Published in final edited form as:

J Comput Graph Stat. 2016 ; 25(3): 748–761. doi:10.1080/10618600.2015.1089774.

Bayesian Ensemble Trees (BET) for Clustering and Prediction in Heterogeneous Data

Leo L. Duan¹, John P. Clancy², and Rhonda D. Szczesniak^{3,4}

¹Department of Mathematical Sciences, University of Cincinnati

²Division of Pulmonary Medicine, Cincinnati Children's Hospital Medical Center

³Division of Biostatistics and Epidemiology, Cincinnati Children's Hospital Medical Center

Abstract

We propose a novel “tree-averaging” model that utilizes the ensemble of classification and regression trees (CART). Each constituent tree is estimated with a subset of similar data. We treat this grouping of subsets as Bayesian Ensemble Trees (BET) and model them as a Dirichlet process. We show that BET determines the optimal number of trees by adapting to the data heterogeneity. Compared with the other ensemble methods, BET requires much fewer trees and shows equivalent prediction accuracy using weighted averaging. Moreover, each tree in BET provides variable selection criterion and interpretation for each subset. We developed an efficient estimating procedure with improved estimation strategies in both CART and mixture models. We demonstrate these advantages of BET with simulations and illustrate the approach with a real-world data example involving regression of lung function measurements obtained from patients with cystic fibrosis. Supplemental materials are available online.

Keywords

Bayesian Mixture of Trees; Dirichlet Process; Ensemble Approach; Heterogeneity

1 Introduction

Classification and regression trees (CART), or decision tree, (Breiman et al., 1984) is a nonparametric approach that provides fast partitioning of data through the binary split tree and an intuitive interpretation for the relation between the covariates and outcome. Aside from simple model assumptions, CART is not affected by potential collinearity or singularity of covariates. From a statistical perspective, CART models the data entries as

⁴Corresponding author. Address: 3333 Burnet Ave, MLC 5041, Cincinnati, OH 45229. Ph: (513) 803-0563. rhonda.szczesniak@cchmc.org.

Supplementary Materials

Appendix Marginal posterior likelihood, Random Forests results, convergence diagnostics, and additional application results. (Appendix.pdf)

Implementation in Python BET sampler code and README file for implementation (BETcode.zip)

The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

conditionally independent given the partition, which not only retains the likelihood simplicity but also preserves the nested structure.

Since the introduction of CART, many approaches have been derived with better model parsimony and prediction. The Random Forests model (Breiman, 2001) generates bootstrap estimates of trees and utilizes the bootstrap-aggregating (“bagging”) estimator for prediction. Boosting (Friedman, 2001, 2002) creates a generalized additive model of trees and then uses the sum of trees for inference. Bayesian CART (Denison et al., 1998; Chipman et al., 1998) assigns a prior distribution to the tree and uses Bayesian model averaging to achieve better estimates. Bayesian additive regression trees (BART, Chipman et al. (2010)) combine the advantages of the prior distribution and sum-of-trees structure to gain further improvement in prediction.

Regardless of the differences in the aforementioned models, they share one principle: multiple trees create more diverse fitting than a single tree; therefore, the combined information accommodates more sources of variability from the data. Our design follows this principle. We create a new ensemble approach called the Bayesian Ensemble Trees (BET) model, which utilizes the information available in the subsamples of data. Similar to Random Forests, we propose taking the average of the trees, in which each tree achieves an optimal fit without any restraints. We determine the subsamples through the Dirichlet process clustering rather than bootstrapping. This setting automates control of the number of trees.

Another motivation to use Dirichlet process is to introduce mixture modeling into the CART framework. In essence, CART can be considered as a partitioning tool that reduces data to several regions, of which each region is expected to follow a simple unimodal distribution. However, this assumption is often violated by two sources of variations: 1) multiple modes exist in the outcome distribution and are not separable with the predictor variables; 2) more than one type of relationship exists between the outcome and predictors.

In the following sections, we first introduce the model notation and its sampling algorithm. Then, we demonstrate the capability of BET to handle the aforementioned two scenarios. Next, we use BET on a regression problem with continuous data on lung function measurements collected on patients with cystic fibrosis. Lastly, we discuss BET results and its possible extensions.

