Bayesian Additive Regression Trees for Bayesian Quadrature

Anonymous Authors[1]

Abstract

Bayesian Quadrature (BQ) is an important tool for solving statistical and scientific problems with an integral at their heart. We propose a new approach to BQ based on Bayesian Additive Regression Trees (BART). BART is easy to tune, automatically handles a mixture of discrete and continuous variables, and has attractive theoretical results (Rockova & Saha, 2019). We show how BART lends itself to an elegant formulation of BQ, with a simple but effective sequential sampling approach. We first present an algorithm that performs BQ with BART on a set of benchmark tests, known as the Genz functions. Our model outperforms both Gaussian processes for BQ and Monte Carlo integration in high dimensional situations ($d > 10$) and non-smooth functions. The second algorithm that we present performs Bayesian survey design with BART, providing an effective alternative to both a simple random sample (Monte Carlo) and a more sophisticated block (stratified) random sampling design.

1. Introduction

On a measure space $(\mathbb{R}^d, \mathcal{B}_{\mathbb{R}^d}, \mu)$ with μ absolutely continuous with respect to the Lebesgue measure, for $d \in \mathbb{N}\backslash\{0\}$ and a Borel measurable function $f : \mathbb{R}^d \to \mathbb{R}$, we wish to approximate the integral:

$$I_f = \int_{\mathbb{R}^d} f d\mu = \int_{\mathbb{R}^d} f(x)p(x)dx, \tag{1}$$

where dx represents integration with respect to the Lebesgue measure, and p is the Radon-Nikodym derivative with respect to the Lebesgue measure, by the Radon-Nikodym theorem. This problem could be understood to be applying the integration operator $\mathcal{L} : V \to \int_{\mathbb{R}^d}(\cdot)d\mu$, where V is a space of Borel measurable functions.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

This problem, Bayesian Quadrature (BQ), generalises classical approaches to numerical integration (e.g. quadrature) and is at the centre of the emerging field of Probabilistic Numerics (PN). PN takes a Bayesian statistical perspective on numerical methods. The most widely applied method in Probabilistic Numerics is Bayesian Optimisation (BO), in which an unknown objective function f is modelled with a Bayesian statistical model, and maximised using a global optimisation method which critically relies upon the posterior distribution of the function. Gaussian processes have played the dominant role in Probabilistic Numerics as a convenient Bayesian nonparametric model for f (e.g. (Rasmussen & Ghahramani, 2002)).

As with BO, BQ critically relies upon a flexible Bayesian statistical model for the integrand f. Although Gaussian processes (GPs) are a powerful non-parametric Bayesian prior for functions f with known posterior consistency results and concentration rates (van der Vaart & van Zanten, 2008) which carry over to BQ (Briol et al., 2015) and promises faster convergence rates than classical Monte Carlo-based integration. While they are virtually the only choice that has been explored in the BQ literature, in practice they suffer from a number of challenges. GPs are often not very effective in high dimensions (Shahriari et al., 2016), evidenced by the fact that in BQ, few applications consider more than about 10 dimensions. While particular kernels can be designed for non-stationarity and categorical variables, both settings nevertheless pose problems. Learning kernel hyperparameters is a long-standing challenge in the Gaussian process literature, made even more difficult by the ubiquity of small sample sizes in BQ (Rasmussen & Ghahramani, 2002). Most BQ approaches rely on simple (e.g. Gaussian) choices for the prior $p(\cdot)$ and kernel (Gaussian again), but neither choice is necessarily appropriate in real settings. Finally, the particular form of the variance of the posterior distribution of a Gaussian process is such that the actual observed values of the function are irrelevant, with a peculiar result—information gained through a sequential, active learning-type setup is not actually incorporated into our model's choices about where to look next (except indirectly through the kernel hyperparameters, which we may retrain.)

As a baseline, classical Monte Carlo integration (Press et al., 2007) only requires the ability to sample from $p(\cdot)$, and enjoys solid theoretical justifications (Durrett, 2010), but

suffers from the problem of having large variance.

BART, a sum-of-trees model extending Bayesian CART, requires little need for hyperparameter tuning, enjoys attractive theoretical results (Rockova & van der Pas, 2017) and is robust, in the sense of not overfitting, to f having unknown regularity (Rockova & Saha, 2019). We show how BART is a natural choice for BQ, lending itself to an elegant formulation of Bayesian Quadrature with a simple but effective sequential design approach.

Viewing a regression tree as a step function , it follows that with a single tree $f(\cdot) = \sum_{i=1}^{n} \theta_j \mathbb{1}_{\Omega_j}(\cdot)$, where $\{\Omega_j\}_j$ is a set of tree-shape partitions of the tree and 1 is an indicator function (Rockova & Saha, 2019), the integral in (1) becomes

$$I_f = \sum_{i=1}^{n} \theta_i p_i, \tag{2}$$

where θ_i is the value of terminal node θ_i, n is the number of terminal nodes of the tree model and $p_j = \mu(\Omega_j) = \int_{\Omega_j} p(x)dx$, with dx indicating integration with respect to the Lesbesgue measure.

