M.Stat. Dissertation

BAYESIAN NONPARAMETRIC GENERALIZATION OF TREE BASED MACHINE LEARNING APPROACHES

March 19, 2021

Arindam Roy Chowdhury*[1] and Dr. Sourabh Bhattacharya †[2]

[1]Indian Statistical Institute, Kolkata
[2]Interdisciplinary Statistical Research Unit, Indian Statistical Institute, Kolkata

*arindam.roy.c1@gmail.com
†sourabh@isical.ac.in, bhsourabh@gmail.com,

1 Introduction

The importance of tree based methods in machine learning research is undeniable. Although Classification and Regression Trees (CART) ((Breiman, Friedman, Olshen & Stone 1984)) have fundamental importance in this regard, ensemble methods that combine a set of tree models, such as boosting (Freund & Schapire 1997 [2]), bagging (Breiman 1996 [1]), random forests (Breiman 2001 [5]), etc., have turned out to be much popular with time. Bayesian versions motivated by these methods, such as Bayesian CART (Chipman, George & McCulloch 1998 [3]), Bayesian Additive Regression Trees (BART) (Chipman, George & McCulloch 2010 [7]), etc. have established themselves as effective weapons for attacking challenging machine learning problems. However, an effective paradigm that can unify all these approaches, seems to be lacking.

In this work, we propose a novel mixture of tree models with random number of components using a flexible Dirichlet process Bayesian framework introduced by (Bhattacharya 2008 [6]). As we shall show, this model includes all the aforementioned tree based approaches as simple special cases. Indeed, when we implement this Bayesian nonparametric model using sophisticated MCMC based approaches combining Gibbs sampler and backfitting MCMC approaches, each such MCMC iteration is expected to yield a relevant model that best explains the underlying data, and such a model may be one of the existing tree models, if borne out by the data. Thus, by model averaging, the posterior predictive of our Dirichlet process model is expected to surpass all the existing tree based approaches in terms of performance. Of course, the true, data-generating model is also expected to learned in terms of minimizing the Kullback-Leibler divergence from the true model. Moreover, the traditional neural network model with one hidden layer is also a special case of our general Dirichlet Process based tree mixture with variable number of components.

We aim to devise a very sophisticated MCMC based Bayesian computational algorithm, that is yet very much amenable to parallel computation, for both CPU and GPU architectures.

2 The Cart Model

2.1 Problem

We have a response variable Y, and a set of predictors $\boldsymbol{x} = (x_1, \cdots, x_p)$. We need to describe the distribution of Y given the predictors.

2.2 Model

The model consists of two components: a tree T with b terminal nodes and a parameter vector $\Theta = (\theta_1, \cdots, \theta_b)$, where the parameter θ_i is associated with the i^{th} terminal node of T.

Each internal node of T has an associated splitting rule that uses a predictor to assign observations to either its left or right child nodes. Hence, the terminal nodes partition the observation space according to the subdivision defined by the splitting rules.

Assumptions:

- If x lies in the region denoted by the i^{th} terminal node, then $y|x$ has distribution $f(y|\theta_i)$, where f is a parametric family indexed by θ_i.

- The distribution of y values inside a terminal node are iid.
- Distribution of y values across different terminal nodes are independent.

For any given tree T, let y_{ij} denote the j^{th} observation associated with the i^{th} terminal node. Here, $j \in \{1, ..., n_i\}, \forall i \in \{1, ..., b\}$, where n_i denotes the number of observations in the i^{th} terminal node. Define:

$$Y \equiv (Y_1, ..., Y_b), \text{where} \quad Y_i \equiv (y_{i1}, ..., y_{in_i})$$

Hence, we have:

$$p(Y|X, \Theta, T) = \prod_{i=1}^{b} f(Y_i|\theta_i) = \prod_{i=1}^{b} \prod_{j=1}^{n_i} f(y_{ij}|\theta_i)$$

f is taken as required for the problem.

3 The Mixture Model

We consider a situation when Y comes from a miture distribution with k components. With the same notions and notations of Chipman et al. (1998), we have the following mixture:

$$[Y|X, \Theta_1, ..., \Theta_M, T_1, ..., T_M] = \sum_{i=1}^{M} p_i f(Y|X, \Theta_i, T_i) \tag{1}$$

where for $i = 1, ..., M, p_i \in [0, 1]$ with $\sum p_i = 1$, and for $i = 1, ..., M$, $(\Theta_i, T_i) \sim G$ and $G \sim DP(\alpha G_0)$, the Dirichlet process with scale parameter α and expected measure G_0 . We assume that under G_0, $[T]$ has a distribution as described in 4.1 and that $[\Theta|T]$ have the same distributions as described in 4.2.

