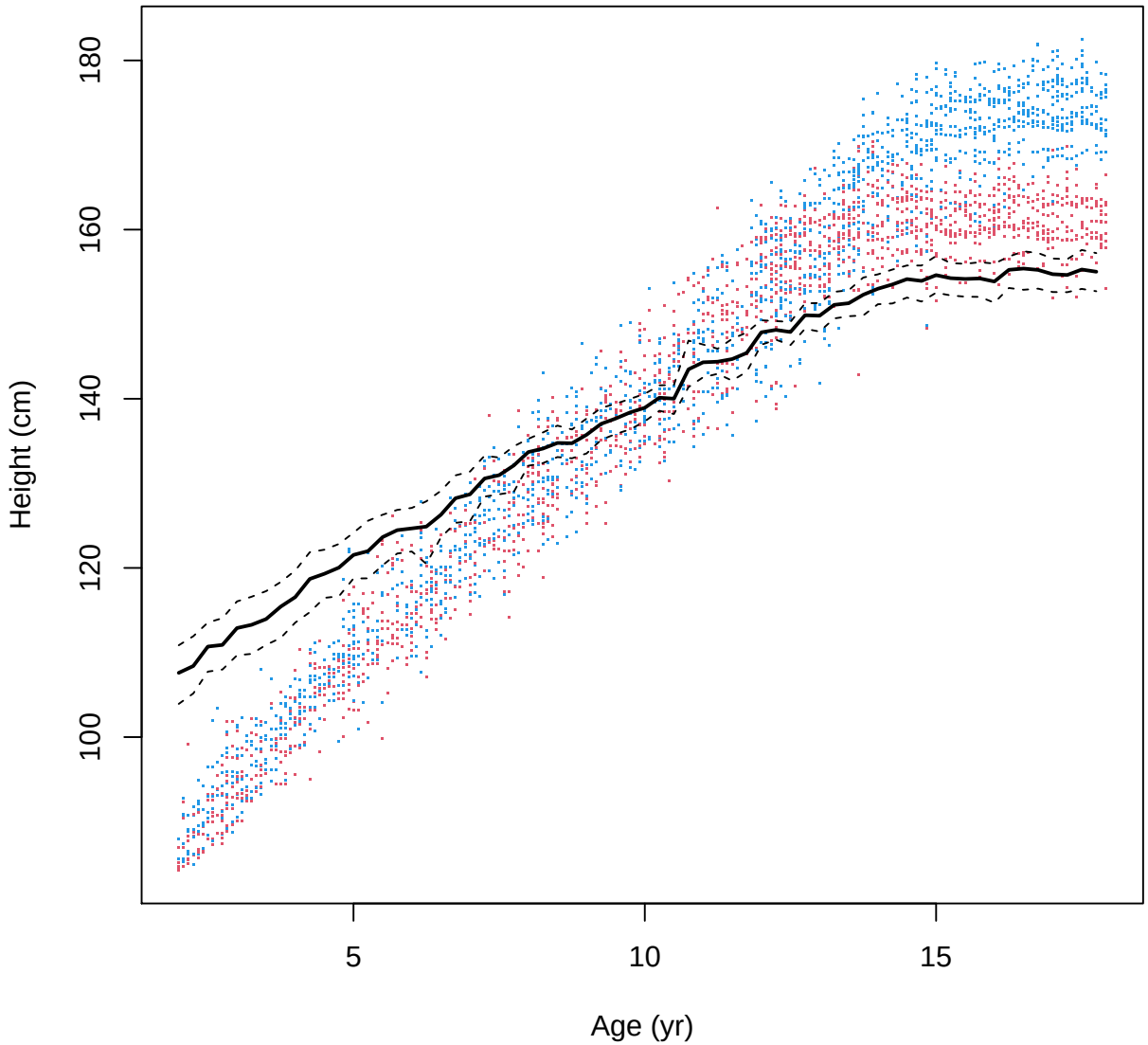


BART fit: M vs. F

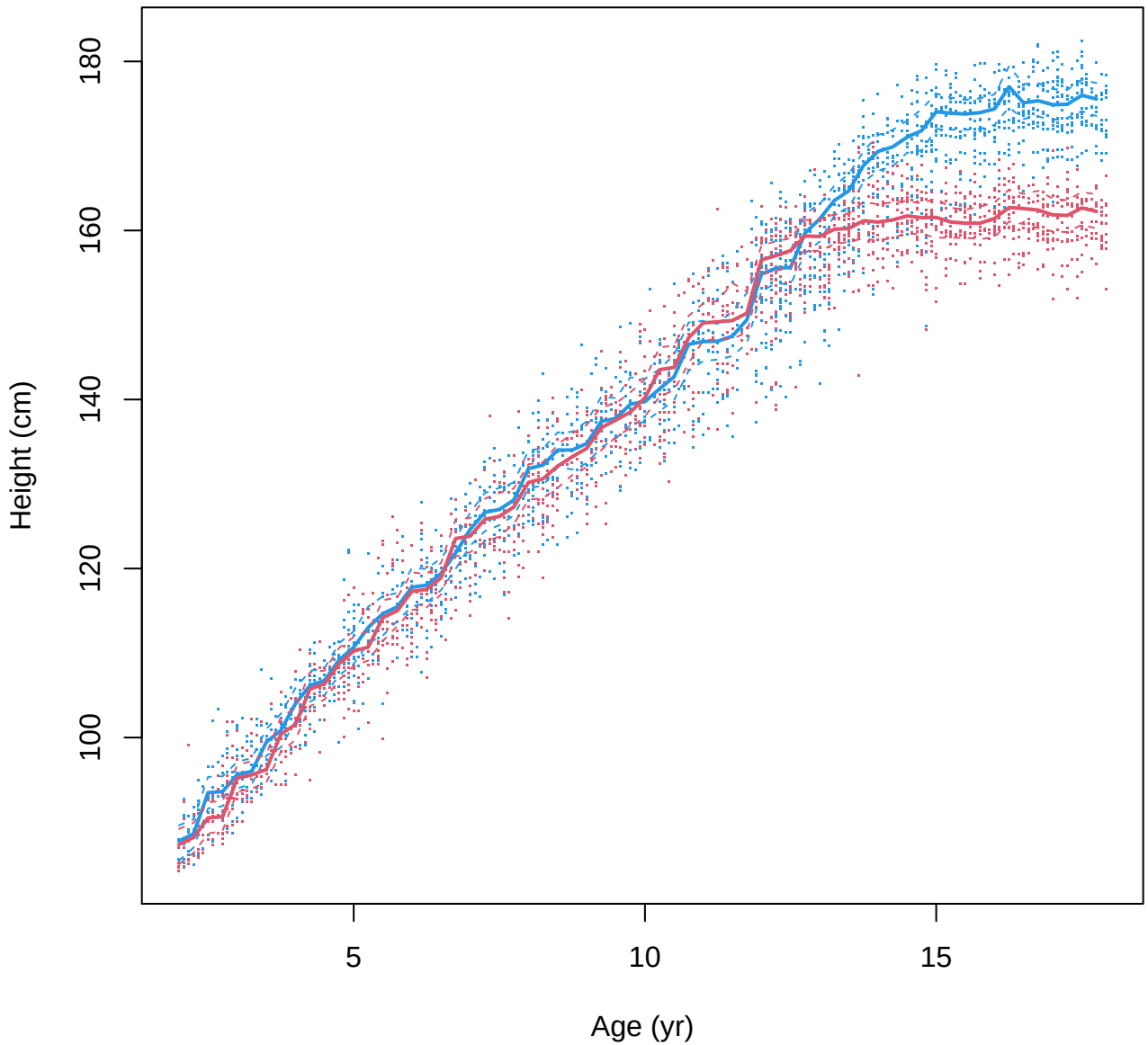