The BART model, simply speaking, is a sum-of-trees model where each tree is constrained by a regularisation prior to be a weak learner (Chipman et al., 2010). When training the model, apart from the training set, we introduce a sequential design approach, a form of active learning (Chipman et al., 2012), of sampling candidates from a known prior distribution, selecting the best candidates and adding them to the training set. Our integral I_f could then be approximated by the sum-of-tree form of Eq. (2). Furthermore, we demonstrate that our sequential design may also be used for quadrature in the context of survey design, where we approximate the moments of stratified and un-stratified population metrics.

2. Bayesian Additive Regression Trees (BART)

Given a p-dimensional input $x = (x_1, \ldots, x_p) \in \mathbb{R}^p$ and its response Y, BART (Chipman et al., 2010) aims to make inference on an underlying and unknown regression function

$$Y = f(x) + \epsilon, \epsilon \sim N(0, \sigma^2), \tag{3}$$

by estimating $f(x) = E(Y|x)$. For a BART model, for every posterior draw we have that for each tree T, with b terminal nodes and a set of parameters $\Theta = \{\theta_1, \ldots, \theta_b\}$, with θ_i associated to the i^{th} terminal node, denote, $g(x; T, \Theta)$ as the step function which assigns each $\theta_i \in \Theta$ to x. Thus, a single tree BART model can be expressed as

$$E(Y|x) = g(x; T, \Theta). \tag{4}$$

Now define the sum-of-trees models as

$$Y = \sum_{j=1}^{m} g_j(x; T_j, \Theta_j) + \epsilon, \epsilon \sim N(0, \sigma^2), \tag{5}$$

where m is the number of trees and $g_j(x; T_j, \Theta_j)$ is the function that assigns $\theta_{ij} \in \Theta_j$ to x.

For a fixed m and a single draw from the posterior, the sum-of-trees model is determined by $(T_1, \Theta_1), (T_2, \Theta_2), \ldots, (T_m, \Theta_m)$ and σ.

In order to perform posterior inference given training data $\mathcal{D}$, the Metropolis-Hastings algorithm is used to propose updates to each tree structure. This is combined with a Bayesian backfitting algorithm. For details, see Chipman et al. (2010). ensures that $(T_1, \Theta_1), \ldots, (T_m, \Theta_m), \sigma$ converges in distribution to $((T_1, \Theta_1), \ldots, (T_m, \Theta_m), \sigma|\mathcal{D})$ and the induced sum-of-trees function

$$f_{\text{BART}}(\cdot) = \textstyle\sum_{j=1}^{m} g(\cdot; T_j, \Theta_j)$$

hence converges in distribution to $f|\mathcal{D}$, the posterior of the true $f(\cdot)$ (Chipman et al., 2010).

3. Methodological Contribution

3.1. Bayesian Quadrature with BART

We now present our Bayesian Quadrature with BART method. Let f be the function we wish to integrate over a measure space. We assume that it is Hölder-continuous for the sake of obtaining nice posterior concentration rates which guarantee that BART will not overfit (Rockova & Saha, 2019). We will learn a function f_{BART} and approximate Eq. (1) with

$$I_f \approx \int_{\mathbb{R}^d} f_{\text{BART}} d\mu = \int_{\mathbb{R}^d} f_{\text{BART}}(x) p(x) dx. \tag{6}$$

Since each tree is a step function, there are coefficients for each indicator function that we will call the "value at the terminal node". We denote the value of the i^{th} terminal node of the j^{th} posterior draw in the k^{th} tree by $\theta_{i,j}^k$. Similarly, $p_{i,j}^k = \mu(\Omega_{i,j}^k)$ as discussed before in Eq (2). The details to obtain $p_{i,j}^k$ are included in subsection 3.2.

Depending on the tree structure, we have the step functions $f_{j,\text{BART}}^1, \ldots, f_{j,\text{BART}}^K$, which stand for each of the K trees, which is the pre-set value in the BART model as the number of trees, in the j^{th} posterior draw. Thus the approximation of Eq. (6) by the j^{th} set of trees becomes

$$I_f^{(j)} \approx \sum_{k=1}^{K} \sum_{i=1}^{b_{k,j}} \theta_{i,j}^k p_{i,j}^k, \tag{7}$$

where $b_{k,j}$ is the number of terminal nodes for tree k.

After fitting BART, we obtain N trees drawn from the posterior, and calculate $I_f^{(1)}, \ldots, I_f^{(N)}$. These N draws from the posterior over the integral can be summarised using the mean and variance as

$$\mathrm{E}[I_f|\mathcal{D}] \approx \frac{1}{N}\sum_{j=1}^{N} I_f^{(j)}, \tag{8}$$

$$\mathrm{Var}[I_f|\mathcal{D}] \approx \frac{1}{N}\sum_{j=1}^{N}\left(I_f^{(j)} - \mathrm{E}[I_f|\mathcal{D}]\right)^2. \tag{9}$$

3.2. Calculating Terminal Node Probabilities

For simplicity, we start by assuming our experiments take place in the unit d-dimensional hypercube $[0,1]^d$. We assume a uniform probability measure μ over the inputs, with constant probability density function $p(x)$. In practice, the measure could instead be Gaussian or discrete.