Due to discreteness of Dirichlet process, $(\Theta_i, T_i); i = 1, ..., M$, are coincident with positive probability. This property implies that 1 reduces to the following form:

$$[Y|X, \Theta_1, ..., \Theta_M, T_1, ..., T_M] = \sum_{i=1}^{k} p_i^* f(Y|X, \Theta_i^*, T_i^*) \tag{2}$$

where $k \leq M$, $(\Theta_i^*, T_i^*); i = 1, ..., k$ are the distinct elements of (Θ_i, T_i); $i = 1, ..., M$, and p_i^* is the sum of p_j over indices j such that (Θ_j, T_j) are coincident. We assume a Dirichlet prior distribution over the mixture probabilities $(p_1, ..., p_M)$; however, the deterministic choice $p_i = 1/M$ may also be envisaged and has been demonstrated to outperform the Dirichlet prior in some important applications.

Importantly, 2 is a mixture of random number of components and hence far more flexible compared to all existing tree-based models. Moreover, if $k \to \infty$ (when $M \to \infty$), then 2 can approximate any stand-alone tree-based model. Furthermore, for various values of the scale parameter α, it can be shown that 2 boils down to various popular tree-based models, such as random forest, bagging, boosting, etc. The single layer feedforward neural network model is also included as a special case. Thus, a prior on α encapsulates all the sensible machine learning models.

3.1 Reparameterization

We define the set of allocation variables $Z = (z_1, \cdots, z_n)'$ where z_i denote the class to which the i^{th} point belongs. Let $\Theta^*_M = \{\theta^*_1, \cdots, \theta^*_k\}$ denote the distinct components in Θ_M. Hence, k is the number of distinct components. We take the configuration vector $C = (c_1, \cdots, c_M)'$ where $c_j = l$ iff $\theta_j = \theta^*_l$ for $j \in \{1, \cdots, M\}$ and $l \in \{1, \cdots, k\}$. Here θ_j denotes the pair (Θ_j, T_j).

Hence, the model can be reparameterized in terms of (Z, C, k, Θ^*_M). Notation:

- $Y = (y_1, ..., y_n)$ is the response variable.
- $j \in \{1, ..., M\}$ is used to index the different trees in the mixture model, where M is the maximum possible number of components.
- $g(ij)$ denotes the terminal node to which the i^{th} sample point is assigned to by the j^{th} tree. Here, $\forall j, g(ij) \in \{1, ..., b_j\}$ where b_j denotes the number of terminal nodes in the j^{th} tree.

Hence, the model:

$$p(z_i = j) = \frac{1}{M}, \quad \forall j \in \{1, ..., M\} \tag{3}$$

$$(\Theta_j, T_j) \sim DP(\alpha, G_0) \tag{4}$$

$$p(y_i | z_i = j, \Theta, T) = f(y_i | \theta_{g(ij)}) \tag{5}$$

Where G_0 is a prior on (θ, T) as described in Section 4. Hence,

$$p(Y | Z, \Theta, T) \propto \prod_{j=1}^{m} \prod_{i: z_i = j} f(y_i | \theta_{g(ij)})$$

For Regression Trees, we consider two possible models:

1. Mean Shift model, with $\theta_{g(ij)} = (\mu_{g(ij)}, \sigma^2)$:

$$y_i | (z_i = j, \theta_{g(ij)}) \sim N(\mu_{g(ij)}, \sigma^2)$$

2. Mean-variance Shift model, with $\theta_{g(ij)} = (\mu_{g(ij)}, \sigma^2_{g(ij)})$:

$$y_i | (z_i = j, \theta_{g(ij)}) \sim N(\mu_{g(ij)}, \sigma^2_{g(ij)})$$

For classification trees, $y_i \in \{C_1, ..., C_k\}$, where C_i denotes the different classes. In this case:

$$f(y_i | z_i = j, \theta_{g(ij)}) = \prod_{k=1}^{K} (p_{g(ij),k})^{I(y_i \in C_k)}$$

where $\theta_g = (p_{g1}, ..., p_{gK})$. Here, $p_{gk} := P(y_i \in C_k | z_i = j)$.