Marginal effect of age assuming independence

$H = 200$, `numcut` = 100

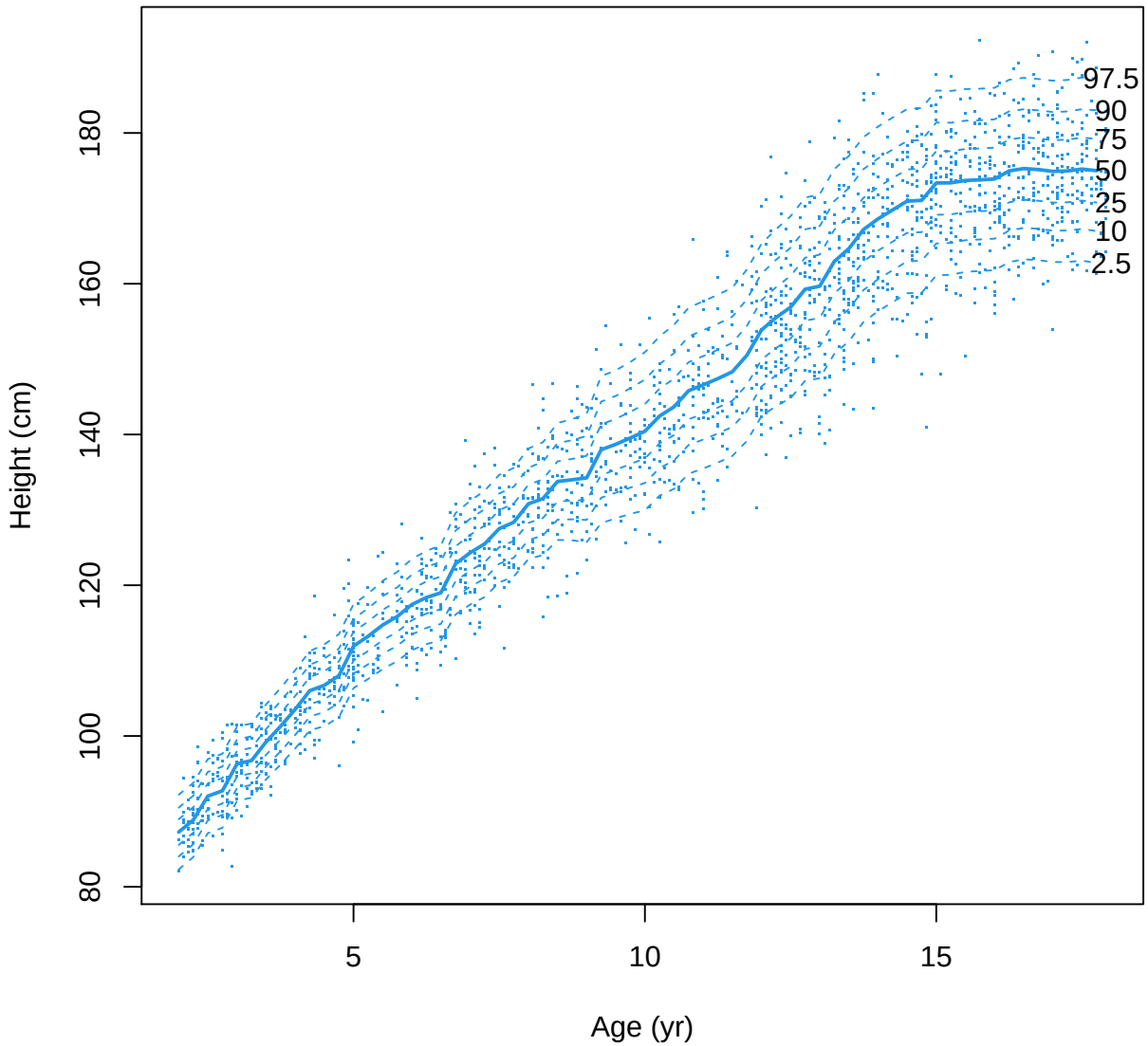


Marginal effect of age: BART predictions for M and F