2 Preliminary Notation

We denote the i th record of the outcome as Y_i , which can be either categorical or continuous. Each Y_i has a corresponding covariate vector $\mathbf{X}_i$. In the standard CART model, we generate a binary decision tree $\mathbf{T}$ that uses only the values of $\mathbf{X}_i$ to assign the i th record to a certain region. As the tree works as a function, we denote the output region as $\mathbf{T}(\mathbf{X}_i)$. In each region, elements of assigned Y are assumed to be identically and independently distributed as $g(\boldsymbol{\theta})$, where g can be any common distribution such as normal, $\boldsymbol{\theta}$ are parameters and vary by regions. Our goals are to find the optimal tree $\mathbf{T}$, predict unknown

Y_s given new values of $\mathbf{X}_s$, and make inference about the importance of different predictor variables in $\mathbf{X}$.

We now extend this framework to further assume that $\{Y_j, \mathbf{X}_j\}$ is from one of infinitely many trees $\{\mathbf{T}_j\}_{j=1 \dots \infty}$. Its true origin is only known up to a probability w_j . Therefore, we need to estimate both $\mathbf{T}_j$ and w_j for each j . Since it is impossible to estimate over all j 's, we only calculate those j 's with non-negligible w_j after a probabilistic truncation.

3 Bayesian Ensemble Tree (BET) Model

We now formally define our proposed model. We use $[\cdot]$ to denote the probability density. Let $[Y_i | \mathbf{X}_i, \mathbf{T}_j]$ denote the probability of Y_i conditional on its origination from the j th tree. The mixture likelihood can be expressed as:

$$[Y_i | \{\mathbf{T}_j(\mathbf{X}_i)\}_{j=1 \dots \infty}] = \sum_j w_j [Y_i | \mathbf{T}_j(\mathbf{X}_i)] \quad (1)$$

where the mixture weight vector has an infinite-dimension Dirichlet distribution with precision parameter α : $\mathbf{W} = \{w_j\}_{j=1 \dots \infty} \sim \text{Dir}_{\infty}(\alpha)$. The likelihood above corresponds to Dirichlet process $Y_i \stackrel{iid}{\sim} DP(\alpha, G)$, in which base distribution G is the binary tree $[Y_i | \mathbf{T}_j(\mathbf{X}_i)]$.

3.1 Hierarchical Prior for $\mathbf{T}$

We first define the nodes as the building units of a tree. We adopt the notation introduced by Wu et al. (2007) and use it to describe the graphical relationship among nodes. We denote the root node with index 0. A given node k (parent) can have either zero (not split) or two (split) child nodes, indexed as left child ($2k+1$) and right child ($2k+2$). We further define two names for two special types of node: if a node has no children, we call it a leaf; if a node has two children both as leaf nodes, we refer to it as a twig node.

Each node has 4 parameters $\{s, c, t, \theta\}$. As the first parameter, $s = 1$ or $s = 0$ denotes the condition whether a node is split or not. To have a meaningful graph, the child node cannot split ($s_{child} = 1$) unless its parent node is split ($s_{parent} = 1$). Therefore, the shape of the tree can be described by this probability factorization:

$$[s_0=1] \prod_{k \in \{l: s_l=1\}} [s_{2k+1} | s_k=1] [s_{2k+2} | s_k=1]$$

where we assume at least the root is split ($s_0 = 1$). We assign the following Bernoulli prior for splitting: $[s_{2k+1} = 1 | s_k=1] = [s_{2k+2} = 1 | s_k=1] = \exp(-d/\delta)$, where $d = \lfloor \log_2(k+1) \rfloor$ and denotes the depth of a node. As the depth increases, the nodes become less likely to split. The choice of δ is empirical, and we found that setting $\delta = 1000$ tends to yield satisfactory prediction performance with a moderately deep tree.

The second parameter c denotes a multinomial draw from $\{1, \dots, p\}$, where p is the number of predictors (or columns) in $\mathbf{X}$. We assume that for each tree, this multinomial distribution

has its probability vector as a p -dimensional Dirichlet distribution, $\boldsymbol{\varphi} \sim \text{Dir}(0.5 \cdot \mathbf{1}_p)$. Therefore, the posterior distribution of $\boldsymbol{\varphi}$ reflects how many times a certain predictor has been chosen to construct the tree. The ranking of posterior probability among p variables can be considered as a probabilistic counterpart as the variable importance index in Random Forests. For this reason, we refer to this posterior probability as a variable ranking probability. In the multiple tree setting, each tree would have its own variable ranking probability.