4 The Prior Specification

The CART model is well-specified by (Θ, T). The Bayesian Cart model introduced in Chipman et al. (1998)[3] provides a prior distribution $p(\Theta, T)$ for (Θ, T). In this section, we describe this

prior, which will serve our purpose for G_0. For convenience, we use the relationship:

$$p(\Theta, T) = p(\Theta|T)p(T)$$

Hence, we first describe the prior on T as $p(T)$. Then, we will describe the parameter prior, given the tree, denoted by $p(\Theta|T)$.

4.1 The Tree Prior Specification

We first specify $p(T)$ using a tree generating stochastic process.

1. Begin by setting T to be the trivial tree with single node (both root and terminal) denoted by η.
2. split terminal node η with probability $p_{\text{split}}(\eta, T)$.
3. If split occurs, assign it a splitting rule ρ according to the distribution $p_{\text{rule}}(\rho|\eta, T)$ to create right and left children nodes.
4. Continue the above steps 2 & 3 until termination.

Here, $p_{\text{rule}}(\rho|\eta, T)$ and $p_{\text{split}}(\eta, T)$ are functions of the tree above. Typical choices for $p_{\text{split}}(\eta, T)$ are:

1. $p_{\text{split}}(\eta, T) \equiv \alpha$ (constant). Its limitation is that it assigns equal probability to all trees with b terminal nodes irrespective of their shapes.
2. $p_{\text{split}}(\eta, T) = \alpha(1 + d_\eta)^{-\beta}$, where d_η is the depth of that root and $\beta > 0$. Here, the probability to split is a decreasing function of depth. For Large values of β, deeper nodes are unlikely to split.

For $p_{\text{rule}}(\rho|\eta, T)$, we can take the uniform prior. i.e, uniformly choosing a variable to split on and taking a split point uniformly at random for that variable.

Next, we specify the parameter prior $P(\Theta|T)$.

4.2 Parameter Priors

Given a tree structure T_j, we describe the prior for Θ_j associated with T_j. Let b_j denote the number of terminal nodes in the T_j. Further, for any fixed j, we introduce the following notation:

- Let $A_{jg} = \{i : z_i = j, g(ij) = g\}$, i.e, the set of points that lie in the g^{th} terminal node of the j^{th} tree.
- $n_g = |A_{jg}|$, i.e, the number of observations in the g^{th} terminal node of the j^{th} tree.
- $Y_{jg} = \{y_i : i \in A_{jg}\}$

4.2.1 Regression Trees

The simplest specification is the conjugate prior:

$$\mu_g|\sigma, \text{ iid} \sim N\left(\bar{\mu}, \frac{\sigma^2}{a}\right)$$

and

$$\sigma^2 \sim IG\left(\frac{\nu}{2}, \frac{\nu\lambda}{2}\right)$$

Some analytic simplifications show that under the above prior, we get:

$$p(Y_j|X_j, T_j) = \frac{ca^{b_j/2}}{\prod_{g=1}^{b_j}(n_g + a)^{1/2}} \times \left(\sum_{g=1}^{b_j}(s_g + t_g) + \nu\lambda\right)^{-(n+\nu)/2}$$

Here, c is a constant. $s_g = \sum_{i \in A_{jg}}(y_i - \bar{Y_{jg}})^2$ and $t_g = [n_g a/(n_g + a)](\bar{Y_{jg}} - \bar{\mu})^2$, where $\bar{Y_{jg}}$ is the mean of Y_{jg}

4.2.2 Classification Trees

In a classification tree where y_i belongs to one of K categories $C_1, ..., C_K$, we have $\Theta_j = (p_1, ..., p_{b_j})$ has the standard dirichilet distribution of dimension $K - 1$ with $\alpha = (\alpha_1, ..., \alpha_K)$:

$$p_1, ..., p_{b_j}|T \text{ iid} \sim Dirichlet(p_i|\alpha) \propto p_{i1}^{\alpha_1 - 1} ... p_{iK}^{\alpha_K - 1}$$

The above prior gives the following marginal distribution:

$$P(Y_j|X_j, T_j) = \left(\frac{\Gamma\left(\sum_k \alpha_k\right)}{\prod_k \Gamma(\alpha_k)}\right)^{b_j} \prod_{g=1}^{b_j} \frac{\prod_k \Gamma(n_{gk} + \alpha_k)}{\Gamma(n_g + \sum_k \alpha_k)}$$

where $n_{gk} = \sum_{i \in A_{jg}} I(y_i \in C_k)$, $n_g = \sum_k n_{gk}$ and $k \in \{1, ..., K\}$

5 Conditional distributions

We have described our model in section 3 and the prior distributions in section 4. Next, we describe our procedure for Gibbs Sampling by obtaining the conditionals for the posterior distribution.