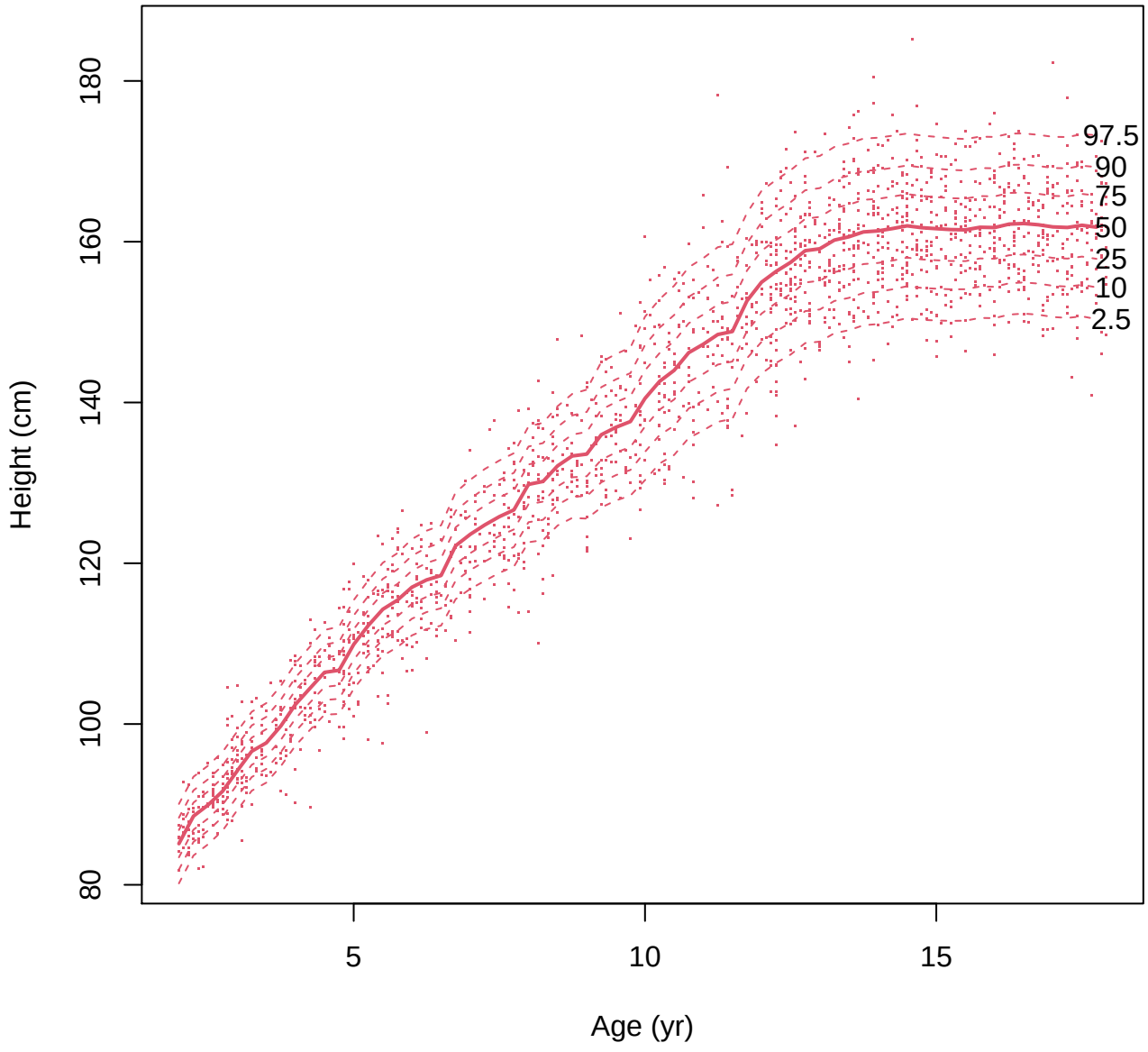


Marginal effect of age: HBART predictions for M