The third parameter t is the splitting threshold that if the value of c th predictor for the i th subject $X_{(i,c)} \geq t$, the node passes the data entry to its left child, otherwise to its right child. This parameter is randomly chosen from the entries of the c th predictor of $\mathbf{X}_c$. The prior distribution is empirical and assumed to be equally likely for all unique values in $\mathbf{X}_c$ assigned to the current node.

Lastly, the parameters $\boldsymbol{\theta}$ in each partition are the ones for the distribution g . For a continuous outcome, one choice for g can be normal distribution; for a binary outcome, g can be a Bernoulli distribution. One ideal prior for $\boldsymbol{\theta}$ is a non-informative prior, such as $\boldsymbol{\theta} = \mathbf{p} \sim \text{Beta}(0.5, 0.5)$ for a binary outcome. However for a continuous outcome, use of a non-informative prior, such as $[\boldsymbol{\theta}] = [\mu, \sigma^2] \propto 1/\sigma^2$ corresponding to a normal outcome, could create posterior impropriety. This is because we cannot guarantee each partition will contain a sufficient number of data points. For example, in our multiple tree fitting approach, it is possible after reassigning the data to another tree (a step documented in the next section), some node in the current tree could end up with 1 or 0 data. Our workaround is to use diffuse prior if a proper non-informative prior is not available. For example, for a normal distribution, we use $\boldsymbol{\theta} = [\mu, \sigma^2]$, where $\mu \sim N(0, 10^5)$ and $\sigma^2 \sim IG(0.1, 0.1)$. This ensures posterior propriety.

3.2 Stick-Breaking Prior for $\mathbf{W}$

The dimension change of the Dirichlet process creates difficulties in estimation. Pioneering studies include exploring infinite state space with the reversible-jump Markov chain Monte Carlo (Green and Richardson, 2001) and with an auxiliary variable for possible new states (Neal, 2000). At the same time, an equivalent construction named the stick-breaking process (Ishwaran and James, 2001) gained popularity for its decreased computational burden. The stick-breaking process decomposes the Dirichlet process into an infinite series of *Beta* distributions:

$$\begin{aligned} w_1 &= v_1 \\ w_j &= v_j \prod_{k < j} (1 - v_k) \text{ for } j > 1 \end{aligned} \quad (2)$$

where each $v_j \stackrel{iid}{\sim} \text{Beta}(1, \alpha)$. This construction provides a straightforward illustration of the effects of adding/deleting a new cluster to/from the existing clusters.

Another difficulty in sampling is that j is infinite. Ishwaran and James (2001) demonstrated that the $\max(j)$ can be truncated to 150 for a sample size of $n = 10^5$, and the results are

indistinguishable from those obtained using larger numbers. Later, Kalli et al. (2011) introduced the slice sampler, which avoids the approximate truncation. Briefly, the slice sampler adds a latent variable $u_i \sim \mathcal{U}(0, 1)$ for each observation. The probability in (1) becomes:

$$[Y_i | \mathbf{X}_i, \{\mathbf{T}_j\}_j] = \sum_j^{\infty} \mathbf{1}(u_i < w_j) [Y_i | \mathbf{X}_i, \mathbf{T}_j] \quad (3)$$

due to $\int_0^1 \mathbf{1}(u_i < w_j) du_i = w_j$. The Monte Carlo sampling of u_i leads to omitting w_j 's that are too small. We found that the slice sampler usually leads to a smaller effective max $j < 10$ for $n = 10^5$, hence more rapid convergence than a simple truncation. Lastly, we use $Z_i = j$ to denote the latent assignment of the i th observation to the j th tree.

4 Posterior Sampling

We now explain the sampling algorithm for the BET model. We initiate the sampler with random assignment of data over many trees. The initial number of trees is considered large enough if the sampler converged to a fewer number of trees. Then, the sampler iterates over three steps: tree fitting, data reassignment and weight updating.

4.1 Tree Fitting: Localized Gibbs Sampling

Each tree with assigned data is fitted in this step. Chipman et al. (1998) used a random choice in the grow/prune/swap/change (GPSC) in one Metropolis-Hastings (MH) step to update the tree structure. For the grow or prune step, GPSC randomly chooses one change operation on one selected node, whereas for the swap or change step, it proposes aggressive change in the tree structure. By restarting with different random seeds, this algorithm has been shown to explore the diverse structure of a single tree quite well. However, as BET is a mixture model, the GPSC method is not suitable, since restarting would lead to a label switching issue.