5.1 Allocation Variables Z

z_i denotes the class to which the i^{th} point belongs.

$$[y_i|z_i = j, C, k, \Theta_M^*] = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2\sigma^2}\left(y_i - \mu_{g(ij)}\right)^2\right\}$$

Hence, we have:

$$[z_i = j \mid Y, Z_{-i}, C, k, \Theta^*_M] \propto \frac{1}{\sigma} \exp\left\{-\frac{1}{2\sigma^2}\left(y_i - \mu_{g(ij)}\right)^2\right\}$$

5.2 Configuration Variables C and k

Let k_j denote the number of distinct values in Θ_{-jM}.

$$[c_j = l \mid Y, Z, C_{-j}, k_j, \Theta^*_M] = \begin{cases} \kappa q_{lj} & \text{if } l \in \{1, \cdots, k_j\} \\ \kappa q_{0j} & \text{if } l = k_j + 1 \end{cases}$$

where,

$$q_{lj} = M_{lj}\left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n_j}{2}} \exp\left[\frac{1}{2\sigma^2}\sum_{i:z_i=j}\left(y_i - \mu_{g(il)}\right)^2\right]$$

Where M_{lj} is the number of times θ^*_l occurs in Θ_{-jM}.

Since this is a non-conjugate case, q_{0j} is not available in closed form. Hence, we modify our Gibbs sampling strategy by bringing in auxiliary variables in a way similar to that of Algorithm 8 in Neal (2000) [4] as described in Mukhopadhyay and Sourabh Bhattacharya 2012 [8]. To clarify, let $\theta^a = (\Theta^a, T^a)$ denote an auxiliary variable. Then, before updating c_j we first simulate from the full conditional distribution of θ^a given the current c_j and the rest of the variables as follows:

- if $c_j = c_l$ for some $l \neq j$, then $\theta^a \sim G_0$
- if $c_j \neq c_l \forall l \neq j$, set $\theta^a = \theta^*_{c_j}$

Once θ^a is obtained, we replace q_{0j} with the tractable expression:

$$q^a_j = \alpha\left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n_j}{2}} \exp\left[\frac{1}{2\sigma^2}\sum_{i:z_i=j}\left(y_i - \mu_{g(ia)}\right)^2\right]$$

once all the c_j are updated, k is obtained as the number of unique values in C.

5.3 Conditionals for $\theta_j = (\Theta_j, T_j)$

We assume that under G_0 , $[T]$ and $[\Theta|T]$ have the same distributions as specified in Sections 3 and 4, respectively, of Chipman et al. (1998). Hence, the posterior here is identical to the posterior distribution of the Bayesian Cart model.

Consider the set:

$$Y_j = \{y_i : z_i = j\}$$

We simulate (T_j, Θ_j) by the posterior search metropolis - hastings algorithm described below. We perform this search for all the distinct components: (Y_j, T_j, Θ_j), $j \in \{1, ..., k\}$:

5.3.1 Tree Posterior Search

For this section, we dropping the subscript j for simplicity. We describe the search for the triplet (Y, T, Θ).

For the metropolis Hastings algorithm, we first consider the following transition kernel:

The kernel $q(T, T^*)$ generates a new tree T^* from the tree T by randomly choosing one of the four steps:

- GROW: Randomly pick a terminal node and split it into two new ones by randomly assigning it a splitting rule according to p_{RULE} used in the prior.
- PRUNE: Randomly pick a parent of two terminal nodes and turn it into a terminal node by collapsing the nodes below it.
- CHANGE: Randomly pick an internal node, and randomly reassign it a splitting rule according to p_{RULE} used in the prior.
- SWAP: Randomly pick a parent-ehild pair that are both internal nodes. Swap their splitting rules unless the other child has the identical rule, in which case swap the splitting rule of the parent with that of both children.

Using the above kernel, we have the following M-H algorithm to generate the markov chain $T_0, T_1, ...$:

1. Generate a candidate T^* from distribution $q(T_i, T^*)$
2. Define:
$$\alpha(T_i, T^*) := \min\left\{\frac{q(T_i, T^*)}{q(T_*, T^i)}\frac{p(Y|X, T^*)P(T^*)}{p(Y|X, T^i)P(T^i)}\ ,\ 1\right\}$$
Set $T_{i+1} := T^*$ with probability $\alpha(T_i, T^*)$ and T_i with probability $1 - \alpha(T_i, T^*)$.

