The Canadian Journal of Statistics
La revue canadienne de statistique

An adaptive sampling scheme guided by BART—with an application to predict processor performance

Author(s): Qingzhao YU, Bin LI, Zhide FANG and Lu PENG

Source: *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, Vol. 38, No. 1 (March/mars 2010), pp. 136-152
Published by: Statistical Society of Canada
Stable URL: http://www.jstor.org/stable/27805220
Accessed: 30-05-2017 13:49 UTC

The Canadian Journal of Statistics
Vol. 38, No. 1, 2010, Pages 136–152
La revue canadienne de statistique

An adaptive sampling scheme guided by BART—with an application to predict processor performance

Qingzhao YU[1]*, Bin LI[2], Zhide FANG[1] and Lu PENG[3]

[1]*Department of Biostatistics, School of Public Health, Louisiana State University Health Sciences Center, Suite 1400, 1615 Poydras Street, New Orleans, LA 70112*
[2]*Department of Experimental Statistics, Louisiana State University, Baton Rouge, LA 70803*
[3]*Department of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA 70803*

Key words and phrases: Adaptive design; Bayesian additive regression trees; sampling method; sequential G-optimization; processor performance.

MSC 2000: Primary 62L05 and 62D05; secondary 62F15.

Abstract: The evaluation of new processor designs is an important issue in electrical and computer engineering. Architects use simulations to evaluate designs and to understand trade-offs and interactions among design parameters. However, due to the lengthy simulation time and limited resources, it is often practically impossible to simulate a full factorial design space. Effective sampling methods and predictive models are required. In this paper, the authors propose an automated performance predictive approach which employs an adaptive sampling scheme that interactively works with the predictive model to select samples for simulation. These samples are then used to build Bayesian additive regression trees, which in turn are used to predict the whole design space. Both real data analysis and simulation studies show that the method is effective in that, though sampling at very few design points, it generates highly accurate predictions on the unsampled points. Furthermore, the proposed model provides quantitative interpretation tools with which investigators can efficiently tune design parameters in order to improve processor performance. *The Canadian Journal of Statistics* 38: 136–152; 2010 © 2010 Statistical Society of Canada

Résumé: L'évaluation de la conception de nouveaux processeurs est une étape importante en génie électrique et informatique. Les architectes utilisent des simulations afin d'évaluer les concepts et de comprendre les compromis et les interactions entre les différents paramètres du modèle de conception. Cependant, à cause de temps de simulation excessif et de la limitation des ressources, il est pratiquement impossible de simuler un devis factoriel complet. Des méthodes d'échantillonnage efficaces et des modèles de prédiction sont requis. Dans cet article, les auteurs proposent une approche automatique pour prédire la performance qui utilise un plan d'échantillonnage adaptatif interagissant avec le modèle prédictif pour choisir les échantillons lors de la simulation. Ces échantillons sont alors utilisés pour construire des arbres de régression bayésiens additifs qui sont à leur tour utilisés pour prédire l'ensemble de l'espace des devis. Des analyses de vraies données et des études de simulation ont montré que cette méthode est efficace. En effet, même si l'échantillonnage est fait sur très peu de points de devis, il génère des prédictions très précises sur les points non échantillonnés. De plus, le modèle proposé fournit des outils d'interprétation quantitatifs permettant aux chercheurs d'ajuster précisément les paramètres du devis afin d'améliorer les performances du processeur. *La revue canadienne de statistique* 38: 136–152; 2010 © 2010 Société statistique du Canada

1. INTRODUCTION

In computer engineering, a new design is usually evaluated by processor simulators, which provide a detailed insight into the performance, power consumption, and complexity of the

* *Author to whom correspondence may be addressed.*
E-mail: qyu@lsuhsc.edu

processor. A design space is composed of different combinations of quantitative and qualitative microarchitectural design factors such as processor frequency, issue width, cache size/latency, and branch predictor settings. The design space is often very large. It is practically infeasible to test the processor performance on every design point for the optimal processor design. This problem is becoming more challenging as more parameters are brought in by chip-multiprocessors (CMPs). Therefore, we need a method which could (1) efficiently select configurations for simulation; (2) use the simulation data to accurately predict processor performance of unsampled configurations; (3) specify factors that are important in predicting processor performance; and (4) identify the joint effects of parameters on processor performance. In this paper, we propose an adaptive sampling scheme, together with the tree-based Bayesian predictive modeling method, to explore the microarchitectural design space and predict processor performance. The method bridges the gap between simulation requirements and costs. The real examples and simulation studies show that this method effectively samples at very few design points but generates highly accurate predictions on the unsampled points. Furthermore, the model provides quantitative interpretation tools that help investigators understand factor effects and interactions on processor performance.

Some research has been done on selecting optimal designs and/or predicting the processor performances in large population spaces. Sacks et al. (1989) reviewed the special characters of computer experiments. Santner, Williams & Notz (2003) gave a survey on space-filling designs and some criterion-based experimental designs. Model-based adaptive designs are also developed. Cohn (1996) proposed an adaptive sampling algorithm based on neural network exploration. For this method, he tried to minimize the generalization error by completely exploring the design space. Kim & Ding (2005) developed an optimal engineering design guided by data-mining methods. Their method adopts feature functions, of which evaluation is computationally economical, as the surrogates for the design objective functions. A design library is generated based on the evaluation of feature functions and then a classification method is applied to create design selection rules. İpek et al. (2006) predicted the performance of memory subsystems, processors, and CMPs via artificial neural networks (ANNs). They combined neural networks and active learning methods to explore large design spaces. Lee & Brooks (2006) used regression models for processor performance and power prediction. Joseph, Vaswani & Thazhuthaveetil (2006) developed linear regression models which characterize the interactions between processor performance and microarchitectural parameters. These models were built via iterative processes directed by Akaike's information criteria (AIC). AIC or Bayesian information criterion (BIC) variable selection method requires pre-transformed variables and a set of interaction bases that have a reasonable linear relation to the response variable. Moreover, we show in this paper that neither AIC nor BIC is efficient in finding complex interactions. Li, Peng & Ramadass (2008) proposed a MART-aided adaptive sampling method. Two algorithms were used in their paper: (1) Multiple Additive Regression Trees (MART, see Friedman, 2001) for predictive model building and (2) adaptive sampling technique for selecting the unsampled points, based on which the MART model is most likely to be improved. Our method adopts Bayesian additive regression trees (BART) as the predictive model. Compared with MART, BART is a model-based approach which enables a full assessment of predictive uncertainty while remaining highly competitive in terms of predictive accuracy (Chipman, George & McCulloch, 2006). The assessment of predictive uncertainty is then utilized to guide the sampling procedures in our proposed method, which in turn lead to more efficient predictive models.

The proposed method includes three components: (1) the predictive modeling method, BART; (2) an active learning method which adaptively selects the most informative design points to improve predictive accuracy; and (3) interpretation tools for BART-fitted models which are used to evaluate the importance of design parameters in predicting the quantity of interest and shed light on the underlying working mechanism. With these components, this method provides a sampling technique and predictive approach to explore large design spaces, and thus could find its application in computer engineering and many other areas.

The article is organized as follows. In Section 2 we review BART model building and predictions and propose inference methods for BART. The BART-guided adaptive sampling (BGAS) as well as theoretical results are presented in Section 3. The implementation of the BGAS and comparisons with other methods are discussed in Section 4. Section 5 includes the conclusions and discussions.