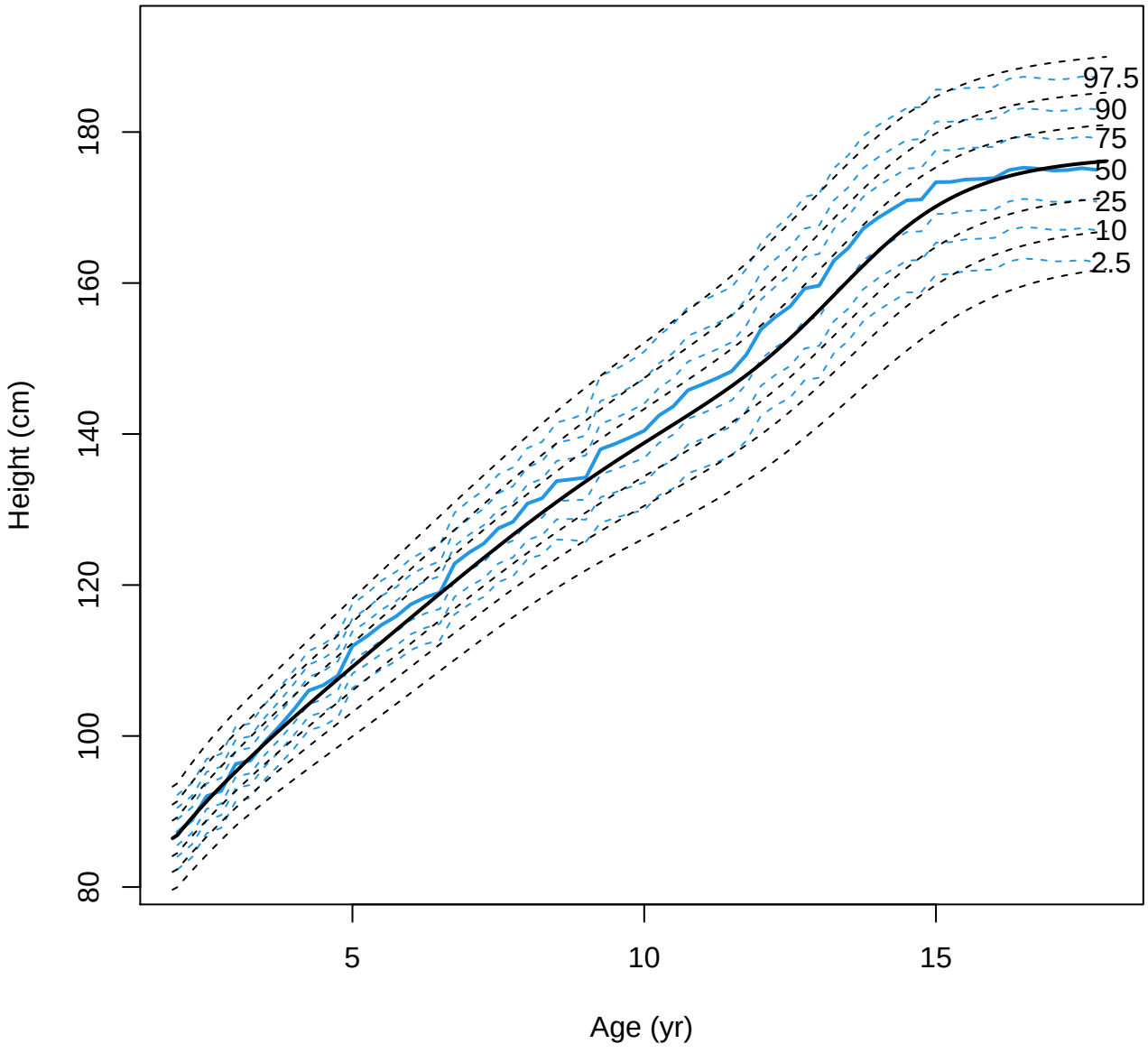
$$H = 300, \widetilde{H} = 60, \texttt{numcut} = 200$$



