# Model Guided Adaptive Design and Analysis in Computer Experiment

**Qingzhao Yu[1], Bin Li[2]*, Zhide Fang[1] and Lu Peng[3]**

[1]*Biostatistics Program, School of Public Health, Louisiana State University Health Sciences Center, New Orleans, LA, USA*

[2]*Department of Experimental Statistics, Louisiana State University, Baton Rouge, LA, USA*

[3]*Department of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA, USA*

Received 8 March 2011; revised 18 June 2011; accepted 22 June 2012
DOI:10.1002/sam.11156
Published online 21 August 2012 in Wiley Online Library (wileyonlinelibrary.com).

**Abstract:** Computer experiments have become increasingly important in several different industries. These experiments save resources by exploring different designs without necessitating real hardware manufacturing. However, computer experiments usually require lengthy simulation times and powerful computational capacity. Therefore, it is often pragmatically impossible to run experiments on a complete design space. In this paper, we propose an adaptive sampling scheme that interactively works with predictive models to sequentially select design points for computer experiments. The selected samples are used to build predictive models, which in turn guide further sampling and predict the entire design space. For illustration, we use Bayesian additive regression trees (BART), multiple additive regression trees (MART), treed Gaussian process and Gaussian process to guide the proposed sampling method. Both real data and simulation studies show that our sampling method is effective in that (i) it can be used with different predictive models; (ii) it can select multiple design points without repeatedly refitting the predictive models, which makes parallel simulations possible and (iii) the predictive model built on its generated samples gives more accurate predictions on the unsampled points than the models built on samples from other methods such as random sampling, space-filling designs and some adaptive sampling methods. © 2012 Wiley Periodicals, Inc. Statistical Analysis and Data Mining 5: 399–409, 2012

**Keywords:** adaptive design; Bayesian additive regression trees; computer experiment; Gaussian process; sequential sampling

## 1. INTRODUCTION

In the automotive, semiconductor, computer engineering, and many other industries, there is a growing emphasis on designing new products by computer experiments [1]. These experiments facilitate exploring different designs without real hardware manufacturing. For example, in computer engineering, a new processor design is usually evaluated by processor simulators, programs that simulate the running behaviors and provide detailed insights into important benchmarks of the processor, such as its performance, power consumption and complexity. However, the computer experiments for each design point usually require long simulation time and powerful computational capacity and the design space is often composed of huge number of design points. For example, a processor design space is composed of different combinations of quantitative and qualitative micro-architectural design factors such as processor frequency, issue width, cache size/latency and branch predictor settings. Owing to practical constraints, it is infeasible to test every design point for the optimal design, so we need a sampling method that can efficiently select an appropriately small number of design points on which statistical models could be built to (i) accurately predict the performance of unsampled design points and (ii) identify the factors and interactions that are important in effecting the performance of interest. Moreover, instead of sequentially choosing one design point, we select multiple design points so that several computers can be used for simultaneous simulations. In this paper, we propose a predictive model guided adaptive sampling (MGAS) scheme. The sampling method selects many design points sequentially and can be used with any predictive models. It bridges the gap between simulation requirements and limited resources. Real-life examples and simulation studies show that this method samples at very few design points

*Correspondence to:* Bin Li (bli@lsu.edu)

yet generates highly accurate predictions of the unsampled points.

The background literature for our method is based on two lines of research. The first line is on the predictive models. A significant amount of work has been done on predicting the quantities of interesting in large population spaces. For example, İpek [2] predicted the performance of memory subsystems, processors and chip multiprocessors (CMPs) via artificial neural networks. They combined neural networks and active learning methods to explore large design spaces. Lee and Brooks [3] applied regression models to processor performance and power prediction. Joseph *et al*. [4] developed linear regression models which characterize the interactions between processor performance and micro-architectural parameters. These models were built via iterative processes guided by Akaike's information criteria (AIC). The 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. The Gaussian process (GP) [5] is popular in computer experiment analysis since the process accounts for the deterministic property in computer experiments. Yu *et al*. [6] proposed using BART in computer experiments, for it can catch important nonlinear effects and interactions. As we observed in practice, different predictive models are preferable in different situations. Therefore, a good sampling method should be readily adaptable with any predictive models.