2. BAYESIAN ADDITIVE REGRESSION TREES

BART is a Bayesian "sum-of-trees" model used to model the relationship between response and explanatory variables (Chipman, George & McCulloch, 2006). The method has shown excellent predictive performance. See, for example, Yu, MacEachern & Peruggia (2006), Zhang, Shih & Muller (2007) and literatures therein. BART employs two algorithms: classification and regression trees (CART) (Breiman et al. ,1984) and "boosting," which builds and combines a collection of models. BART is defined by a statistical model with priors and likelihoods. This model-based approach enables a full assessment of predictive uncertainty. In this section, we briefly review BART in terms of the tree model, the boosting algorithm, the prior selection, and the model fitting process. We review and propose BART inference methods at the end of the section.

2.1. Two Algorithms

CART is a binary recursive partitioning algorithm that provides a nonparametric alternative to traditional parametric models for regression and classification problems. Specifically, CART splits a multidimensional covariate space into two regions at each iteration. To do so, an optimal variable and a split point are selected by a comprehensive test on all variables and their realized values in the covariate space. The split continues on one or both of these subregions until some pre-specified stop rules are met. Then responses are modeled as a constant in each terminal region. Although CART represents information in a way that is intuitive and easy to be visualized, it is usually not as accurate as its competitors.

Boosting is one of the recent enhancements to tree-based methods that have met with considerable success in predictive accuracy. In boosting, models such as regression trees are fitted iteratively to the training data and appropriate methods are employed to put extra weights on observations modeled poorly by the current collection of trees.

MART is a special case of the generic gradient boosting approach on trees. Given n observations of the form $\{y_i, \mathrm{x}_i\}_1^n = \{y_i, x_{i1}, \ldots, x_{ip}\}_1^n$ and any differentiable loss function $L(y, F(\mathrm{x}))$, MART considers the problem of finding a function, $F(\mathrm{x})$, which predicts the response y by the input vector x, such that the expected loss, $E_{y,\mathrm{x}}\{L(y, F(\mathrm{x}))\}$, is minimized over the joint distribution of (y, x). Technically, MART approximates the target function $F(\mathrm{x})$ by an additive expansion of trees

$$\hat{f}(x) = \sum_{m=1}^{M} \nu b_H(\mathrm{x}; \gamma_m),$$

where M is the total number of trees; $b_H(\mathrm{x}; \gamma_m)$ is an H-terminal node tree (which partitions the input space into H-disjoint regions); γ_m is the parameter vector in the mth tree; and $\nu \in (0, 1]$ is the "shrinkage" parameter which controls the *learning rate* of the procedure. The M trees, $\{b_H(\mathrm{x}; \gamma_m)\}_{m=1}^M$, are built sequentially with response $y - \sum_{i=1}^{m-1} \nu b_H(\mathrm{x}; \gamma_i)$ and covariate vector x for $m = 1, \ldots, M$. Empirical results have shown (see, e.g., Friedman, 2001; Friedman

& Meulman, 2003) that smaller ν can often improve predictive accuracy. In practice, the number of trees M can be chosen by monitoring predictive performance on a validation set. We stop the modeling process when the reduction of the predictive error on the validation set becomes negligible. Note that H is related to the order of interactions considered in the model. For example, if $H = 2$, MART fits an additive model with no interactions. A detailed MART algorithm and related interpretation tools can be found in Friedman (2001).

2.2. BART Review

BART is a Bayesian version sum-of-trees model. BART differs from MART in several aspects: how it stochastically selects individual trees; how it weakens individual trees (controls the learning rate); and how it performs the iterative fitting by Bayesian backfitting on a fixed number of trees.

To build a tree, MART uses the CART algorithm. The tree stops growing when some preset rules, such as the total number of terminal nodes or the maximum length of the tree, are met. In BART, priors are set for each individual tree. These priors include: (i) the probability of each variable being a splitting variable at each interior node; (ii) the conditional distribution of the splitting rule assignment in each interior node, given the splitting variable; and (iii) the probability of a node at depth d being nonterminal.

Let $T_i (i = 1, 2, \ldots, m)$ be the ith tree consisting of a set of interior node decision rules and a set of b terminal nodes, and $M_i = \{\mu_{i1}, \mu_{i2}, \ldots, u_{ib}\}$ be the set of parameters associated with the corresponding terminal nodes in T_i. For an observation (y, x), denote $g(\mathrm{x}; T_i, M_i)$ as the ith Bayesian tree which assigns a $\mu \in M_i$ to x. That is, $E(y|\mathrm{x}, T_i) = \mu_{ij}$, if x falls in the jth terminal node, $j = 1, \ldots, b$. Technically, BART can be expressed as

$$y = f(\mathrm{x}) + \epsilon, \tag{1}$$

where $f(\mathrm{x}) = g(\mathrm{x}; T_1, M_1) + g(\mathrm{x}; T_2, M_2) + \cdots + g(\mathrm{x}; T_m, M_m)$ and $\epsilon \sim N(0, \sigma^2)$.

MART weakens the individual tree by a learning parameter ν while BART does so via setting priors. The parameters, μ, associated with each terminal node in a tree are assumed to be normally distributed with mean 0 and variance σ_μ^2. We pre-shift and rescale y so that it is very likely to be in (–0.5, 0.5). Then we set σ_μ^2 so that with a very high prior probability, the expected value of transformed y is in the interval (–0.5, 0.5). This prior has the effect of shrinking the tree parameters μ toward 0, limiting the effect of each individual tree component in (1) by keeping it small.

The total number of trees in a BART model is pre-specified. A novel feature of BART is that it employs a backfitting MCMC algorithm to collect samples from the induced posterior over the sum-of-trees model space (readers are referred to Hastie & Tibshirani, 1998, for details about the backfitting algorithm and Chipman, George & McCulloch, 2006, about BART). The sample can then be used for enhanced inference. For example, a single posterior mean estimate could be used to predict y given x. Moreover, pointwise uncertainty intervals for $f(\mathrm{x})$ are easily obtained by the corresponding quantiles, a property that is essential in our adaptive design scheme.

2.3. Inferences From BART

In this section, we review and propose methods for inferences based on BART. The application of these techniques to real data and simulation studies is presented in Section 4.

To assess the importance of each covariate in predicting the quantity of interest, we compare the relative mean square error (MSE) change when each covariate is omitted from the model fitting. Let $\mathrm{x}_{\setminus j}$ be the set of all covariates excluding the jth covariate; $\hat{f}_{\mathrm{x}}(\mathrm{x}_i)$ be the estimator of

y_i based on a model built on $\mathbf{x}$. The relative MSE is defined as

$$\text{RMSE}_j = \frac{\sum_{i=1}^{n} (y_i - \hat{f}_{\mathbf{x}_{\backslash j}}(\mathbf{x}_{i\backslash j}))^2/n}{\sum_{i=1}^{n} (y_i - \hat{f}_{\mathbf{x}}(\mathbf{x}_i))^2/n}, \quad j = 1, 2, \ldots, p, \tag{2}$$

where p is the number of covariates used to build the BART and $\{\mathbf{x}_{i\backslash j}\}_{i=1}^{n}$ are the data values of $\mathbf{x}_{\backslash j}$. RMSE_j approaching 1 means little difference in predictive error whether or not the jth covariate is used in model fitting. The larger the RMSE_j, the more important the jth covariate in fitting the final model.