To prevent non-convergence and label switching issues, we want each component tree to have a relatively stable structure. On the other hand, to reduce the number of needed trees, we want each tree to “grow deep” so that it can thoroughly describe its assigned data. For these two purposes, it would be ideal to use an algorithm that updates nearly all parameters as one iteration, while keeping the move as local as possible. Therefore, we propose the following localized Gibbs sampling scheme.

First, we update sequentially the parameters s in all the leaf nodes (the ones without any children), which is equivalent to proposing “grow” change to every leaf. Then, we update s , c , t in the all twig nodes (each of these node has two leaf children), which are equivalent to exploring “prune”, “predictor change” and “threshold change”, respectively. In these procedures, the Metropolis-Hastings criteria are used. We accept the proposal if a continuous uniform (0,1) random variable u has $u < \alpha$; otherwise we reject the proposal and keep the current value. The acceptance probabilities are calculated in the following manner:

For s_k ,

if $s_k = 0$:

$$\alpha = \frac{\prod_i [Y_i | \mathbf{T}_{new}(\mathbf{X}_i)] [s_k=1] [s_{2k+1}=0] [s_{2k+2}=0]}{\prod_i [Y_i | \mathbf{T}_{cur}(\mathbf{X}_i)] [s_k=0]}$$

if $s_k = 1$:

$$\alpha = \frac{\prod_i [Y_i | \mathbf{T}_{new}(\mathbf{X}_i)] [s_k=0]}{\prod_i [Y_i | \mathbf{T}_{cur}(\mathbf{X}_i)] [s_k=1] [s_{2k+1}=0] [s_{2k+2}=0]}$$

For c_k and t_k :

$$\alpha = \frac{\prod_i [Y_i | \mathbf{T}_{new}(\mathbf{X}_i)] [c_{k,new}]}{\prod_i [Y_i | \mathbf{T}_{cur}(\mathbf{X}_i)] [c_{k,cur}]}$$

In the Metropolis-Hasting step, the one-step grow proposal is a small transformation from a terminating node $s_k = 0$ to a open node $s_k = 1$ with two child terminating nodes $s_{2k+1} = 0$ and $s_{2k+2} = 0$. We choose the marginal likelihood with θ integrated out in each leaf and form $\prod_i [Y_i | \mathbf{T}_i] = \prod_m M_m$, where m is index for the leaf nodes. The alternative representation may be using conditional likelihood including θ , however, we find that it takes ~ 30 additional cycles for θ to converge, each time when the data partition is changed (i.e. when tree structure is changed). Therefore, using marginal likelihood to assess the posterior of the tree structure is more efficient. The marginal likelihoods for the normal and binary outcomes are provided in the Appendix.

The presented formulas are similar to those in GPSC and exhibit simplicity due to the

cancellation of the constant proposal density. The likelihood ratio $\frac{\prod_i [Y_i | \mathbf{T}_{new}]}{\prod_i [Y_i | \mathbf{T}_{old}]}$ is also quite simple since it involves at most redistribution of data in a subtree with three nodes, and the remainder of the likelihood in both parts is canceled. The Gibbs sampler explores every direction of micro change by sampling through all the terminals of the tree. While at the same time, it is only localized to the leaf and twig nodes, so it does not bring dramatic change. As one can imagine, a dramatic change in the high-level branch of a tree would require the sequential and successful pruning of its children from the bottom, which is unlikely. This leads to a quick convergence to a local minimum of cost function (negative log-likelihood). Similar to the other Gibbs samplers for mixture models, this algorithm does not focus on finding the global minimum.

4.2 Data Reassignment: Slice Sampler

We first adopt the latent assignment variable Z as documented by Diebolt and Robert (1994). Since the Dirichlet process involves an infinite number of components, we also take advantage of the latent uniform random variable $u_i \sim U(0, 1)$ introduced by Kalli et al. (2011). The probability of having the i th observation assigned to the j th tree (a multinomial draw) is

$$[Z_i=j|Y_i, \mathbf{W}, \mathbf{T}_j, \boldsymbol{\theta}] = w_j [Y_i|\mathbf{T}_j, \boldsymbol{\theta}] = \int \mathbf{1}(w_j > u_i) [Y_i|\mathbf{T}_j, \boldsymbol{\theta}] du_i \quad (4)$$

We replace the last integral with the Monte Carlo sample of u_j , which has posterior distribution $u_j \sim U(0, w_j)$. As a direct result, given u_j in the multinomial draw, the component trees with mixture weight $w_{(\cdot)}$ u_j will have 0 probability of being drawn. This leads to an effective truncation of component number, but theoretic equivalence to the infinite sampling. The method is referred to as the slice sampler by Kalli et al. (2011). As we start with a large number of components and let the algorithm reduce it to a smaller number with assigned data (and the rest with no data). Unlike reversible-jump Markov chain Monte Carlo (Green and Richardson, 2001), this sampler does not require explicitly adding or deleting a tree.