The second line of background research concerns effective sampling methods that select a small sample of points for simulation, based on which effective predictive models can be built. Sacks *et al*. [7] reviewed the special characteristics of computer experiments. Santner *et al*. [8] presents a survey on space-filling designs and some criterion-based experimental designs. Model-based adaptive designs are also developed. For instance, MacKay [9] aimed at minimizing the predictive variance by selecting the sample points with maximum variance. Cohn [10] proposed an adaptive sampling algorithm via neural network exploration, in which he minimized the generalized error by completely exploring the design space. Seo *et al*. [5] demonstrated that both Mackay's and Cohn's methods performed well in accelerating and improving learning. Kim and Ding [11] developed an optimal engineering design guided by data-mining methods. Their method adapted feature functions, of which evaluation was computationally economical, as the surrogates for the design objective functions. A design library was generated by evaluating the feature functions, and then a classification method was applied to create design selection rules. Gramacy and Lee [12,13] proposed to use the GP at local regions split by a tree model and then to measure the predictive uncertainty to guide subsequent experimental runs.

Our work has its roots in ref. 6, where the authors proposed a BART-aided adaptive sampling method. In this paper, we generalize the active learning process, so that the proposed sampling method can be applied with general Frequentist or Bayesian predictive models, giving us the flexibility to choose models that have good predictive performances and/or can provide quantitative interpretation tools that help investigators understand the factors and their interaction effects on the quantity of interest. When BART is the predictive model, the proposed method reduces to the method described in ref. 6, except that the theories in this paper are more precise with fewer approximations in sequentially estimating the change in predictive variances when a new design point is added. This paper develops a theoretical framework to evaluate predictive uncertainties of unsampled design points and their changes due to additional design points. This enables us to consider the associations among design points when selecting multiple design points, without resampling or refitting the predictive model. The proposed sampling method is especially useful in an asynchronous parallel environment, where several computing agents are used independently for computer experiments.

The article is organized as follows. In Section 2, we present the MGAS scheme. In Section 3 we develop the theoretical framework and generalize the sampling method so that it can be guided by more general predictive processes. The implementation of the general model guided adaptive sampling (GMGAS) method and its comparisons with other methods are discussed in Section 4. Section 5 includes the conclusions and further research topics.

## 2. MODEL 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, refs 6, 14–17 for more discussions.

The purpose of adaptive sampling is to actively select samples, which are used to build models with good predictive accuracy. Usually, mean squared error is used to measure predictive accuracy. Let $y$ and $\hat{y}$ be the true and predicted values respectively for the quantity of interest. We want to minimize $E[(y-\hat{y})^2]$, where the expectation is over the training and testing data. Note that the expected error can be decomposed as

$$E[(y-\hat{y})^2] = E[(y-E(y))^2] + [E(\hat{y})-E(y)]^2 + E[(\hat{y}-E(\hat{y}))^2]. \quad (1)$$

The first term in the right-hand side of Eq. (1), denoted as $\sigma^2$, is the variance of random errors, invariant to which training and testing data sets are used. Many computer experiments are deterministic, i.e. $\sigma^2 = 0$. That is, repeated sampling at the same design point will generate the same response. The second term of Eq. (1) is the squared predictive bias. Since $E(y)$ is unknown, usually we do not know the bias. The third term is the predictive variance, $\text{var}(\hat{y})$, which highly depends on how the training data sets are formed, i.e. the sampling scheme for the training data. In this paper, we focus on choosing a sampling method to minimize the predictive variance.

There are two popular active learning schemes. The first method, suggested by MacKay [9], noted as ALM), aims to select the design point where the predictor exhibits maximum variance. The second method, proposed by Cohn [10], noted as ALC), measures the reduction in predictive variance averaged over the design space and chooses the point that maximizes the reduction. Both methods are efficient, as shown by Seo *et al*. [5]. Our method intends to adaptively sample at more than one design point thereby enabling parallel experiments, and saving time and computational resources for initiating new sampling processes and building new predictive models. The problem we encountered is that the design points are usually highly correlated and therefore tend to have similar predictive variances. In order to achieve global accuracy, we should select sampling points that well represent the entire design space. It is usually more efficient to sample at the uncorrelated design points with relatively high predictive variances than at design points with the highest predictive variances but highly correlated. Therefore, whenever a point is chosen, the points that are highly correlated with it should have a decreased chance of being selected. This is reasonable in that the information of a design point could greatly reduce the predictive variances of other design points highly correlated with it. We propose the following sequential sampling algorithm, Algorithm 2, that actively estimates the change in predictive variances of unsampled design points when new design points are sampled. The estimated variances are then used to guide future selection of design points. In the algorithm, $n_1$ and $n_2$ are preset sizes of the initial sampling and the sequential sampling respectively.