Chipman, George & McCulloch (2006) proposed measuring the dependence of the fitted model on a subset of variables by the posterior partial dependence. Their method is derived from the partial dependence plots used to make inference on MART. Given $\tilde{f}(\mathbf{x})$, a simulation from the posterior sum-of-trees model space, and any subset $\mathbf{x}_{(s)}$ of the input variables indexed by $s \subset \{1, \ldots, p\}$, the corresponding partial dependence of $\tilde{f}(\mathbf{x})$ on $\mathbf{x}_{(s)}$ is

$$\tilde{F}_s(\mathbf{x}_{(s)}) = E_{\mathbf{x}_{\backslash s}}[\tilde{f}(\mathbf{x})],$$

where $E_{\mathbf{x}_{\backslash s}}[\cdot]$ is the expectation over the joint distribution of all the input variables excluding those in s. Then the partial dependence can be estimated by

$$\hat{F}_s(\mathbf{x}_{(s)}) = \frac{1}{N} \sum_{i=1}^{N} \tilde{f}(\mathbf{x}_i), \tag{3}$$

where $\mathbf{x}_i = (\mathbf{x}_{(s)}, \mathbf{x}_{i\backslash s})$ and $\{\mathbf{x}_{i\backslash s}\}_1^N$ are the data values of $\mathbf{x}_{\backslash s}$. Note that BART simulates from the posterior sum-of-trees model space. Therefore, we draw simulations from the posterior distribution of $\hat{F}_s(\mathbf{x}_s)$, from which we make inference on a single variable or a group of variables of interest.

3. PREDICTIVE MODEL GUIDED ADAPTIVE DESIGN

We first describe the proposed sampling scheme which combines the nonparametric tree-based predictive model BART with advanced sampling techniques to efficiently explore the microarchitectural design space. Then we justify the method in Section 3.2 and extend the sampling scheme to linear predictive models in Section 3.3.

3.1. BART Guided Adaptive Sampling

Adaptive sampling, also known as active learning in machine learning literature, involves sequential sampling schemes that use information gleaned from previous observations to guide the sampling process. Several empirical and theoretical studies have shown that samples selected adaptively outperform those obtained from conventional sampling schemes in learning a target function. See, for example, Freund et al. (1993), Sung & Niyogi (1995), Saar-Tsechansky & Provost (2001), and Li, Peng & Ramadass (2008). We propose the following sequential sampling algorithm:

ALGORITHM 3.1. *BART Guided Adaptive Sampling Algorithm*

```
(1) Randomly sample $n_1$ points from the design space.
(2) (a) Fit BART model with sample points and use the model to predict unsampled points.
    (b) Calculate the posterior predictive variance for all points: $V_j = \text{var}[f(\mathbf{x}_j)]; j = 1, \ldots, N$, where $N$ is the size of the design space.
    (c) Let $q = 0$, $V = \max_j(V_j)$,
```

(i) *Select $k = \arg\max_j V_j$ and let $q = q + 1$.*
(ii) *Calculate the posterior predictive correlation between the* jth *and* kth *design points, $\rho_{jk} = \mathrm{corr}(f(x_j), f(x_k))$, $j = 1, \ldots, N$.*
(iii) *Let $V_j = V_j \times (1 - \rho_{jk})$.*
(iv) *If $q < n_2$, go back to 2(c)i. Otherwise repeat Step 3.1(2) until stopping criterion is met.*

Remark 1. Generally we want n_1 and n_2 to be small for the best sampling and prediction. But n_1 should be large enough so that BART can be built on the initial n_1 sample points. If n_2 is large, it could speed the sampling process with a trade-off of model accuracy. We want n_2 to be large enough so that with the additional n_2 sample points, predictive variance of the newly built BART could be significantly reduced. Step 2(c)iv could also be "Go back to 2(c)i if $\max_j(V_j)/V > z$, otherwise repeat Step 2," where $z(< 1)$ is pre-specified to control the improvement of the predictive variance. In Section 4, we try different n_2 and find that if the model predictive performance is good, moderate change of n_2 will not have a large influence on the final predictive accuracy.

Remark 2. In general, we stop the procedure based on either the time/cost constraint or the convergence of some performance measure. The former is purely user-dependent. For the latter, we can monitor the procedure by a cross-validation measure or by the predictive performance on an independent test set. Since we consider the cases for which the stopping issue is potentially user-dependent, we pre-set the total sample size throughout the paper.

The rationale for the above sequential sampling is the bias–variance decomposition—note that the decomposition is originally proposed for the squared error loss, but it can be generalized to other losses such as the zero-one loss for classification. In practice, since the bias is unknown before measuring, we can only measure the predictive variance. To increase predictive accuracy, one should sample from design points with high predictive variances. However, clustered points tend to have similar predictive variances. In order to achieve global accuracy, we should select sampling points that are representative of the whole design space. Therefore, whenever a point is chosen, the other points that are highly correlated with this point have a lower chance of being selected. This corresponds to the assumption that the inclusion of a point in the model would result in greater decrease in predictive variance in those more correlated points.

3.2. Justification of the Adaptive Sampling

In this section, we heuristically justify the reason why we downweight the predictive variance for candidate design points by $1 - \rho$, where ρ is the predicted correlation coefficient between the chosen sample and other candidate design point.

We use the squared error to measure predictive accuracy. Let y and $\hat{y}$ be the true and predicted values, respectively. We want to minimize $\sum_{i=1}^{N} (y_i - \hat{y}_i)^2$. Note that in terms of Bayesian analysis, given data D, the posterior risk is $E[(y - \hat{y})^2|D] = E[(y - E(y))^2|D] + E[(\hat{y} - E(y))^2|D]$. The first term in the right-hand side is the posterior predictive variance (var(y|D)), and the second term is the posterior squared predictive bias. We focus on minimizing the posterior predictive variance because $E(y)$, and therefore the bias term, is unknown. The posterior predictive variance depends not only on the predictive model and the sampling method but also on the underlying population distribution. Assume that the true model is $y = f + \epsilon$, where f is a structure function of known information (e.g., the covariates $\mathbf{x}$), ϵ is the random error, and f and ϵ are independent. Then, $\mathrm{var}(y|D) = \mathrm{var}(f|D) + \mathrm{var}(\epsilon|D)$. Two assumptions are needed for Lemma 3.1: (1) if a design point is sampled, the posterior predictive variance for the structure part of the point reduces to zero, that is, $\mathrm{var}(f|D) = 0$. This assumption fits the deterministic nature of the response in processor experiments, in that when the design points are decided, the values of the response variable are also decided. Jones, Schonlau & Welch (1998) use linear models for computer experiments. Their model also satisfies this assumption. (2) For a Bayesian additive model, any two sampled points y_1 and y_2 can be decomposed into three parts: $y_1 = f_0 + f_1 + \epsilon_1$ and $y_2 = f_0 + f_2 + \epsilon_2$, where f_0 is the structure component shared by y_1 and y_2, f_1, and f_2 are the independent structure parts of y_1 and y_2 separately, and ϵ_1 and ϵ_2 are the random components. We further assume that f_0, f_1, f_2, ϵ_1, and ϵ_2 are independent.

Lemma 3.1. *Under the above assumptions and notations, if y_1 is sampled, the posterior predictive variance of y_2 is reduced to $\sigma_2^2 - \rho\sigma_1\sigma_2$, where ρ is the correlation coefficient between y_1 and y_2, σ_1^2 and σ_2^2 are the predictive variances of y_1 and y_2 separately before y_1 is sampled.*

The proof of Lemma 3.1 is in Appendix. In practice, the main goal of sampling is to select one sample from design points with large variances. After y_1 (with large σ_1^2) is selected, we want to select a second point y_2 which also has a large variance. This enables us to assume $\sigma_2^2 \approx \sigma_1^2$. Therefore, comparing $\sigma_2^2 - \rho\sigma_1\sigma_2$ is approximately equivalent to comparing $\sigma_2^2 - \rho\sigma_2^2$. To reduce the total posterior risk, we first sample a design point with the largest posterior predictive variance and then by Lemma 3.1, the posterior predictive variances for other candidate points are reduced to about $(1 - \rho)$ times the original variances. Note that when $y_2 = y_1$, σ_1 reduces to 0, which agrees with Assumption 1. If $\rho = 0$, the sampling of y_1 would not improve the predictive variance of y_2. Since $\rho < 1$, the posterior predictive variance of y_2 is always larger than or equal to 0 after y_1 is selected.