In this step, one modification in the likelihood is that we now use the conditional likelihood of $[Y_i|\mathbf{T}_j, \boldsymbol{\theta}]$ with explicit values of $\boldsymbol{\theta}$. This is due to the fact that the conditional likelihood $\boldsymbol{\theta}$ is easier to evaluate than the marginal likelihood for each individual data point. As mentioned before, we obtain the estimates of $\boldsymbol{\theta}$ by running ~ 30 cycles for $\boldsymbol{\theta}$. To clarify, we use the marginal likelihood without $\boldsymbol{\theta}$ to assess the tree structure change, as computational efficiency is desired; and given the tree structure, we use the conditional likelihood with $\boldsymbol{\theta}$ to compute the affinity of data to each tree, as computational convenience is preferred.

4.3 Weight Updating: Stick-Breaking Sampler

As the last step, we use the stick-breaking sampler to update the weight $w_{(\cdot)}$.

$$\begin{aligned} v_j &\sim \text{Beta}(1 + \sum_i \mathbf{1}(Z_i=j), \alpha + \sum_i \mathbf{1}(Z_i > j)) \\ w_j &= v_j \prod_{k < j} (1 - v_k) \end{aligned} \quad (5)$$

As the self-reinforcing nature, the component with more data assignment $\sum_i \mathbf{1}(Z_i=j)$ will tend to have larger w_j . This contributes to collapsing the component number to only a few as the posterior sample converges.

5 Simulation Studies

In this section, we demonstrate the unique advantages of BET in handling heterogeneous data (see Supplemental Material for software implementation). Through simulations, we show how it overcomes the shortcomings of the basic CART and simple tree averaging method. Logically, we assume the heterogeneity may come from two sources: 1) the distribution of Y in each partition after a tree is fitted; 2) the relationship between Y and $\mathbf{X}$. In the former case, this is equivalent to imagining that the data arise in a mixture distribution inside each leaf; in the latter case, the data can be viewed as a combination of several distinct populations, for which the same predictors $\mathbf{X}$ play different roles across different subgroups. Regardless of the source of heterogeneity, we show that BET provides a nice accommodation to these situations in the same “mixture of trees” framework.

5.1 Heterogeneity in the distribution of Y

One of the strengths of the decision tree lies in its ability to partition the data based on the predictors $\mathbf{X}$. Even if the outcome Y has multiple modes, after the partitioning, in each leaf it may become unimodal. That being said, sometimes the information in $\mathbf{X}$ is not sufficient to separate the modes. Equivalently, this can be viewed as the data takes a mixture distribution inside some partition, or “a tree of mixtures”. To illustrate this case, we conduct a simulation with the following data generation scheme (Table 1). Clearly, each of the partitions 1, 2 and 3 has two modes.

To assess the performance of the model, we make the distances between the modes as small as only one standard deviation (0.5). As shown in Figure 1(a), the two groups have a large amount of overlap. Obviously, the two groups cannot be separated by a decision tree. One solution may be prescribing mixture distribution inside each leaf; however, this is not necessary. As we show in Figure 1(b), BET successfully identified two trees with the same splitting criterion in $\mathbf{X}$, but two different estimates in Y for each partition. Despite the challenges arising from the small distance, all of the means are correctly estimated from the bimodal distributions. Comparing Figure 1 (a) and (b), one may notice the tree assignments in (b) (stars and circles) are not exactly the same as we use in the data generation (a). This is not surprising since the labels are exchangeable in a mixture distribution.

Next, for comparison, we applied CART (with maximum depth = 5) and Random Forests (with number of trees = 1000) on the same data. As shown in Figure 4(a), since there is only one tree in CART, it converges to roughly 3 partitions in the results, with some overfitting. Random forests do have multiple trees in the model; however, due to the lack of the systematic assignment of the data to the tree, each tree merely represents a bootstrap sample. Therefore, only the bootstrap aggregate can be used for estimation. As shown in Figure 4(b), it yields similar results of three partitions, in each the estimate seems to be equal to the average of the two true modes. In summary, the results illustrate the necessity of a mixture framework for such data, and show that BET provides an equivalent representation of “a tree of mixtures” data via a “mixture of trees” approach.