For a tree in a posterior draw, the terminal node probability p is obtained through multiplying the probability at leaf node θ_l at level l of the tree along the branch. Suppose we have a d-dimensional uniform prior $p(x)$ in range $[0,1]^d$, and x_r is the rth element in x, given the range $(R_{l,L}^r, R_{l,U}^r)$ of possible x_r being allocated to the node θ_l, and the cutpoint C_l^r on x_r, the probability at the next left-branch node making decision on is

$$p_{l+1,\text{Left}} = \frac{C_l^r - R_{l,L}^r}{R_{l,U}^r - R_{l,L}^r} \tag{10}$$

and the probability of the next right branch node is

$$p_{l+1,\text{Right}} = 1 - p_{l+1,\text{Left}} = \frac{R_{l,U}^r - C_l^r}{R_{l,U}^r - R_{l,L}^r}. \tag{11}$$

Figure 1 gives a brief illustration of how we find the node probability on d-dimensional variables.

3.3. Sequential Bayesian Quadrature with BART

We propose a simple and elegant sequential design approach to BQ using BART. The idea of sequential design is to select samples to improve model performance by adding more training data to the original data set, through an active learning procedure.

Our method, summarised in Algorithm 1 is as follows: we randomly generate a set of candidates $\mathcal{C} = \{c_1, \ldots, c_L\} \sim p(x)$. Using our existing BART model which we have already fit f_{BART}, we use this model to calculate the posterior predictive variance for each candidate. Our selection criterion is very simple: the point that we are least certain about is the one that we will query next. Thus we maximize the posterior predictive variance:

$$c^* = \text{argmax}_{c\in C}\mathrm{Var}[f_{\text{BART}}(c)|\mathcal{D}] \tag{12}$$

Having selected c^* we query the true function f to obtain $y_c^* = c^*$ and update our training set.

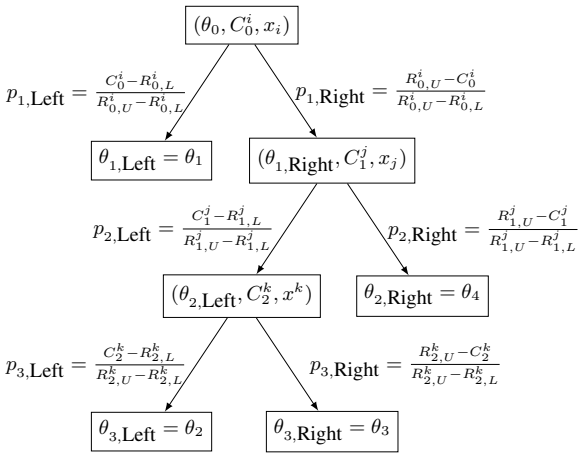

Figure 1. Example of a decision tree with 3 levels for a d-dimensional input and uniform prior. Pairs $(\theta_{s,\text{Left}}, C_l^r, x_r)$ and $(\theta_{l,\text{Right}}, C_l^r, x_r)$ are the nodes and their associated cutpoints at level l on x_r. Terminal node probabilities can be worked out by multiplying along the branches.

Algorithm 1 Sequential Design

Input:
training set $\mathcal{D} = \{x_1, \ldots, x_n\}$,
response $\mathcal{Y} = \{y_1, \ldots, y_n\}$,
number of iterations M
probability density function $p(x)$
for M iterations **do**
  obtain N samples from the posterior distribution of the BART model with data $\mathcal{D}$ and $\mathcal{Y}$
  sample candidate set $\mathcal{C} = \{c_1, \ldots, c_L\} \sim p(x)$
  find $f_{\text{BART}}^{(1)}(c), \ldots, f_{\text{BART}}^{(N)}(c)$ for all $c \in \mathcal{C}$
  find $c^* = \text{argmax}_c \mathrm{Var}[f_{\text{BART}}(c)|\mathcal{D}]$
  find response $y_c^* = f(c^*)$
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{c^*\}$, $\mathcal{Y} \leftarrow \mathcal{Y} \cup \{y_c^*\}$
end for

4. Model Performance on Benchmark Tests

To test our algorithm, we use a standard benchmark set of multidimensional integrands proposed by Genz[1] (Genz, 1984), consisting of six families of functions with a set of parameters which can be varied to vary the level of difficulty of the integration problem.

[1] http://www.sfu.ca/~ssurjano

We examine the rate of convergence of our BART approach to BQ when estimating the integrals in Eq. (1) for each Genz family and compare it two baselines: Monte Carlo integration and Bayesian Quadrature with Gaussian processes. The Genz functions are defined on the unit d-dimensional hypercube $[0,1]^d$ and this is taken as the domain of integration. For simplicity, we choose as our prior distribution $p(x)$ the uniform distribution over $[0,1]^d$. The Genz functions have two sets of parameters, d "ineffective" parameters u and d "effective" parameters a which vary the level of difficulty. We use the default setting of $u = [0.5, \ldots, 0.5]^\top$ and re-scale a suitably as the dimension increases to ensure numerical stability. Specifically, this is done by bounding the L_1-norm of a so that numerical stability is obtained (Schürer, 2001). To generate draws from $p(x)$ we use Latin Hypercube sampling (Press et al., 2007). As ground truth, we analytically compute the integrals for these Genz test functions, which we include with further explanation in the Appendix.