One of the benefits from BART is that BART produces a MCMC sample from the induced posterior over the sum-of-trees model space, which can readily be used to keep track of the uncertainty of prediction and to estimate the correlations among design points. Li, Peng & Ramadass (2008) use MART-guided design to improve prediction, where they use MART as the predictive model and sequentially sample design points that have the largest predictive variances while the minimum distance among each other is maximized. They have to use resampling method to estimate the predictive variance. However, they use the resampling method by Maximin distance design, which ignores the difference of the importance of each parameter on predicting the quantity of interest. If the information on the importance of parameters in prediction is available, it should be used adaptively to further guide the sampling process.

BART is chosen as the predictive model because it has great predictive performance if the true models are complicated. Even when the true model is linear, BART is also competitive. Note that our sampling scheme can be applied to any model which can estimate the uncertainty of prediction and the correlations among design points. We present a linear regression model example in Section 3.3.

3.3. Sequential Designs With Linear Regression Model

The sequential construction of optimal designs can also be used in combination with other predictive models. Different predictive model should have different sequential change of variance for a candidate point after a certain design point is selected. Lemma 3.2 considers a linear regression model: $y = \mathbf{x}\beta + \epsilon$, where $\epsilon \sim N(0, \sigma^2)$. Let $\mathbf{X}_0$ be the design matrix used to build the linear model, and $\mathbf{x}_i$ and $\mathbf{x}_j$ are design points not in $\mathbf{X}_0$. Denote

$$\rho_{ij} = \sqrt{\mathbf{x}_i'(\mathbf{X}_0'\mathbf{X}_0)^{-1}\mathbf{x}_j\mathbf{x}_j'(\mathbf{X}_0'\mathbf{X}_0)^{-1}\mathbf{x}_i[(\mathbf{x}_i'(\mathbf{X}_0'\mathbf{X}_0)^{-1}\mathbf{x}_i)(\mathbf{x}_j'(\mathbf{X}_0'\mathbf{X}_0)^{-1}\mathbf{x}_j)]^{-1}}$$

and $\sigma_i^2 = \sigma^2\mathbf{x}_i'(\mathbf{X}_0'\mathbf{X}_0)^{-1}\mathbf{x}_i$.

Lemma 3.2. *For linear regression models,*

(1) *if sampling at point* $\mathbf{x}_j$, *the predictive variance at design point* $\mathbf{x}_i$ *is reduced to* $\sigma_i^2 \times [1 - \rho_{ij}^2\sigma_j^2/(\sigma^2 + \sigma_j^2)]$, *where* ρ_{ij} *is the predictive correlation coefficient at* $\mathbf{x}_i$ *and* $\mathbf{x}_j$, σ_i^2 *and* σ_j^2 *are the predictive variances between* $\mathbf{x}_i$ *and* $\mathbf{x}_j$, *respectively, right before sampling at* $\mathbf{x}_j$*;*

(2) *sampling at design point* $\mathbf{x}_k = \arg\min_j\{\mathbf{x}_j'(\mathbf{X}_0'\mathbf{X}_0)^{-1}\mathbf{x}_j\}$ *would minimize the current maximum predictive variance.*

The proof of Lemma 3.2 is in Appendix. The lemma implies that for linear regression models, our sampling algorithm actually constructs a sequential G-optimal design.

4. EXAMPLES

We apply our method to a real computer architectural design study and two simulation studies. We find that in the architectural design study BGAS is more efficient than both the simple random sample (SRS) and the space-filling method, Maximin. We also find that BGAS performs better than the adaptive sampling method guided by MART. Furthermore, we use the real data to illustrate how to make statistical inference and to evaluate variable importance based on results from BGAS.

Two design spaces—one small and one large—are simulated to demonstrate the efficiency of our method. The small one is used to show why BGAS is an efficient method from the perspectives of adaptive sampling scheme and the superior predictive accuracy of BART over linear models. As to inference, our method shows its ability in detecting interactions. We use the large simulation to illustrate how to apply BGAS in large design spaces, which are more common in reality. In the study, randomly selected test data sets are used to compare BGAS with SRS and Maximin. Also, we select different n_2 to show that a moderate change in n_2 would not affect the sampling and therefore the predictive performance. It is more efficient to select multiple points without re-running the model.

4.1. A Computer Architectural Design Study

We first choose a relatively small design space of size 1,600 so that we can actually sample all the design points to evaluate the predictive accuracy of different methods. There are six parameters in this design space. Each parameter has 2, 4, or 5 levels, which results in 1,600 different designs. We then test the processor performance of the 1,600 different designs on two CPU benchmarks GCC and TWOLF from the Standard Performance Evaluation Corporation (SPEC) CPU 2000, which are widely used in the computer industry and academia to measure the bottlenecks and overall performance of the processor. The six parameters are L1CS, L1CBS, L1CA, L2CS, L2CBS, and L2CA. L1 and L2 are two level caches which store recently visited data and instructions. They are important to a processor's performance and power consumption. Here L1CS and L2CS represent L1 and L2 cache sizes separately. A cache is divided into many blocks. L1CBS and L2CBS denote the L1 and L2 cache block sizes, respectively. Usually a cache is divided into several small groups, each having a few blocks. This cache organization is so-called "set-associative" where each group is called as a set. The number of blocks in a group (or set) is "cache set associativity," which are recorded by L1CA and L2CA in this experiment.

For comparison, we use the predicted R^2, defined as

$$\mathrm{PR}^2 = \frac{\mathrm{ssto} - \mathrm{sse}}{\mathrm{ssto}} \times 100\%,$$

where $\mathrm{ssto} = \sum_{i=1}^{N}(y_i - \bar{y})^2$ is the total sum of squared errors before model fitting, with N being the size of the design space, y_i the observed processor performance, and $\bar{y}$ the mean performance; and $\mathrm{sse} = \sum_{i=1}^{N}(y_i - \hat{y}_i)^2$ is the sum of squared errors after model fitting, with

$\hat{y}_i$ being the predicted processor performance. The prediction is based on the model built by a small sample. We term PR^2 the predicted R^2 to distinguish it from the traditional R^2 which is used to measure the model "goodness-of-fit." In calculating PR^2, only a small proportion of the design space (at most 200 out of the 1,600 points in this example) are used for model fitting.

We start with 20 design points randomly selected from the design space and then sample 10 additional design points using BGAS each time until we have a sample of size 200. To compare BGAS, SRS, and Maximin, we use BART as the predictive model for all three sampling methods. In BART fitting, we skip the first 1,000 simulated sum-of-trees as burn-ins and then keep one from every four simulation. In this way, we obtain 1,000 simulations from the posterior model. During the process, we make sure that the simulation converges to a stationary distribution and the simulations are reasonably independent. For the Maximin method, we use the same distance function as that in Li, Peng & Ramadass (2008). We repeat all sampling methods 100 times. Table 1 gives the average sample sizes (of the 100 repetitions) needed to reach the critical predictive R^2 values in predicting the two process performance, GCC and TWOLF, respectively.