Here, it is important to note that $p(Y|X, T)$ can be explicitly obtained by integrating over the parameters Θ, as has been shown in the section 4.2.

5.3.2 Parameter Posterior

Once we have drawn T_j from its posterior, we next simulate Θ_j by gibbs sampling:

$$p(Y_j, \Theta_j|Z, T) \propto \prod_{g=1}^{b_j}\left(\prod_{i:g(ij)=g}\frac{1}{\sigma}\exp\left[-\frac{1}{2\sigma^2}(y_i - \mu_g)^2\right]\right)\times\prod_{g=1}^{bj}\frac{\sqrt{a}}{\sigma}\exp\left[\frac{a}{2\sigma^2}(\mu_g - \bar{\mu})^2\right]\times\frac{\beta^\alpha}{(\sigma^2)^{\alpha+1}}\exp\left(-\frac{\beta}{\sigma^2}\right)$$

Hence, given σ^2, posterior for μ_g is $N(\mu_1, \sigma_1^2)$ where:

$$\sigma_1^2 = \left(\frac{a}{\sigma^2} + \frac{n_g}{\sigma^2}\right)^{-1} = \frac{\sigma^2}{a + n_g}$$

and

$$\mu_1 = \sigma_1^2 \left(\frac{a\bar{\mu}}{\sigma^2} + \frac{n_g \bar{Y_{gj}}}{\sigma^2} \right)$$

Again, given μ_g for all the terminal nodes of T_j, we have the posterior of σ^2 as:

$$\sigma_j^2 \mid (\mu_{j1}, ..., \mu_{jb_j}),\ T_j \sim IG\left(\frac{\nu}{2} + \frac{n_j}{2}, \frac{\nu\lambda}{2} + \frac{1}{2} \sum_{g=1}^{b_j} \sum_{i \epsilon A_{jg}} (y_i - \mu_{jg})^2 \right)$$

where $A_{jg} = \{i : z_i = j, g(ij) = g\}$, i.e, the set of points that lie in the g^{th} terminal node of the j^{th} tree.

References

[1] Leo Breiman. In: *Machine Learning* 24.2 (1996), pp. 123–140. DOI: 10.1023/a:1018054314350. URL: https://doi.org/10.1023/a:1018054314350.

[2] Yoav Freund and Robert E Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Journal of Computer and System Sciences* 55.1 (Aug. 1997), pp. 119–139. DOI: 10.1006/jcss.1997.1504. URL: https://doi.org/10.1006/jcss.1997.1504.

[3] Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. "Bayesian CART Model Search". In: *Journal of the American Statistical Association* 93.443 (Sept. 1998), pp. 935–948. DOI: 10.1080/01621459.1998.10473750. URL: https://doi.org/10.1080/01621459.1998.10473750.

[4] Radford M. Neal. "Markov Chain Sampling Methods for Dirichlet Process Mixture Models". In: *Journal of Computational and Graphical Statistics* 9.2 (2000), pp. 249–265. ISSN: 10618600. URL: http://www.jstor.org/stable/1390653.

[5] Leo Breiman. In: *Machine Learning* 45.1 (2001), pp. 5–32. DOI: 10.1023/a:1010933404324. URL: https://doi.org/10.1023/a:1010933404324.

[6] Sourabh Bhattacharya. "Gibbs Sampling Based Bayesian Analysis of Mixtures with Unknown Number of Components". In: *Sankhyā: The Indian Journal of Statistics, Series B (2008-)* 70.1 (2008), pp. 133–155. ISSN: 09768386, 09768394. URL: http://www.jstor.org/stable/41234427.

[7] Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. "BART: Bayesian additive regression trees". In: *The Annals of Applied Statistics* 4.1 (Mar. 2010), pp. 266–298. DOI: 10.1214/09-aoas285. URL: https://doi.org/10.1214/09-aoas285.

[8] Sabyasachi Mukhopadhyay and Sourabh Bhattacharya. "Perfect Simulation for Mixtures with Known and Unknown Number of Components". In: *Bayesian Analysis* 7.3 (Sept. 2012), pp. 675–714. DOI: 10.1214/12-ba723. URL: https://doi.org/10.1214/12-ba723.