4.1. Implementation Details

4.1.1. BART-BQ

We use the *bart* function in the `dbarts` package (R Core Team, 2018) when implementing the BART model. We mostly use default hyperparameter settings, following (Chipman et al., 2010). However, we found that for our experiments, 50 trees was sufficient (as opposed to a default of 200). We also scaled the prior variance of the node parameters by $k = 5$ (as opposed to a default of $k = 2$) to ensure that a larger range of values can be assigned to the response for the difficult integration problems.

The prior of the node parameters, or the coefficients of the indicator functions, is given a Gaussian prior distribution in the Bayesian setting. Through the rescaling operation and the prior assignment, larger probability is assigned to the range of the response variable, following the works of (Chipman et al., 2010). The prior of the residual variance σ receives an inverse chi-squared distribution $\nu\lambda/\chi^2_\nu$ with $(\nu, \lambda) = (3, 0.9)$ as suggested in (Chipman et al., 2010), which helps us avoid overfitting. Lastly, the size of tree is controlled by the parameters α and β mentioned by (Hugh A. Chipman, 1998). We use the default setting $(0.95, 2)$ throughout the experiments.

To ensure the convergence of our MCMC procedure, we use a burn-in period of 1000 iterations, followed by 1000 draws. We then thin by a factor of 20 to obtain a final set of 50 draws. Finally, as the response data is implicitly scaled by the *bart* function onto interval $[-0.5, 0.5]$ to optimize performance (Chipman et al., 2010), we re-scale them by the following inverse transformation

$$\hat{y}^* = (y^* + 0.5)(\boldsymbol{y}_{max} - \boldsymbol{y}_{min}) + \boldsymbol{y}_{min}. \tag{13}$$

Starting from an initial sample of size 100, we implement sequential Bayesian Quadrature with Gaussian Process 500 times by iteratively selecting a new point using Algorithm 1.

4.1.2. Sequential Bayesian Quadrature with Gaussian processes

Our main competitor is Sequential Bayesian Quadrature with Gaussian processes (GP-BQ), the leading BQ approach presented in literature. It works by placing a Gaussian process prior on the integrand in Eq. (1) (thus also on the integral I_f) and inference on the true integral value is made by considering the posterior mean and variance (Rasmussen & Williams, 2005). As in BART-BQ, we refine this method through an active learning procedure. 500 new sample points that maximise the posterior sample variance are added sequentially via almost the same scheme. Furthermore, we choose the Gaussian kernel given by

$$k(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2l^2}\right). \tag{14}$$

Ideally, hyperparameters should be learned, either by maximizing the marginal likelihood (the usual practice) or placing a prior on them and then sampling. However, for computational convenience, we set $\sigma = 1$ in all cases by standardizing the responses y and apply the median heuristic, choosing the bandwidth l as the median distance between points drawn from a uniform prior over the unit hypercube of dimension d. Unfortunately, this default approach meant that we sometimes obtained negative posterior variances, a problem that has been discussed previously in the literature (Rasmussen & Ghahramani, 2002). Details of our complete formulation can be found in the appendix.

With an active procedure, although costly, it significantly increases the posterior concentration rate of the mean to the true integral value, as illustrated in Figure 2.

4.1.3. Monte Carlo Integration

We further compare our method with the crude Monte Carlo estimation (Press et al., 2007)

$$I_f \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i), \tag{15}$$

where x_i are again generated by Latin Hypercube sampling with sample size n increasing from 1 to 500.

4.2. Experimental Results

Table 1 shows the outcome of our benchmark tests for all six Genz families with dimensions 1, 2, 3, 5, 10 and 20. For

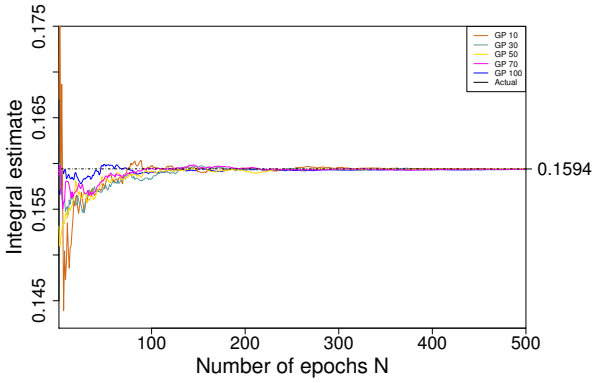

Figure 2. Convergence of the GP estimates of the oscillatory Genz family of dimension 10 with 12, 30, 50, 70 and 100 candidates for sequential design. The black dotted line is the true integral value.

Table 1. A comparison of each method on the 6 Genz test integrands, over a range of dimensions. The best estimate in terms of the root mean square error (RMSE) distance to the true value on the last iteration is shown in bold. In low dimensions, all methods are competitive, while BART seems to perform better in dimension $d = 20$. BART also does better for functions with discontinuities (the "disc" family).