Table 1 shows that BGAS improves the predictive performance more quickly than SRS and Maximin in that the adaptive design needs a much smaller sample size to achieve the critical accuracy. Furthermore, it becomes much harder (requires a larger sample) for SRS and Maximin to reach higher PR^2 (say, e.g., PR$^2 \geq 0.97$).

We also compare BGAS with the adaptive sampling method guided by MART in Li, Peng & Ramadass (2008, denoted by LI). To make the methods comparable, we start LI with 20 Maximin samples and then adaptively choose 10 additional design points at each iteration until a total of 200 samples are selected. The process is repeated 100 times. Figure 1 compares all four methods in terms of the mean predicted PR^2 of the 100 repetitions. It is clear that BGAS is superior to SRS, Maximin, and LI. Note that both Maximin and LI start with 20 Maximin samples, but LI uses MART while Maximin uses BART as predictive models. We also combine LI with random forest (RF) as a predictive model for a comparison.

We obtain box plots of the relative MSE in Equation (2) for the six covariates in Figure 2. The locations of these box plots indicate the importance of the six covariates in predicting GCC

TABLE 1: The average sample size needed to get the critical predictive R^2.

	PR^2							
	0.80	0.85	0.90	0.95	0.96	0.97	0.98	0.99
GCC performance								
SRS	50	60	70	100	110	120	150	> 200
Maximin	50	60	70	100	130	140	180	> 200
BGAS	50	50	60	80	90	100	110	140
TWOLF performance								
SRS	30	30	40	90	110	130	160	> 200
Maximin	30	30	50	100	120	160	190	> 200
BGAS	30	30	40	80	100	120	140	170

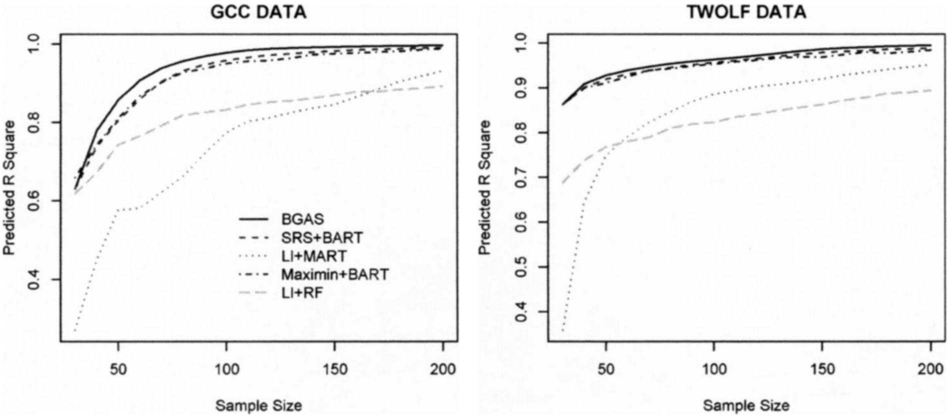

FIGURE 1: For the real data set, comparison of BGAS with SRS, Maximin, and LI. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

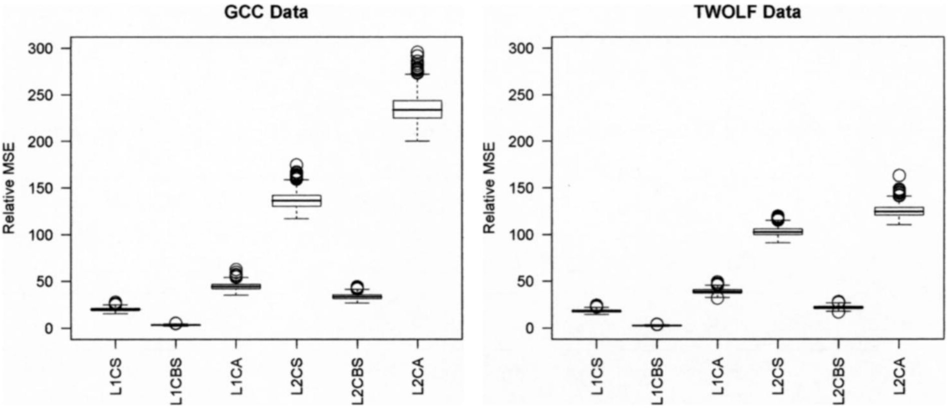

FIGURE 2: For the real data set, the box plots of RMSE of six covariates in predicting the processor performance.

and TWOLF. We find that "L2CA" is the most important parameter in predicting both processor performances, though in predicting TWOLF, the difference between "L2CA" and"L2CS" is not significant (the two 95% confidence intervals are overlap).

In Figure 3, we obtain the partial dependence plots for GCC data to show how the processor performance changes with different settings of each covariate, where the x-axis shows the values of the corresponding covariate and y-axis is the partial dependence. We can learn the marginal effect of each covariate on the GCC from the plot. For example, we observe that the marginal effect of GCC decreases when the level of *L2CA* increases. We can also use the partial dependence plot on a subset of covariates (not shown) to check the joint effect of covariates on processor performance. We do not find any important interactions in this data set.

For GCC data, Table 2 shows the average final sampling results (of the 200 × 100 samples) for each covariate based on BGAS, SRS, and Maximin separately. In the table, each cell shows the percentage of times that the corresponding level of the covariate is selected. By SRS and Maximin, all levels of a covariate have about the same chance of being selected. BGAS chooses samples differently and thus results in better predictions as demonstrated in Table 1.

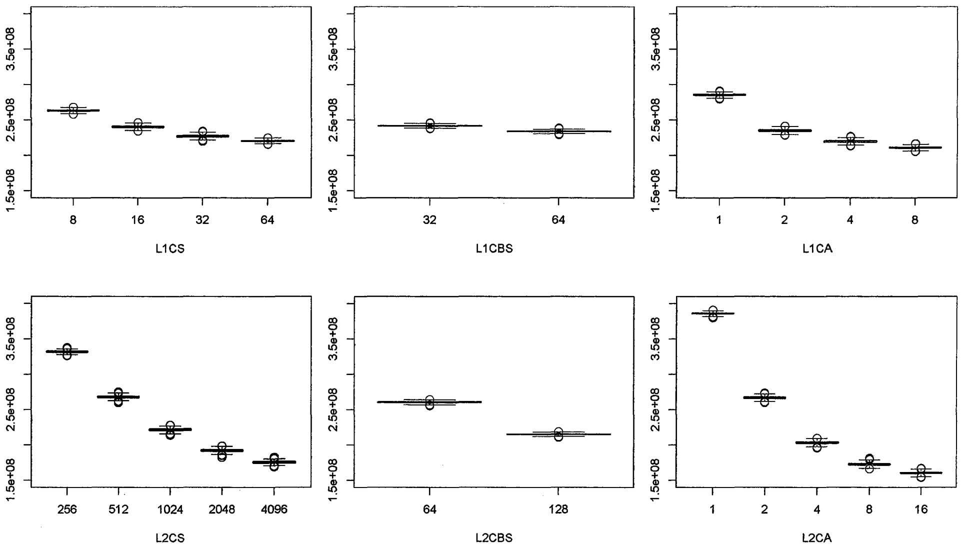

FIGURE 3: For the real data set, partial dependence plots of each covariate in predicting the GCC.

TABLE 2: Proportion of the corresponding level of each covariate being selected in the sample.