5.2 Heterogeneity in the Relationship between Y and $\mathbf{X}$

We now illustrate the second case of heterogeneity: the relationship between Y and $\mathbf{X}$ differs from one subset of the data to another. In this case, the partitioning criterion in $\mathbf{X}$ changes across subgroups of data; therefore, more than one tree is required in the model. This scenario is analogous to having different regression coefficients (β 's) for several subgroups of data. We generate the data according to the setup in Table 2.

For the clarity of illustration, we plot the means of the six partitions in Figure 2(a). Obviously, the partitioning schemes in $\mathbf{X}$ are different with some overlapping in Partition 3 and 6. The fitted means from BET are shown in Figure 2(b). Apart from some label changes, two partitioning schemes are found with most of the means correctly estimated.

It is worth mentioning that we did encounter some non-convergence in the number of data assigned in each tree: we have about 100 data points that kept switching from one tree to another. After closer scrutiny, it becomes evident that this was caused by the data in the

overlapping area in $X_1, X_2 \in [0.6, 0.9] \times [0.0, 0.4]$. Since both trees have the same estimate of 3.0 for Y in that region, the data assignments become quite random. To support this argument, we moved the mean of Partition 6 to 3.5, and the model indeed converged to stable data counts.

We again test the performance of CART and Random Forests on the simulated data (Figure 3). Like the previous simulation, these models lead to the average estimate over the data in each region. Without addressing the data structure, the averaging over two distinct means can cause large variance in some region, such as the one in $X_1, X_2 \in [0.0, 0.4] \times [0.0, 0.4]$.

6 Cystic Fibrosis Data Example

To demonstrate the use of BET on real-world data, we now use lung function data obtained from a national cystic fibrosis (CF) patient registry (Foundation, 2012) (available upon request by contacting the Cystic Fibrosis Foundation). Briefly, percent predicted of forced expiratory volume in 1 second (FEV₁%) is a continuous measure of lung function in CF patients obtained at each clinical visit. We have previously demonstrated that the rates of FEV₁% change nonlinearly over patient in this cohort (Szczesniak et al., 2013), which can be accommodated by decision tree methods.

We used the 263,615 data entries from the CF dataset described in our previous analysis. Here, we utilize eight clinical covariates: baseline FEV₁%, age, gender, infections from Methicillin-resistant *Staphylococcus aureus*, *Pseudomonas aeruginosa*, Burkholderia cepacia, and CF-related diabetes, (abbreviated as MRSA, PA, BC, CFRD, respectively) and state/federal insurance (a measure of socioeconomic status, SES). We randomly selected 80% of data as training and the remaining 20% as a testing set. We collected the posterior samples from 20,000 steps of Markov chain Monte Carlo after 10,000 steps as burn-in. This process took about 4 hours using our Python implementation. The convergence characteristics of the chain were diagnosed by the trace plots of the mixture weights and the likelihood (see Appendix).

We illustrate the prediction results of BET in Figure 5(a) and (b). With the assumption of one constant mean per partition, the predicted line takes the shape of step functions, which correctly captured the declining age-related trend of FEV₁%. The prediction appears to be unbiased, as the difference between the predicted and true values are symmetric about the diagonal line. We also computed the difference metrics with the true values. The results are listed in Table 3.

As shown in the table, besides the BET model, we also tested other popular tree methods (using corresponding Python packages), such as CART (Therneau et al. (1997)), Random Forests (Breiman (2001)), and Boosting (Friedman (2001)). We do not claim that BET always has better prediction performance than existing methods; however, it demonstrates great parsimony among all the ensemble methods – as the MAP estimates for the cystic fibrosis data show that BET only requires 2 trees to produce the prediction as accurately as Random Forests, which require 40 trees. This is a huge benefit from the self-reinforcing clustering of the Dirichlet process.

Lastly, we focus on the variable selection issue. Although some information criteria have been established in this area, such as AIC, BIC and Bayes Factor, their mechanisms are complicated by the necessity to fit different models to the data for multiple times. The inclusion probability (reviewed by O'Hara et al. (2009)) is an attractive alternative, which penalizes the addition of more variables through the inclusion prior. In the tree-based method, however, since multiplicity is not a concern, it is possible to compare several variables of similar importances at the same time, without inclusion or exclusion.