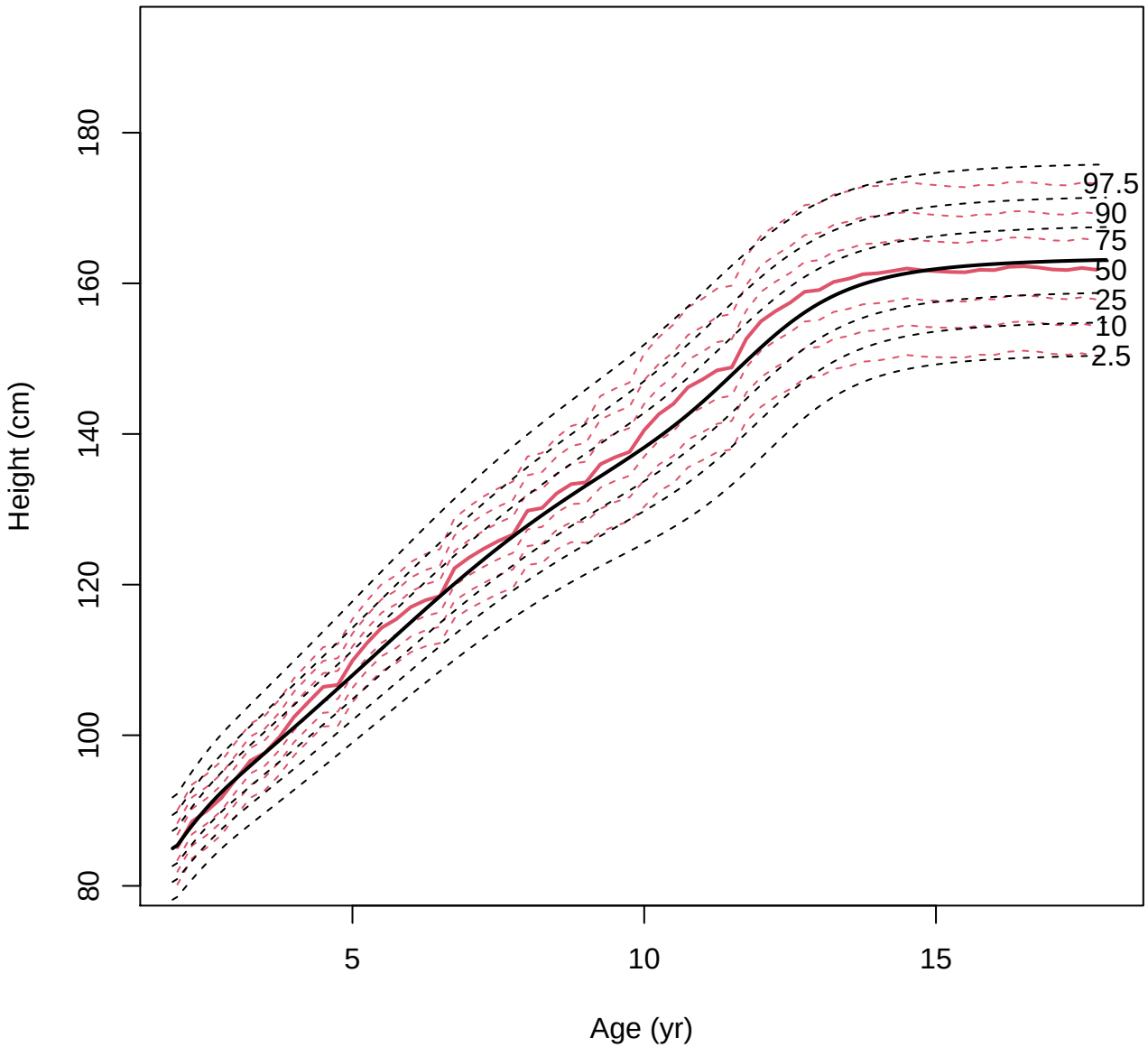
Marginal effect of age: HBART predictions for F