	Covariates										
	L1CBS		L1CA				L2CS				
	32	64	1	2	4	8	256	512	1,024	2,048	4,096
BGAS	50	50	32	23	20	25	27	19	17	17	20
SRS	50	50	26	25	25	25	20	20	20	19	20
Maximin	50	50	25	25	25	25	20	20	21	21	19
	L2CBS		L1CS				L2CA				
	64	128	8	16	32	64	1	2	4	8	16
BGAS	52	48	31	21	21	27	35	19	15	14	18
SRS	50	50	26	25	25	25	20	21	20	19	20
Maximin	50	50	26	25	25	25	20	20	20	20	20

4.2. Simulations

4.2.1. Simulation 1

We have shown in Section 4.1 from a real data set that BGAS could improve predictions with a small sample size. In this section, we use a simulation to investigate the source of the improvement: the BART predictive model and/or the adaptive sampling. We also show how BART can help in identifying important interactions.

We have five covariates, each of which has six levels $(0, 0.2, 0.4, 0.6, 0.8, 1)$. Thus, the design space is composed of 7,776 design points. The response variable is simulated from the

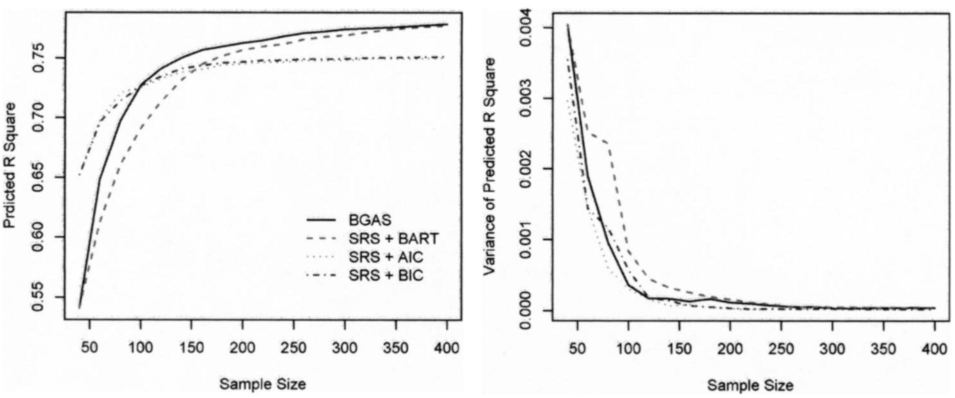

FIGURE 4: For Simulation 1, comparison of adaptive experimental design with SRS and BART with linear regression model. Left plot shows the mean predictive R^2 of each method as the sample size increases. Right plot shows the variance of predictive R^2 of the 20 repetitions. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

equation $y = 9e^{-3(1-x_1)^2}e^{-3(1-x_2)^2} - 0.65e^{-2(x_3-x_4)} + 2\sin^2(x_5\pi) + e$, where e is the normally distributed random error with mean 0 and signal to noise ratio of 4 (the variance is approximately 2.06).

We start to sample at 40 points and add 20 points at each iteration until we collect a total of 400 samples, about 5% out of the 7,776 design points. We repeat the process 20 times to account for the randomness in initial sampling and the MCMC process. BGAS design is compared with SRS, both using BART as the predictive model. Furthermore, BART is compared with the linear models chosen by AIC and BIC, and all models are built on randomly selected samples. The left plot in Figure 4 shows the mean PR^2 of the 20 repetitions. The right plot shows the corresponding variance of the PR^2.

From Figure 4, we see that the AIC and BIC model selection criteria outperforms BART in model building and predictions when the sample size is very small. As sample size increases, BART shows much better predictive performance. BGAS consistently outperforms SRS. Moreover, the predictive variance for BGAS is consistently smaller than that of SRS with BART. BART combined with BGAS has approximately the same predictive variance as the linear regression models.

From the simulation formula, there should be interaction effects between x_1 and x_2, and between x_3 and x_4. The effect of x_5 is independent from the other predictors. We check whether BART or linear regression models can detect the correct interaction terms. Figure 5 shows some dependence plots of the response on the combination of two covariates. The dependence plots are based on one of the final sampling results from BGAS. All plots from the 20 repetitions look similar. We observe that BART successfully identifies the interactions between x_1 and x_2, and x_3 and x_4. Note that one can also use standard statistical tests to check whether the interactions are significant based on the predictions of the model. The AIC and BIC modeling automatically chooses variables from all main effects, square terms, and all two-factor interactions. The model with optimal AIC or BIC score is chosen as the best model. Note that only hierarchical models are considered by AIC and BIC. Table 3 includes the number of times, out of 20 repetitions, that each term is selected by AIC/BIC. AIC detects the x_1x_2 interaction five times while BIC does so only once. Neither AIC nor BIC detects x_3x_4 interaction. This indicates that BART outperforms AIC and BIC model selection criteria in detecting interactions.

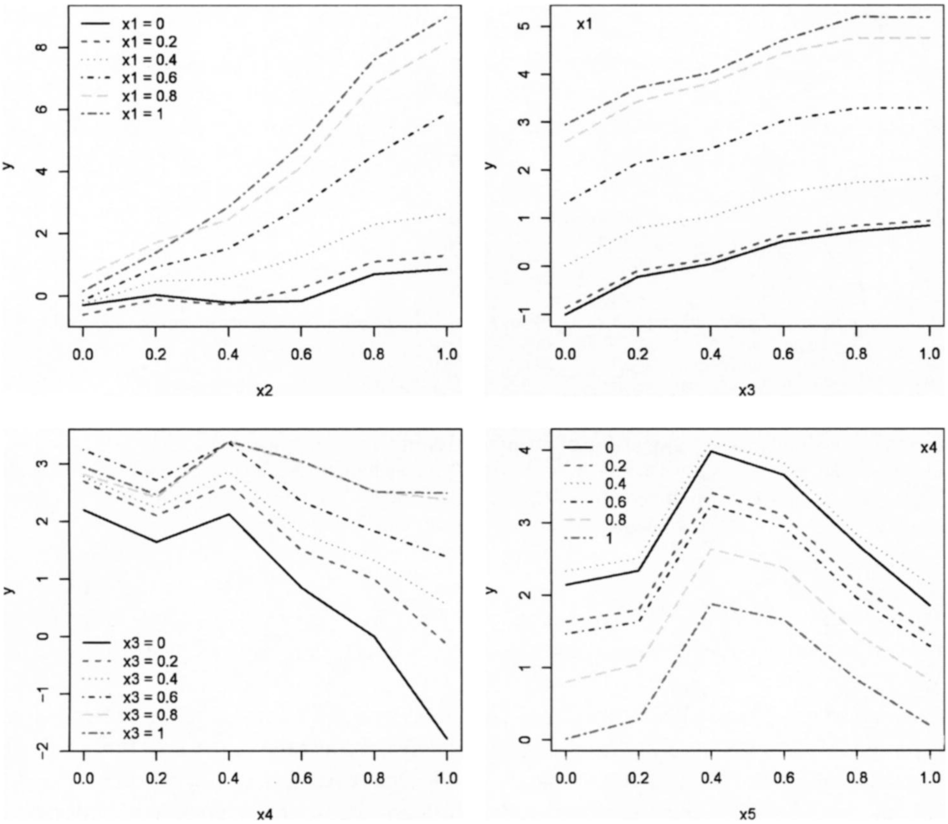

FIGURE 5: For Simulation 1, partial plots of subsets of the response y on the combination of two covariates to check for possible interactions in the BART fitted model. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

TABLE 3: The count that each term is selected in the model by AIC or BIC out of the 20 repetitions in Simulation 1.