Since multiple trees are present, we use the weighted posterior $\bar{\xi} = \sum_j w_j \xi_j$ as the measure for variable importance. We plot the variable ranking probability $\bar{\xi}$ for each covariate (Figure 5). The interpretation of this value follows naturally as how likely a covariate is chosen in forming the trees. Therefore, the ranking of $\bar{\xi}$ reveals the order of importance of the covariates. This concept is quite similar to the variable importance measure invented in Random Forests. The difference is that the index used in Random Forests is ranked according to the decrease of accuracy after covariate permutation; the BET index is purely a probabilistic measure. Regardless of this difference, the ranking of variables from the two models are remarkably similar (see Appendix for results of Random Forests): baseline FEV₁% and age are the two most important variables, while gender and MRSA seem to play the least important roles of the clinical covariates.

7 Discussion

Empirically, compared with a single model, a group of models (an ensemble) usually has better performance in inference and prediction. This view is also supported in the field of the decision trees. As reviewed by Hastie et al. (2009), the use of decision trees has benefited from multiple-tree methods such as Random Forests, Boosting and Bayesian tree averaging. One interesting question that remains, however, is how many models are enough, or, how many are more than enough?

To address this issue, machine learning algorithms usually resort to repeated tests of cross validation or out-of-bootstrap error calculation, in which ensemble size is gradually increased until performance starts to degrade. In Bayesian model averaging, the number of steps to keep is often “the more, the better”, as long as the steps have low autocorrelation. Our proposed method, BET, demonstrates a more efficient way to create an ensemble with the same predictive capability but much smaller size. With the help of the Dirichlet process, the self-reinforcing behavior of large clusters reduces the number of needed clusters (sub-models). Rather than using a simple average over many trees, we showed that using a weighted average over a few important trees can provide the same accuracy.

It is worth comparing BET with other mixture models where the component distributions are typically continuous and unimodal. In the BET model, each component tree is discrete, and more importantly, multi-modal itself. This construction could have created caveats in model fitting, as one can imagine only obtaining a large ensemble of very small trees. We circumvented this issue by applying Gibbs sampling in each tree, which rapidly increases the fit of tree to the data during tree growing, and decreases the chance that they are scattered to more clusters.

It is also of interest to develop an empirical algorithm for BET. One possible extension is to use a local optimization technique (also known as the “greedy algorithm”) under some randomness to explore the tree structure. This implementation may be facilitated by existing CART packages to grow trees for subsets of data. Users may employ these existing routines then update clustering using aforementioned techniques.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

The authors are grateful to the Cystic Fibrosis Foundation Patient Registry Committee for their thoughtful comments and data dispensation. Partial funding was provided by the Cystic Fibrosis Foundation Research and Development Program (grant number R457-CR11) and National Institutes of Health (grant number K25 HL125954).

References

- Breiman L. Random forests. *Machine learning*. 2001; 45(1):5–32.
- Breiman, L.; Friedman, J.; Olshen, R.; Stone, C. *Classification and Regression Trees*. Wadsworth and Brooks; Monterey, CA: 1984.
- Chipman HA, George EI, McCulloch RE. Bayesian CART model search. *Journal of the American Statistical Association*. 1998; 93(443):935–948.
- Chipman HA, George EI, McCulloch RE, et al. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*. 2010; 4(1):266–298.
- Denison DG, Mallick BK, Smith AF. A bayesian CART algorithm. *Biometrika*. 1998; 85(2):363–377.
- Diebolt J, Robert CP. Estimation of finite mixture distributions through bayesian sampling. *Journal of the Royal Statistical Society. Series B (Methodological)*. 1994:363–375.
- Foundation CF. Cystic fibrosis foundation patient registry 2012 annual data report. 2012.
- Friedman JH. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*. 2001:1189–1232.
- Friedman JH. Stochastic gradient boosting. *Computational Statistics & Data Analysis*. 2002; 38(4): 367–378.
- Green PJ, Richardson S. Modelling heterogeneity with and without the Dirichlet process. *Scandinavian journal of statistics*. 2001; 28(2):355–375.
- Hastie, T.; Tibshirani, R.; Friedman, J. *The elements of statistical learning*. Vol. 2. Springer; 2009.
- Ishwaran H, James LF. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*. 2001; 96(453):161–173.
- Kalli M, Griffin JE, Walker SG. Slice sampling mixture models. *Statistics and computing*. 2011; 21(1): 93–105.
- Neal RM. Markov chain sampling methods for Dirichlet process mixture models. *Journal of computational and graphical statistics*. 2000; 9(2):249–265.
- O’Hara RB, Sillanpää MJ, et al. A review of Bayesian variable selection methods: what, how and which. *Bayesian analysis*. 2009; 4(1):85–117.
- Szczesniak RD, McPhail GL, Duan LL, Macaluso M, Amin RS, Clancy JP. A semiparametric approach to estimate rapid lung function decline in cystic fibrosis. *Annals of epidemiology*. 2013; 23(12):771–777. [PubMed: 24103586]
- Therneau TM, Atkinson EJ, et al. An introduction to recursive partitioning using the rpart routines. 1997
- Wu Y, Tjelmeland H, West M. Bayesian CART: Prior specification and posterior simulation. *Journal of Computational and Graphical Statistics*. 2007; 16(1):44–66.