**Algorithm 1** Model Guided Adaptive Sampling (MGAS) Algorithm

1. Randomly sample at $n_1$ points from the design space, denote the collection of design points as $D$.

2. (a) Fit predictive model $f$ on $D$ and use the model to predict unsampled points.

   (b) Calculate the predictive variances for all points: $V_j = \text{var}[f(\mathbf{x}_j)|D]$; $j = 1, \ldots, N$, where $N$ is the size of the design space.

   (c) Let $q = 0$, $V = \max_j(V_j)$,

      i. Estimate the decrease in $V_j$ if an additional design point $\mathbf{x}_l$ is sampled: $dV_{j|l} = V_j - \text{var}[f(\mathbf{x}_j)|D \cup \{\mathbf{x}_l\}]$, $j, l = 1, \ldots, N$.

      ii. Method 1 (ALM): Select $k = \arg\max_l [\max_j dV_{j|l}]$; or
      Method 2 (ALC): Select $k = \arg\max_l (\sum_{j \notin D} dV_{j|l})$.

      iii. Let $q = q + 1$, $D = D \cup \{\mathbf{x}_k\}$ and $V_j = V_j - dV_{j|k}$.

      iv. If $q < n_2$, go back to 2(c)i. Otherwise repeat Step 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. However, $n_1$ should be large enough to build predictive models. If $n_2$ is sufficiently large, it could speed up 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, the predictive variance from the newly built predictive model 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 prespecified to control the improvement of the predictive variance. In this way, we do not need to prespecify $n_2$. In Section 4, we try a different $n_2$ and find that if the fitted model is close enough to the true model, a moderate change in $n_2$ will not have a big 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 in which the stopping issue is potentially user-dependent, we preset the total sample size throughout the paper.

**Remark 3.** The two methods in Step 2(c)ii are based on the two different active learning schemes proposed by MacKay [9] and Cohn [10]. These two methods are competitive. Their performances are evaluated in Section 4.

## 3. GENERAL MODEL GUIDED ADAPTIVE SAMPLING

To implement Algorithm 2, it is important to calculate predictive variances, and to estimate the change in

predictive variances for all design points after an additional design point is added. This calculation highly depends on the predictive models. For most nonparametric procedures, it is difficult to estimate the sequential changes of predictive variances. For most parametric models, the predictive variances and their changes may not be in an analytical form and/or the calculation could be expensive. In this section, we generalize the method in Algorithm 2 so that under certain assumptions, MGAS can easily be implemented with both Bayesian and Frequentist parametric predictive models or nonparametric procedures.

The change in posterior predictive variance depends not only on the predictive model and the sampling method, but also on the underlying population distribution. Assume the true model has the form $y = f + \epsilon$, where $f$ is a structure function of known information (e.g., the covariates $\mathbf{x}$), $\epsilon$ is the random error which could be 0 for deterministic functions, and $f$ and $\epsilon$ are independent. Therefore, $\text{var}(y|D) = \text{var}(f|D) + \text{var}(\epsilon|D)$. We need two assumptions to generalize MGAS: (A1) If a design point is sampled, the predictive variance of the structure part for the point reduces to zero, i.e. $\text{var}(f|D) = 0$. The motivation for this assumption and examples can be found in ref. 1. (A2) For any two design points $\mathbf{x}_i$ and $\mathbf{x}_j$, their structure functions can be decomposed into two parts: $f(\mathbf{x}_i) = af_0^{(i,j)} + f_i^{(i,j)}$ and $f(\mathbf{x}_j) = bf_0^{(i,j)} + f_j^{(i,j)}$, where $a$ and $b$ are unknown parameters, $f_0^{(i,j)}$ is the common structure component shared by $y_i$ and $y_j$, and $f_i^{(i,j)}$ and $f_j^{(i,j)}$ are the specific structure parts of $y_i$ and $y_j$ separately. We further assume that $f_0^{(i,j)}$, $f_i^{(i,j)}$ and $f_j^{(i,j)}$ are independent. Therefore, the variance of $y_i$ can be decomposed into three parts, $a^2 var(f_0^{(i,j)}) + \text{var}(f_i^{(i,j)}) + \text{var}(\epsilon_i)$, one shared by $y_i$ and $y_j$, one distinctive for $y_i$, and one by random error. Meanwhile, $a^2 var(f_0^{(i,j)})$ accounts for the reduction in $\text{var}(y_i)$ if we learn the shared part.