	Covariates									
	x_1	x_2	x_3	x_4	x_5	x_1x_2	x_1x_3	x_1x_4	x_1x_5	x_2x_3
AIC	5	20	20	20	20	5	0	2	1	1
BIC	1	20	20	20	20	1	0	0	0	0
	x_1^2	x_2^2	x_3^2	x_4^2	x_5^2	x_2x_4	x_2x_5	x_3x_4	x_3x_5	x_4x_5
AIC	2	15	0	0	20	0	2	0	0	0
BIC	0	19	0	0	20	0	0	0	0	0

4.3. Simulation 2

In BGAS, we need to calculate the predictive variances of unsampled design points and the correlation coefficients between the selected design point and all the unsampled points in order to select next samples. When the design space is huge, it might become unrealistic to select samples from the whole design space because of the calculation burden. In Simulation 2, we illustrate how to implement BGAS and evaluate its performance when the design space is too large to sample from. Also, we use different n_2 in Algorithm 3.1 to show how the choice of n_2 affects predictive performance. We first use simple random sampling to choose a proportion of all design points as candidate points and then use BGAS to sample from the candidate design space. For the simulation, we modify the formula in Simulation 1 so that there are seven covariates and each variable has five levels. The simulation function is

$$y = 9e^{-3(1-x_1)^2}e^{-3(1-x_2)^2}e^{-3(1-x_3)^2} - 0.65e^{-2(x_4-x_5)} + 2\sin^2(x_6\pi) - 1.77x_7 + e,$$

where e is the normally distributed random error with signal to noise ratio of 4 (here, the variance is approximately 0.59). Thus, there are a total of 78,125 design points. In the simulation, we start with 30 design points. We randomly choose 8,000 design points to form a candidate design space from which we apply BGAS each time for an additional 5, 10, 15, 20, 30, or 50 design points until we get at least 200 sample points. At each iteration, we randomly select another 5,000 sample points as the test set to evaluate the predictive performance of the models. The process is repeated 30 times to account for randomness.

The upper panel of Figure 6 compares BGAS, SRS, and Maximin with BART as the predictive model. In BGAS, we choose $n_2 = 10$. The upper left plot shows the means of predictive R^2 and the

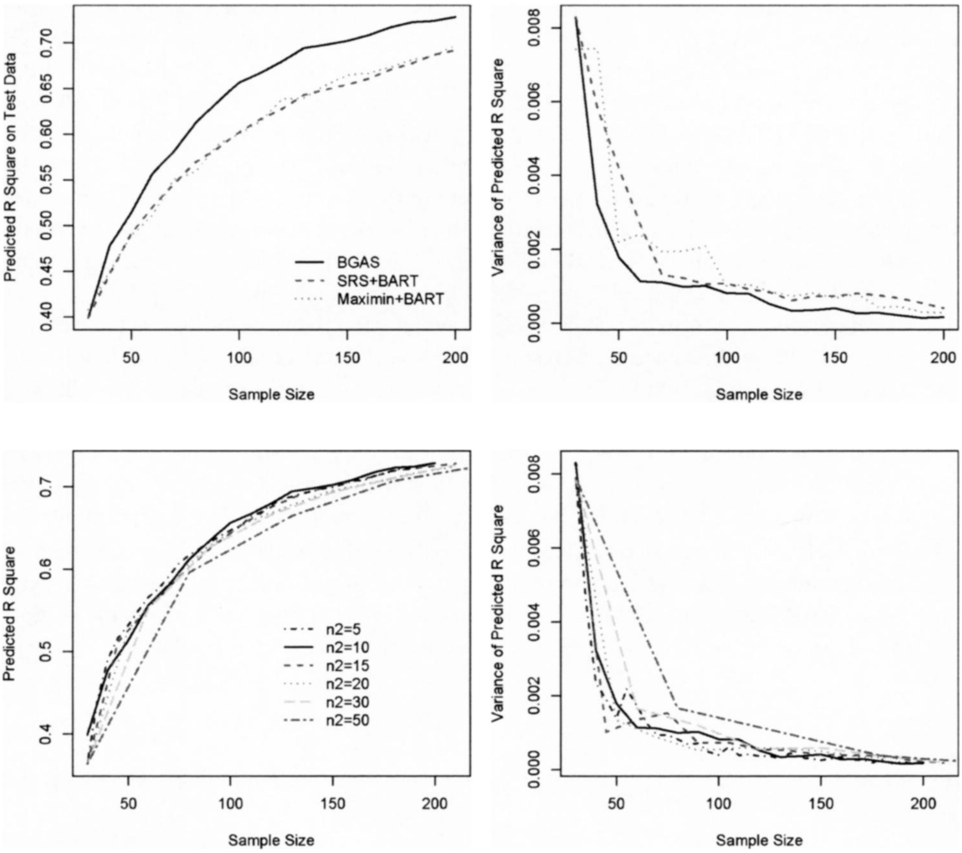

FIGURE 6: For Simulation 2, upper panel compares BGAS with SRS and Maximin, lower panel compares BGAS with different n_2s. Left panel shows the mean predictive R^2 of each method as the sample size increases. Right panel shows variance of the predictive R^2 of 30 repetitions. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

upper right plot shows the corresponding variances. We find that BGAS uniformly outperforms SRS and Maximin by showing higher predictive R^2 but lower variances of these R^2. The lower panel of Figure 6 compares predictive performance when n_2 is chosen at different values. One can see that the differences in PR^2 and their variances are negligible when n_2 ranges from 5 to 30. These changes become significant when n_2 is 50, where PR^2 is lower with a higher variance.

Note that theoretically, there should be a little gain in PR^2 or predictive accuracy when n_2 is small. But a smaller n_2 results in a greater number of times to run the BGAS algorithm and to initiate a new simulation process. To draw the same number of simulations, the time to run the BGAS algorithm and to initiate a new simulation process is approximately inversely proportional to n_2. In designing the BGAS algorithm, we try to best approximate the posterior variance so that we can choose multiple design points without rerunning models or incurring too much error. As in this simulation, the predictive error is approximately the same when n_2 is between 5 and 30. In practice, we can set up a cost function on both time and the loss of predictive accuracy when $n_2 > 1$, then solve for the best n_2.

5. CONCLUSIONS AND DISCUSSSIONS

Architectural design space exploration has recently become very challenging because of the large number of parameters introduced by advanced circuit integration technology. In this paper we propose an adaptive design scheme with BART being the predictive model, aiming to efficiently sample a small proportion of the design space with high predictive accuracy. Other predictive models that can estimate the uncertainty of prediction well can also be used in this sampling scheme. Compared with other methods in the literature, BART has the following advantages: (1) BART stochastically searches the model space and provides a simulation from the posterior distributions of interest, which can be readily used in the adaptive design; (2) tree-based methods are particularly well suited for the discrete (either ordinal or nominal variables) design space parameters; (3) BART can achieve extremely accurate predictions which has been proven empirically and theoretically; (4) BART is highly robust with regard to the tuning parameter values so that the practitioners need minimal knowledge to tune the model; (5) BART also comes with model interpretation tools which can help us understand the underlying mechanism. In this paper, we demonstrate the success of BGAS using a real data set and two simulation studies. Other statistical modeling methods such as Gaussian Process regression (Seo et al., 2000) and Treed Gaussian Process Models (Gramacy & Lee, 2008) are designed especially for computer experiments. As a future field of research, we will explore the possibility of using the Gaussian process at the tree terminal nodes in BART. And then the proposed sampling scheme could be used with the new model. Most likely, the predictive performance could be further improved.

The proposed method can also be used to find optimal designs from huge spaces. Instead of randomly selecting candidate design points, we choose design points that are predicted of extreme values. Since BART automatically provides a posterior interval prediction, we could sample all the points whose posterior intervals overlap that of the predicted extreme value.