(a) The scatterplot of the data. Two groups of the data are marked in gray/black stars and circles, which overlap heavily. (b) The fitted values using the BET model. Two groups are successfully separated with most of the means correctly estimated.

Figure 1.

The BET model accommodates mixture distribution in Y via a “mixture of trees” approach.

(a) The fitted values using the CART method. (b) The fitted values using the Random Forests.

Figure 2.

The BET model successfully identified two populations from the multiple modes.

(a) The means used in the data generated by two (b) The fitted values using the BET model. Two different partitioning schemes (gray/black stars populations are successfully separated with most and circles). of the means correctly estimated.

Figure 3.

For comparison, using CART or Random Forests method does not consider the heterogeneity of the relationships between Y and $\mathbf{X}$. The simple average estimates can result in large variance in the estimates in some region.

(a) The fitted values using the CART method. (b) The fitted values using the Random Forests.

Figure 4.

For comparison, using CART or Random Forests method does not address the multiple modes in the data. As a result, the estimates converge to the average of the two modes in each partition.

(a) True values (dots) and the 95% credible interval of the BET predicted (solid) in a single subject. (b) True values and the BET predicted means: the estimates are piecewise constant and seem unbiased.

(c) Number of trees in the MCMC sample

(d) The variable selection probabilities

Figure 5.
Results of FEV₁% fitting and prediction.

Table 1

The data generation scheme for the case of multimodal distribution inside the same tree. The modes are not separable by the tree partitioning based on X . The two modes inside each leaf are close and only one standard deviation apart.

Partition Index	Data Count	X_1	X_2	Y
1	100	$U(0.1, 0.4)$	$U(0.1, 0.4)$	$N(1.0, 0.5^2)$
2	100	$U(0.1, 0.4)$	$U(0.6, 0.9)$	$N(2.0, 0.5^2)$
3	100	$U(0.6, 0.9)$	$U(0.1, 0.9)$	$N(3.0, 0.5^2)$
1	100	$U(0.1, 0.4)$	$U(0.1, 0.4)$	$N(1.5, 0.5^2)$
2	100	$U(0.1, 0.4)$	$U(0.6, 0.9)$	$N(2.5, 0.5^2)$
3	100	$U(0.6, 0.9)$	$U(0.1, 0.9)$	$N(3.5, 0.5^2)$

Table 2

The data generation scheme for the case of heterogeneous relationships between $\mathbf{X}$ and Y . Two different partitioning schemes exist in the space of $\mathbf{X}$, comparing partitions $\{1,2,3\}$ and $\{4,5,6\}$.

Partition Index	Data Count	X_1	X_2	Y
1	100	$U(0.1,0.4)$	$U(0.1,0.4)$	$N(1.0, 0.5^2)$
2	100	$U(0.1,0.4)$	$U(0.6,0.9)$	$N(2.0, 0.5^2)$
3	100	$U(0.6,0.9)$	$U(0.1,0.9)$	$N(3.0, 0.5^2)$
4	100	$U(0.1,0.4)$	$U(0.6,0.9)$	$N(1.5, 0.5^2)$
5	100	$U(0.6,0.9)$	$U(0.6,0.9)$	$N(2.5, 0.5^2)$
6	100	$U(0.1,0.9)$	$U(0.1,0.4)$	$N(3.0, 0.5^2)$

Table 3

Cross-validation results with various methods applied on cystic fibrosis data.

Model	Number of trees used for prediction	RMSE	MAD
CART	1	14.49	8.66
Random Forests	5	10.68	6.03
Boosting	500	13.88	8.37
Bayesian CART	10,000	10.50	6.24
Random Forests	40	9.87	5.56
BET	2	9.90	5.55
Bayesian averaging of BET	10,000	9.52	5.25