Marginal effect of age: HBART vs. CDC for M



Marginal effect of age: HBART vs. CDC for F



Conclusion

- These slides provide you with an introduction to BART
- BART is a flexible Bayesian nonparametric method for MLR
- Supporting continuous, categorical and time-to-event outcomes
- As an ensemble it has excellent out-of-sample performance
- The tree branching penalty prevents over-fitting
- The BART and HBART priors have convenient default settings
- Using HBART for growth charts is an interesting application
- The BART/BART3 R packages are user-friendly and dependable
- The rbart/hbart R packages are maturing

Posterior computation for BART

- In order to generate samples from the posterior for f, we sample the structure of all the trees $\mathcal{T}_h$, for $h = 1, \ldots, H$; the values of all leaves μ_{hn} for $n \in \mathcal{L}_h$ within tree h; and, when appropriate, the error variance σ^2
- Additionally, with the sparsity prior, there are samples of the vector of splitting variable selection probabilities $[s_1, \ldots, s_P]$ and, when the sparsity parameter is random, samples of θ
- The leaf and variance parameters are sampled from the posterior using Gibbs sampling
- Since the priors on these parameters are conjugate, the Gibbs conditionals are specified analytically
- For the leaves, each μ_{hn} is drawn from a Normal conditional density
- The error variance, σ^2, is drawn from a scaled inverse Chi-square conditional

Posterior computation for BART

- Drawing a tree from the posterior requires a Metropolis-within-Gibbs sampling scheme, i.e., a Metropolis-Hastings (MH) step within Gibbs sampling
- For single-tree models, four different proposal mechanisms are defined in ChipGeor98
- The complementary BIRTH/DEATH proposals are essential (the two other proposals are CHANGE and SWAP which are optional)
- For programming simplicity, the `BART` package only implements the BIRTH and DEATH proposals each with equal probability
- BIRTH selects a leaf and turns it into a branch, i.e., selects a new variable and cut-point with two leaves "born" as its descendants
- DEATH selects a branch leading to two terminal leaves and "kills" the branch by replacing it with a single leaf

Posterior computation for BART

- we present the acceptance probability for a BIRTH proposal
- (a DEATH proposal is the reversible inverse of a BIRTH proposal)
- The algorithm assumes a fixed discrete set of possible split values for each x_j
- the leaf values, μ_{hn}, are integrated over so that our search in tree space is over a large, but discrete, set of possibilities
- At the mth MCMC step, let $\mathcal{T}^m$ denote the current state for the hth tree and $\mathcal{T}^*$ denotes the proposed hth tree (subscript h is suppressed for convenience). $\mathcal{T}^*$ are identical $\mathcal{T}^m$ except that one terminal leaf of $\mathcal{T}^m$ is replaced by a branch of $\mathcal{T}^*$ with two terminal leaves

Posterior computation for BART

- The proposed tree is accepted with the following probability:

$$\pi_{\text{BIRTH}} = \min\left(1, \frac{[\mathcal{T}^*]}{[\mathcal{T}^m]} \frac{[\mathcal{T}^m|\mathcal{T}^*]}{[\mathcal{T}^*|\mathcal{T}^m]}\right)$$

  where $[\mathcal{T}^m]$ and $[\mathcal{T}^*]$ are the posterior probabilities of $\mathcal{T}^m$ and $\mathcal{T}^*$ respectively
- These are the targets of this sampling, each consisting of a likelihood contribution and prior contribution
- Additionally, $[\mathcal{T}^m|\mathcal{T}^*]$ is the probability of proposing $\mathcal{T}^m$ given current state $\mathcal{T}^*$ (a DEATH) and $[\mathcal{T}^*|\mathcal{T}^m]$ is the probability of proposing $\mathcal{T}^*$ given current state $\mathcal{T}^m$ (a BIRTH)