MethodDim	Method	d = 1	2	3	5	10	20
cont	BART	**0.00127**	0.00274	0.00171	**0.000394**	0.000247	**0.000165**
	MC	0.00506	**0.000711**	0.00243	0.00748	**7.95e-05**	0.000278
	GP	0.00873	0.000932	**6.07e-05**	0.00126	0.00104	0.0163
copeak	BART	0.00117	0.00019	9.06e-05	1.12e-06	2.89e-07	**4.52e-08**
	MC	**0.000109**	**1.94e-05**	7.79e-06	9.09e-07	5.25e-08	2.47e-07
	GP	0.00148	6.06e-05	**7.21e-06**	**4.74e-07**	**2.35e-08**	7.12e-07
disc	BART	NA	**375**	2.78	**0.00284**	**0.000431**	**0.00241**
	MI	NA	405	**2.06**	0.111	0.0262	0.0119
	GP	NA	536	10.9	0.0979	0.0418	0.0171
gaussian	BART	0.00588	0.00103	**0.00111**	**0.000917**	0.00121	0.000396
	MC	0.00735	**0.000678**	0.00294	0.00476	0.00686	**0.000226**
	GP	**0.00381**	0.00235	0.00672	0.00643	**0.000836**	0.0154
oscil	BART	0.018	0.0072	0.025	0.0366	0.00169	**0.000128**
	MC	**0.0118**	0.0296	0.0418	0.0243	0.0212	0.00175
	GP	0.0598	**5.8e-05**	**0.0136**	**0.000736**	**0.000245**	0.0153
prpeak	BART	**1170**	54000	27200	**2840**	6.14e-08	6.47e-50
	MC	2160	**11000**	59800	11900	1.16e-07	**3.68e-50**
	GP	1580	39800	**16200**	6870	**5.79e-09**	1.92e-47

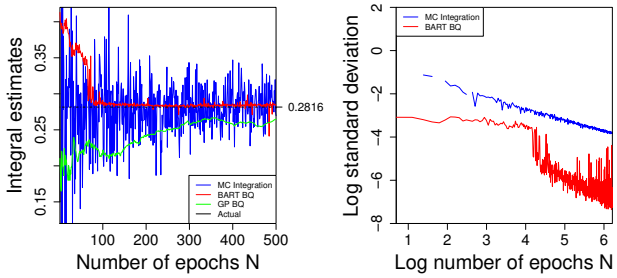

Figure 3. Convergence of the integral of the discontinuous Genz function of dimension 20 run with 500 epochs of sequential design. The true integral value is 0.2816, as indicated above.

illustration purposes, we only show a couple of the convergence plots and we refer the reader to the supplementary material for the full set of results.

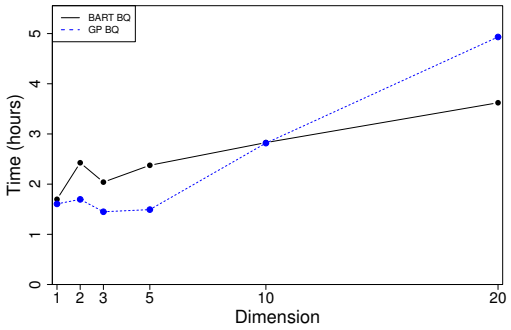

Figure 4. Run time of BART-BQ and GP-BQ with 500 additional samples collected using sequential design.

As would be expected theoretically, BART-BQ outperforms the other two methods when the function is non-smooth, e.g. the discontinuous family. Indeed, existing Bayesian quadrature methods are known to struggle when estimating discontinuous functions due to the need to tune hyperparameters and the choice of the kernel function (for GP-BQ) (Rasmussen & Ghahramani, 2002). Furthermore, both GP-BQ and Monte Carlo work well mostly for low dimensions (< 20). There is a clear degradation in convergence for dimension 20 for our experiments, as shown in Table 1.

Figure 3 displays the rate of convergence for the discontinuous family with dimension 20. Note that the posterior variance for GP empirically always tends to zero, as the predictive variance does not depend on the response (Rasmussen & Williams, 2005), or even negative due to numerical issues (Rasmussen & Ghahramani, 2002) and is out of scale, thus being omitted in the plot. The BART-BQ estimates converge remarkably fast compared to GP-BQ. It also appears to be unbiased as opposed to a systematic bias exhibited by the GP estimator, which is most likely due to the undesired functional form of this family. The standard error of Monte Carlo estimates, on the other hand, decreases at the rate of $\frac{1}{\sqrt{N}}$, which is illustrated in Figure 3. But the standard error exhibited by crude Monte Carlo is still larger than BART-BQ. Note also that there is an abrupt decrease in the variance of BART-BQ. This may be due to the addition of a particular point in a previously unexplored region.

BART-BQ is outperformed by GP-BQ when estimating the integral of the oscillatory family. We show the convergence plot for dimension 5 in Figure 5 as an example. This result is probably due to the fact that the oscillatory family is relatively smooth, and so it can be modeled well by a Gaussian process with a smooth kernel (van der Vaart & van Zanten, 2008), while BART is more appropriate for non-smooth functions. Nevertheless, the difference in performance between the two methods is significant.