LEMMA 1: Under the above assumptions and notations, if design points $\mathbf{x}_i, \mathbf{x}_j$ and $\mathbf{x}_k$ are not sampled and used to build the initial model, then if $\mathbf{x}_k$ is sampled to update the model,

1. the posterior predictive variance of $y_j$ is reduced by $\frac{\sigma_{jk}^2}{\sigma_k^2}$;
2. the posterior covariance of $y_i$ and $y_j$ is reduced by $\frac{\sigma_{ik}\sigma_{jk}}{\sigma_k^2}$.

where $\sigma_i^2$ is the predictive variance of $y_i$ and $\sigma_{ij}$ is the covariance between $y_i$ and $y_j$ before the model updating.

The proof of Lemma 1 is in the Appendix. Lemma 1(1) implies that the more correlated the design points $\mathbf{x}_j$ and $\mathbf{x}_k$ are, the larger the reduction in posterior predictive variance of $y_j$ after the point $\mathbf{x}_k$ is sampled. Lemma 1(2) shows how to calculate the sequential change of predictive covariances without fitting a new model with new samples. The lemma indicates that the analytical forms of $f_0^{(i,j)}$, $f_i^{(i,j)}$ or $f_j^{(i,j)}$ are not needed for sampling purposes.

In the cases that (A1) and (A2) do not hold, we can consider the following way to estimate the reduction in predictive variance of $y_j$ after $\mathbf{x}_k$ is sampled. If we predict $y_j = \eta(y_k)$ by the linear model of the form $\hat{\eta}(y_k) = \alpha + \beta y_k$, the variance of $y_j$ can be decomposed [18] to

$$\text{var}(y_j) = \text{var}\{\eta(y_k) - \hat{\eta}(y_k)\} + \frac{\sigma_{jk}^2}{\sigma_k^2},$$

where the first term measures the lack of fit and the second term is the same as in Lemma 1(1), measuring the variance of $y_j$ that is explained by this fitted approximation with $y_k$. By these arguments, MGAS can be generalized to the GMGAS as following:

**Algorithm 2** General Model Guided Adaptive Sampling Algorithm

1. Randomly sample $n_1$ points from the design space, denote the collection of design points as $D$.
2. (a) Fit predictive model $f$ on $D$ and use the model to predict unsampled points.

   (b) Calculate the predictive variance–covariance between all points:

   $$V_{ij} = \text{cov}(f(\mathbf{x}_i), f(\mathbf{x}_j)|D); i, j = 1, \ldots, N.$$

   (c) Let $q = 0$,

   i. Method 1 (ALM): Select $k = \arg\max_l [\max_j d V_{j|l}] = \arg\max_l V_{ll}$; or
   Method 2 (ALC): Select $k = \arg\max_l (\sum_{j \notin D} \frac{V_{jl}^2}{V_{ll}})$.

   ii. Let $q = q + 1$, $D = D \cup \{\mathbf{x}_k\}$ and $V_{ij} = V_{ij} - \frac{V_{ik}V_{jk}}{V_{kk}}$.

   iii. If $q < n_2$, go back to 2(c)i. Otherwise repeat Step (2) until stopping criterion is met.

Algorithm 2 generalizes MGAS so that it can be implemented as long as we can obtain the predictive variance–covariance matrix for the unsampled design points.

**Remark 1.** For Method 1 in Step 2(c)i, the second equality holds since if $k = \arg\max_l V_{ll}$, then $V_{kk} \geq V_{jj}V_{ll}/V_{ll} \geq V_{jl}^2/V_{ll} = dV_{j|l}$ by Lemma 1(1), for any $j$ and $l$.

**Remark 2.** If the design space is very huge or any covariate is continuous, the practitioner can consider adapting space-filling methods such as maximum entropy designs to select well-spaced candidate subspaces to represent the whole design space. Then the predictive variance–covariance matrix is estimated only at the candidate subspaces.