APPENDIX

Proof of Lemma 3.1. Before y_1 is sampled, by Assumption 2, the predictive variance for y_1 and y_2 is

$$\begin{aligned} \mathrm{var}(y_1) &= \mathrm{var}(f_0) + \mathrm{var}(f_1) + \mathrm{var}(\epsilon_1) = \sigma_1^2 \\ \mathrm{var}(y_2) &= \mathrm{var}(f_0) + \mathrm{var}(f_2) + \mathrm{var}(\epsilon_2) = \sigma_2^2 \\ \mathrm{cov}(y_1, y_2) &= \mathrm{var}(f_0) = \rho\sigma_1\sigma_2 \end{aligned}$$

After y_1 is observed, we have $\text{var}(f_0) = 0$ by Assumption 1, while the posterior variance $\text{var}(f_2)$ and $\text{var}(\epsilon_2)$ would not change because of their independence to y_1. Thus, the posterior variance for y_2 becomes $\text{var}(f_2) + \text{var}(\epsilon_2) = \sigma_2^2 - \text{var}(f_0) = \sigma_2^2 - \rho\sigma_1\sigma_2$. ■

Proof of Lemma 3.2.

1. For the linear model $y = \mathbf{x}\beta + \epsilon$, where $\epsilon \sim N(0, \sigma^2)$. Assume $\mathbf{X}_0$ is the design matrix used to build the linear model, and $\mathbf{x}_j$ is a design point not in $\mathbf{X}_0$. Then after $\mathbf{x}_j$ is sampled, the predictive variance for design point $\mathbf{x}_i$ becomes

$$\begin{aligned}\text{var}_i &= \mathbf{x}_i'(\mathbf{X}_0'\mathbf{X}_0 + \mathbf{x}_j\mathbf{x}_j')^{-1}\mathbf{x}_i\sigma^2\\ &= [\mathbf{x}_i'(\mathbf{X}_0'\mathbf{X}_0)^{-1}\mathbf{x}_i - \mathbf{x}_i'(1 + \mathbf{x}_j'(\mathbf{X}_0'\mathbf{X}_0)^{-1}\mathbf{x}_j)^{-1}(\mathbf{X}_0'\mathbf{X}_0)^{-1}\mathbf{x}_j\mathbf{x}_j'(\mathbf{X}_0'\mathbf{X}_0)^{-1}\mathbf{x}_i]\sigma^2\\ &= \mathbf{x}_i'(\mathbf{X}_0'\mathbf{X}_0)^{-1}\mathbf{x}_i\sigma^2\left[1 - \frac{\mathbf{x}_i'(\mathbf{X}_0'\mathbf{X}_0)^{-1}\mathbf{x}_j\mathbf{x}_j'(\mathbf{X}_0'\mathbf{X}_0)^{-1}\mathbf{x}_i}{(\mathbf{x}_i'(\mathbf{X}_0'\mathbf{X}_0)^{-1}\mathbf{x}_i)(1 + \mathbf{x}_j'(\mathbf{X}_0'\mathbf{X}_0)^{-1}\mathbf{x}_j)}\right]\\ &= \sigma_i^2\left[1 - \rho_{ij}^2 \times \frac{\sigma_j^2}{\sigma^2 + \sigma_j^2}\right].\end{aligned}$$

2. Let $\mathbf{x}_k = \arg\min_j\{\mathbf{x}_j'(\mathbf{X}_0'\mathbf{X}_0)^{-1}\mathbf{x}_j\}$. After $\mathbf{x}_j$ is sampled, the predictive variance for $\mathbf{x}_k$ becomes

$$\begin{aligned}\text{var}_k &= \sigma_k^2\left[1 - \rho_{kj}^2 \times \frac{\sigma_j^2}{\sigma^2 + \sigma_j^2}\right]\\ &\geq \sigma_k^2\left[1 - \frac{\sigma_j^2}{\sigma^2 + \sigma_j^2}\right]\\ &= \frac{\sigma^2\sigma_k^2}{\sigma^2 + \sigma_j^2} \geq \frac{\sigma^2\sigma_k^2}{\sigma^2 + \sigma_k^2}.\end{aligned}$$

■

ACKNOWLEDGEMENTS

The authors thank the editor, associate editor, and referees for constructive comments and suggestions that helped to improve the presentation of the paper.

BIBLIOGRAPHY

L. Breiman, J. H. Friedman, R. Olshen & C. Stone (1984). "*Classification and Regression Trees*." Wadsworth, Pacific Grove.

H. A. Chipman, E. I. George & R. E. McCulloch (2006). BART: Bayesian additive regression trees. *Technical Report*, University of Chicago, http://arxiv.org/abs/0806.3286v1.

D. A. Cohn (1996). Neural network exploration using optimal experiment design. *Neural Network*, 9, 1071–1083.

J. H. Friedman (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29, 1189–1232.

J. H. Friedman & J. J. Meulman (2003). Multiple additive regression trees with application in epidemiology. *Statistics in Medicine*, 22, 1365–1381.

Y. Freund, H. S. Seung, E. Shamir & N. Tishby (1993). Information, prediction, and query by committee. *Proceedings of the Advances in Neural Information Processing Systems*, pp. 483–490.

R. B. Gramacy & H. K. H. Lee (2008). Bayesian Treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 403, 1119–1130.

T. Hastie & R. Tibshirani (1998). Bayesian backfitting. *Statistical Science*, 15, 193–223.

E. İpek, S. A. McKee, B. R. Supinski, M. Schulz & R. Caruana (2006). Efficiently exploring architectural design spaces via predictive modeling. *Twelfth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS XII)*, San Jose, CA.

D. R. Jones, M. Schonlau & W. J. Welch (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13, 455–492.

P. Joseph, K. Vaswani & M. Thazhuthaveetil (2006). Use of linear regression models for processor performance analysis. *Proceedings of the 12th IEEE Symposium on High Performance Computer Architecture (HPCA-12)*, pp. 99–108.

P. Kim & Y. Ding (2005). Optimal engineering system design guided by data-mining methods. *Technometrics,* 47, 336–348.

B. Lee & D. Brooks (2006). Accurate and efficient regression modeling for microarchitectural performance and power prediction. *Twelfth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS XII)*, San Jose, CA, October 2006.

B. Li, L. Peng & B. Ramadass (2008). Efficient MART-aided modeling for microarchitecture design space exploration and performance prediction. *2008 ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 2008)*, Annapolis, MD.

M. Saar-Tsechansky & F. Provost (2001). Active learning for class probability estimation and ranking. *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pp. 911–920.

J. Sacks, W. J. Welch, T. J. Mitchell & H. P. Wynn (1989). Design and analysis of computer experiments. *Statistical Science*, 4, 409–435.

T. J. Santner, B. J. Williams & W. I. Notz (2003). "*The Design and Analysis of Computer Experiments.*" Springer Verlag, New York.

S. Seo, M. Wallat, T. Graepel & K. Obermayer (2000). Gaussian process regression: Active data selection and test point rejection. *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, IEEE, Springer-Verlag, London, pp. 241–246.

K. Sung & P. Niyogi (1995). Active learning for function approximation. *Proceedings of the Advances in Neural Information Processing Systems*, 7, 593–600.

Q. Yu, S. N. MacEachern & M. Peruggia (2006). Bayesian synthesis. *Technical Report #773*, Department of Statistics, The Ohio State University.

X. Zhang, T. Y. Shih & P. Muller (2007). A spatially-adjusted Bayesian additive regression tree model to merge two datasets. *Bayesian Analysis,* 2 (3), 611–634.

Received 28 May 2009
Accepted 30 August 2009