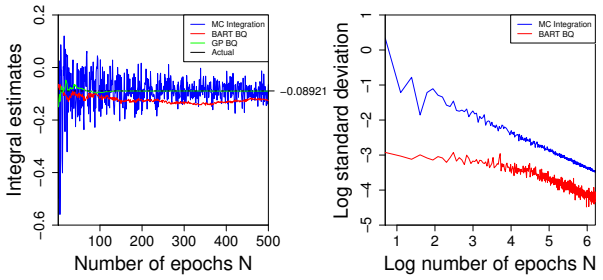

Figure 5. Convergence of the integral of the oscillatory Genz function of dimension 5 run with 500 epochs of sequential design. The actual integral is -0.08921 as indicated above.

4.3. Running time

We compare the run time of BART-BQ to that of GP-BQ as we vary the dimensionality d of the input space. As BART relies on an MCMC routine, which does not have a deterministic running time, we follow (Chipman et al., 2010) and analyse its running time empirically in Figure 4, where we average the run-times for all Genz families for a given dimension. We see that the time complexity of GP-BQ grows more rapidly as the dimension increases. than of that BART-BQ. This agrees with the empirical results in (Chipman et al., 2010) which found that BART was relatively insensitive to the dimensionality.

GP-BQ is usually proposed in settings where it is costly to obtain samples, so the sample size is low and the $O(n^3)$ running time of GPs is not an issue. There are, however, certain settings in which n might be very large—for example, a large set of initial samples, on the order of $10,000$ could be obtained through parallel random sampling. But for $n > 10,000$, GPs become too costly (without further approximations which may decrease the effectiveness of BQ). By contrast, BART was shown empirically to have $O(n)$ time complexity (Chipman et al., 2010).

5. Bayesian Survey Design

We now present a novel use of BQ-BART in the context of a different integration problem: survey sampling. We propose a new method called "Bayesian Survey Design." Classically, the gold standard in survey sampling is the simple random sample, i.e. Monte Carlo. Given a set of survey respondents, estimating the population mean is equivalent to calculating the expectation of the integral of the response values (an unknown function) over the respondents who were sampled (the prior). Having phrased this problem as an integration problem, we propose the use of Bayesian Quadrature for survey design, as an alternative to the simple random sampling.

We further consider using demographic variables to model the response variable. Bayesian hierarchical models are often used in this setting to analyse survey data, in order to stabilise estimates, make them more representative, and to borrow strength when making sub-population estimates for underrepresented subgroups or locations (Gelman & Hill, 2006). Suppose we have a small set of survey responses consisting of interest, income, and categorical demographic variables (race, age, sex, etc). In addition, we have a much larger set of individuals for whom demographic variables are known but for whom the response variable income is unknwon. We can consider surveying any of these individuals to ask them their income. Monte Carlo sampling would choose them at random, regardless of their demographics. We consider the use of Sequential Bayesian Quadrature with BART to intelligently choose the next individual to survey. We hope that this will lead to lower variance estimates with smaller sample sizes, making survey sampling more efficient. A standard refinement of simple random sampling is block (also known as stratified) random sampling. By conditioning on a variable of interest (e.g. race), we ensure more diversity in our samples and thus obtain lower variance estimates. Our approach holds the promise of automatically ensuring a diverse sample.

5.1. Exploratory Data Analysis

We obtain the 2017 American Community Survey Public Use Microdata Sample (PUMS) from Wyoming (Bureau, 2017), with attributes `Mobility`, `Employment`, `Sex`, `Education`, `Disability`, `Health insurance`, `Own child`, `Race`, and response `Total person income`. The goal is to predict the average income of the the population. Our main interest is to estimate the average income of the working population and so we eliminate the observations with negative or zero income. Figure 6 shows the histogram of distribution of income.

To enable experiments on a more refined model, and test whether our approach is competitive with stratified random sampling, we categorise the population into two groups by education level: either below or higher than high school education.

5.2. Experimental Design

We are first going to estimate the mean income of the whole population with some initial observations and add M 'best candidates' to the survey by Sequential Bayesian Quadrature with BART. Then to make the problem more complex, we stratify our data by education level and estimate the population mean for each category, using the same setup.

The details of our approach differ slightly from those ex-

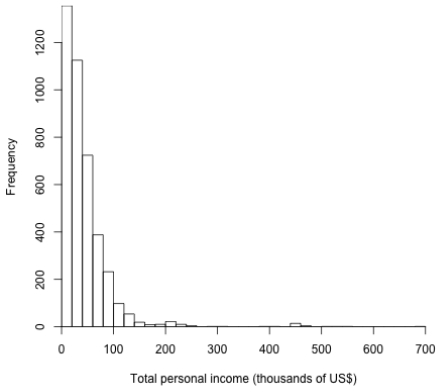

Figure 6. Histogram of total personal income (thousands of US$).

plained in Algorithm 1. Assume we are given a set of samples $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$, where x_i is a demographic covariate vector and y_i is a real-valued label. We have a further set of candidates $\mathcal{C} = \{c_1, \ldots, c_L\}$ with known covariates but unknown responses. The goal of our design is to sequentially select M individuals. Algorithm 2 describes our sequential design approach. We use a discrete uniform prior and assign each candidate an equal $\frac{1}{L}$ chance to be selected.