Posterior computation for BART

- First, we describe the likelihood contribution to the posterior
- Let y_n denote the partition of y corresponding to the leaf node n given the tree $\mathcal{T}$
- Because the leaf values are a priori conditionally independent, we have $[y|\mathcal{T}] = \prod_n [y_n|\mathcal{T}]$
- So, for the ratio $\frac{[\mathcal{T}^*]}{[\mathcal{T}^m]}$ after cancellation of terms in the numerator and denominator, we have the likelihood contribution:

$$\frac{[y_L, y_R|\mathcal{T}^*]}{[y_{LR}|\mathcal{T}^m]} = \frac{[y_L|\mathcal{T}^*]\,[y_R|\mathcal{T}^*]}{[y_{LR}|\mathcal{T}^m]}$$

  where y_L is the partition corresponding to the newborn left leaf node; y_R, the partition for the newborn right leaf node; and $y_{LR} = \begin{bmatrix} y_L \\ y_R \end{bmatrix}$
- N.B. the terms in the ratio are the predictive densities of a Normal mean with a known variance and a Normal prior for the mean

Posterior computation for BART

- Similarly, the terms that the prior contributes to the posterior ratio often cancel since there is only one "place" where the trees differ and the prior draws components independently at different "places" of the tree
- Therefore, the prior contribution to $\frac{[\mathcal{T}^*]}{[\mathcal{T}^m]}$ is

$$\frac{\mathbf{P}[\psi_n = 1]\,\mathbf{P}[\psi_l = 0]\,\mathbf{P}[\psi_r = 0]\,s_j}{\mathbf{P}[\psi_n = 0]} = \frac{\alpha(t(n)+1)^{-\beta}\left[1-\alpha(t(n)+2)^{-\beta}\right]^2 s_j}{1-\alpha(t(n)+1)^{-\beta}}$$

where $\mathbf{P}[\psi_n]$ is the branch regularity prior, s_j is the splitting variable selection probability, n is the chosen leaf node in tree $\mathcal{T}^m$, $l = 2n$ is the newborn left leaf node in tree $\mathcal{T}^*$ and $r = 2n+1$ is the newborn right leaf node in tree $\mathcal{T}^*$

Posterior computation for BART

- Finally, the ratio $\frac{[\mathcal{T}^m|\mathcal{T}^*]}{[\mathcal{T}^*|\mathcal{T}^m]}$ is

$$\frac{[\mathbf{DEATH}|\mathcal{T}^*]\,[n|\mathcal{T}^*]}{[\mathbf{BIRTH}|\mathcal{T}^m]\,[n|\mathcal{T}^m]\,s_j}$$

  where $[n|\mathcal{T}]$ is the probability of choosing node n given tree $\mathcal{T}$
- N.B. s_j appears in both the numerator and denominator of the acceptance probability $\pi_{\mathbf{BIRTH}}$, therefore, canceling which is mathematically convenient

Posterior computation for BART: uncertainty intervals

- Suppose that we want to estimate f at some value x whether from the training or testing
- The standard Bayesian estimate is $\hat{f}(x) = M^{-1} \sum_m f_m(x)$
- Construct a Bayesian $(1-\alpha) \times 100\%$ credible interval from the $\alpha/2$ and the $1-\alpha/2$ quantiles of the posterior
- $(f_{(\alpha/2)}(x), f_{(1-\alpha/2)}(x))$
- Construct a Bayesian $(1-\alpha) \times 100\%$ prediction interval
- Generate $\tilde{y}$ from the predictive distribution: $\tilde{y}_m \sim \mathrm{N}\left(f_m(x),\ \sigma_m^2\right)$
- Select the $\alpha/2$ and the $1-\alpha/2$ quantiles of $\tilde{y}$
- $(\tilde{y}_{(\alpha/2)}, \tilde{y}_{(1-\alpha/2)})$