Algorithm 2 Sequential Design for Survey

Input:
set of labeled samples $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$,
candidate set $\mathcal{C} = \{c_1, \ldots, c_L\}$
M number of new samples to collect
for M iterations **do**
  fit BART with $\mathcal{D}$
  find posterior predictive distribution $f_{\text{BART}}(c)$ for all $c \in \mathcal{C}$
  find $c^* = \text{argmax}_c \text{Var}[f_{\text{BART}}(c)|\mathcal{D}]$
  obtain the survey response y_c^* for c^*
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(c^*, y_c^*)\}$
  $\mathcal{C} \leftarrow \mathcal{C} \setminus \{c^*\}$
end for

In the real world, the size of new samples M could be limited by the budget or time available for conducting the survey, or other constraints.

Our BART model can be used to provide posterior predictive distributions for each of the remaining individuals whose income information is missing. Denote $f_P^j(x_i)$ as the prediction of individual x_i given by the kth draw from the posterior, we have the posterior estimated expected value of income

$$E[Y|\mathcal{D}] = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{N}\sum_{i=1}^{N} f_P^j(x_i), \tag{16}$$

where N is the number of individuals with unknown income and K is the number of posterior draws.

The posterior variance can similarly be calculated by

$$Var[Y|\mathcal{D}] = \frac{1}{K}\sum_{j=i}^{K}\left(\frac{1}{N}\sum_{i=1}^{N} f_P^j(x_i) - E[Y|\mathcal{D}]\right)^2. \tag{17}$$

We compare to simple random sampling (Monte Carlo) and block random sampling, stratifying by education level.

5.3. Experimental Results

Our dataset had 4,076 individuals. We begin by observing 50 random individuals, leaving the rest as a set of candidates with known demographics and unknown responses. As will be mentioned in Section 6, BART requires a minor tuning process. To achieve the best performance, we build the model with 40 posterior draws, each with 50 trees and the parameters specified in the previous section.

We compare the mean estimates from the different procedures, as well as the standard error of simple random sampling and block random sampling versus BART's posterior standard deviation calculated from Eq. (17).

Figure 7 shows that by collecting further income information from 1000 individuals selected using Sequential Bayesian Quadrature, BART's estimation of average income converges much faster, with lower standard error and higher accuracy than either Monte Carlo or block random sampling.

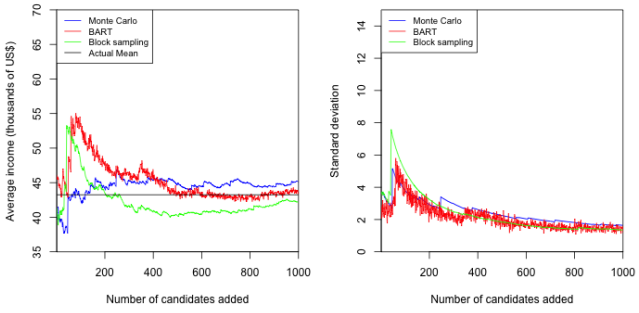

Figure 7. Estimation (left) and standard error (right) of average total income (thousands of US$) with different sampling methods.

To test our method in a more difficult setting, we stratify by education level, using the exact same model, but make estimates only for the average income of people with education level beyond high school. The BART-BQ design outperforms simple random sampling in Figure 8 in terms of both

posterior standard deviation and estimation. Interestingly, BART-BQ still converges after only 500 samples.

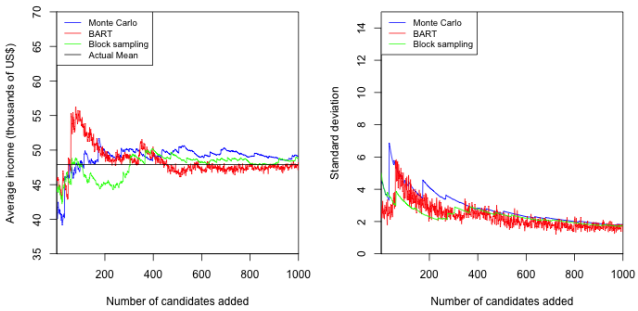

Figure 8. Estimation (left) and standard error (right) of average total income (thousands of US$) of population with education level beyond high school, given by different sampling methods.

In the above experiments, all of the three methods provide estimates within a reasonable error margin, but BART-BQ's survey design achieves a convergence rate faster with relatively accurate estimates and lower standard error. The experiments support our proposal that when existing information and ability of obtaining further information are limited, BART-BQ with sequential design performs better than traditional random sampling methods, with little manual tuning of the method required.

Moreover, in real life, we may be able to obtain a larger initial dataset, which will lead to further improvements in BART-BQ's performance.

6. Discussion

We have shown that BART-BQ provides convincing outcomes for estimating integrals. The highlight is that the method significantly resolves some of the key issues of the current state-of-the-art algorithms, GP-BQ and Monte Carlo integration. Performance in high dimensions is one big advantage of BART over the other methods, as we have discovered through our integration of Genz functions. The ease of tuning, compared to that of GP-BQ, is also a significant advantage.

Given that BART is a non-parametric method, obtaining an effective BART model will still require choosing good parameters. For example, increasing number of trees and number of posterior draws will significantly decrease the variability of prediction but with a longer training time; a good burn-in period would also improve the overall performance of the model. The tuning procedure also is much simpler than that of the GP, where the choice of the kernel and its parameters are somewhat difficult to interpret mathematically.

There are now a few questions that need to be addressed in the future regarding BART-BQ. We have demonstrated its capability in both numerical integration of Genz functions and in inference through survey design. However, we still need to show mathematically that under certain regularity conditions, we are able to obtain good posterior concentration rates for the BART integral approximations to the true integral. The theory for the BART function has already been established (Rockova & Saha, 2019), and so it remains to resolve it for the case when we apply a linear integration operator to our integrand of interest.

The next step would be to extend to Bayesian regression and classification problem. For the regression case, an integral of interest could be a mixture model where the integral is not analytically obtainable or we cannot evaluate the normalised posterior distribution. As for classification, an immediate to problem to solve would be evaluating the predictive probability for a Gaussian process classification model (Rasmussen & Williams, 2005) but with different link functions. It would also be valuable to establish the regularity conditions and posterior concentration rates for such problems, laying out solid theoretical foundations for such applications.

So far we have assumed to be working with μ being a probability measure, and so this fits naturally into the application to statistical models, with the probability density function being the Radon-Nikodym derivative with respect to the Lebesgue measure. However, it would also be desirable to extend BART-BQ to the case of other measures μ such that $\mu(\mathbb{R}^d) \neq 1$, which could be used to solve non-probabilistic quadrature problems.

Furthermore, it would also valuable be extend BART to model a further class of distributions. For example, there is a need to develop tree-based models that fit into the generalised additive model (GAM) framework, where we could be dealing with exponential family distributions or even extreme value models, which could be valuable in applications such as football prediction models (Baio & Blangiardo, 2010) or rainfall modelling (Davison & Huser, 2015).

References

Baio, G. and Blangiardo, M. Bayesian hierarchical model for the prediction of football results. *Journal of Applied Statistics*, 37(2):253–264, 2010. doi: 10.1080/02664760802684177. URL https://doi.org/10.1080/02664760802684177.

Briol, F.-X., Oates, C., Girolami, M., and Osborne, M. A. Frank-wolfe bayesian quadrature: Probabilistic integra-

tion with theoretical guarantees. In *Advances in Neural Information Processing Systems*, pp. 1162–1170, 2015.

Bureau, U. C. 2017 American Community Survey PUMS data for Wyoming, 2017.

Chipman, H., Ranjan, P., and Wang, W. Sequential design for computer experiments with a flexible bayesian additive model. *Canadian Journal of Statistics*, 40(4):663–678, 2012. doi: 10.1002/cjs.11156. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cjs.11156.

Chipman, H. A., George, E. I., and McCulloch, R. E. BART: Bayesian additive regression trees. *Ann. Appl. Stat.*, 4 (1):266–298, 03 2010. doi: 10.1214/09-AOAS285. URL https://doi.org/10.1214/09-AOAS285.

Davison, A. and Huser, R. Statistics of extremes. *Annual Review of Statistics and Its Application*, 2(1):203–235, 2015. doi: 10.1146/annurev-statistics-010814-020133. URL https://doi.org/10.1146/annurev-statistics-010814-020133.

Durrett, R. *Probability: Theory and Examples*. Cambridge University Press, New York, NY, USA, 4th edition, 2010. ISBN 0521765390, 9780521765398.

Gelman, A. and Hill, J. *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press, 2006.

Genz, A. Testing multidimensional integration routines. In *Proc. Of International Conference on Tools, Methods and Languages for Scientific and Engineering Computation*, pp. 81–94, New York, NY, USA, 1984. Elsevier North-Holland, Inc. ISBN 0-444-87570-0. URL http://dl.acm.org/citation.cfm?id=2837.2842.

Hugh A. Chipman, E. I. G. . R. E. M. Bayesian cart model search. *Journal of the American Statistical Association*, pp. 935–948, 1998.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007. ISBN 0521880688, 9780521880688.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018. URL https://www.R-project.org/.

Rasmussen, C. E. and Ghahramani, Z. Bayesian monte carlo. In *Proceedings of the 15th International Conference on Neural Information Processing Systems*, NIPS'02, pp. 505–512, Cambridge, MA, USA, 2002. MIT Press. URL http://dl.acm.org/citation.cfm?id=2968618.2968681.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

Rockova, V. and Saha, E. On Theory for BART. *AISTATS 2019*, art. arXiv:1810.00787, October 2019.

Rockova, V. and van der Pas, S. Posterior concentration for bayesian regression trees and their ensembles. *arXiv preprint arXiv:1708.08734*, 2017.

Schürer, R. Parallel high-dimensional integration: Quasi-monte carlo versus adaptive cubature rules. In Alexandrov, V. N., Dongarra, J. J., Juliano, B. A., Renner, R. S., and Tan, C. J. K. (eds.), *Computational Science — ICCS 2001*, pp. 1262–1271, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. ISBN 978-3-540-45545-5.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, Jan 2016. ISSN 0018-9219. doi: 10.1109/JPROC.2015.2494218.

van der Vaart, A. W. and van Zanten, J. H. Rates of contraction of posterior distributions based on gaussian process priors. *The Annals of Statistics*, 36(3):1435–1463, 2008. ISSN 00905364. URL http://www.jstor.org/stable/25464673.