**Remark 3.** The predictive model could be any Bayesian or Frequentist predictive process. For the general Bayesian models, we simulate predictions from the posterior distributions and then calculate the predictive variance–covariance matrix from the simulation [6]. For the general Frequentist models, the prediction simulation can be obtained by using bootstrap method [19]. An example can be found in ref. 17. Note that we provide a general method to obtain the predictive variance–covariance matrices. Simulation and approximation may not be necessary for some parametric models, where the matrices have analytical forms. Even when the analytical forms become available, our method sometimes can simplify the calculation without loss of efficiency. An example is given in Section 4.

In this paper, we use BART very often as the predictive model to guide the adaptive sampling. BART is a Bayesian 'sum-of-trees' used to model the relationship between the response and explanatory variables [20]. The method has shown excellent predictive performance. (See, for example, refs 21 and 22 and literature therein.) Readers interested in BART are referred to the original paper by Chipman *et al*. [20] and the paper by Yu *et al*. [6], who reviewed BART and its inference methods. We utilize BART-guided adaptive sampling (BGAS) to explore examples in Section 4.

## 4. EXAMPLES

We apply MGAS along with different predictive models to a real-life design study and two simulation studies. We observe that in the real design study, when using with the same predictive model, BART or multiple additive regression trees (MART, [23]), MGAS outperforms both the simple random sample (SRS) and the space-filling method, maximin. We also use the real data to evaluate the effect of $n_2$ on the BGAS predictive performance. We find that a moderate increase in $n_2$ would not significantly affect the predictive performance, while it is more efficient to select multiple points without reinitiating the sampling process or rerunning the model.

The first simulation is based on a deterministic real function. In this simulation, we demonstrate the flexibility of MGAS with any predictive models. The GP is a popular model in computer experiments and has been extended to the treed GP. Both processes can cooperate with MGAS. By so doing, the predictive performance from the design points selected by MGAS can be improved if compared with traditional adaptive sampling methods such as ALC and ALM.

The second simulation aims to demonstrate the efficiency of our method in dealing with nonlinear real functions. It shows why BGAS is efficient in terms of adaptive sampling and the superior predictive accuracy of BART over linear models. We also compare the estimated predictive variances from Lemma 1 with the true variances in this simulation.

### 4.1. A Computer Architectural Design Study

We test MGAS on a computer architectural design space of relatively small size, 1600 design points. With the small design space, 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 of which has 2, 4 or 5 levels. We then test the processor performance of the 1600 different designs on two CPU benchmarks, GCC and TWOLF, from the Standard Performance Evaluation Corporation (SPEC) CPU 2000. These 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 that 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 the so-called set-associative where each group is termed a set. The number of blocks in a group (or set) is cache set associativity, which is recorded by L1CA or L2CA in this experiment.

For comparison, we use the predicted R-square ($PR^2$, defined by Yu *et al*. [6]), to measure the 'goodness-of-prediction'. $PR^2$ is calculated on the whole design space. It is different from the traditional R-square which is used to measure 'goodness-of-fit' of the model. When calculating $PR^2$, only a small proportion of the design space (at most 200 out of the 1600 points in this example) are used for model fitting.

We start with 30 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. We use both methods in Algorithm 3 Step 2(c)i, denoted by BGAS1 and BGAS2 separately. To compare BGAS, SRS and maximin, we use the same predictive

**Table 1.** The average sample size needed to get the critical $PR^2$.

| | GCC Performance | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $PR^2$ | 0.80 | 0.85 | 0.90 | 0.95 | 0.96 | 0.97 | 0.98 | 0.99 |
| SRS | 51 | 60 | 70 | 97 | 105 | 120 | 148.5 | >200 |
| Maximin | 53 | 62 | 75 | 140 | 162 | 172 | 176 | >200 |
| BGAS1 | 47 | 55 | 63 | 81 | 86 | 95 | 105 | 138 |
| BGAS2 | 47 | 52 | 61 | 79 | 87 | 94 | 110 | 144 |
| | TWOLF Performance | | | | | | | |
| $PR^2$ | 0.80 | 0.85 | 0.90 | 0.95 | 0.96 | 0.97 | 0.98 | 0.99 |
| SRS | 31 | 34 | 46 | 90 | 108 | 134 | 164 | >200 |
| Maximin | 30 | 34 | 53 | 134 | 153 | 170 | 178 | >200 |
| BGAS1 | 31 | 34 | 43 | 73 | 93 | 118 | 138 | 158 |
| BGAS2 | 31 | 34 | 41 | 74 | 91 | 117 | 139 | 170 |

model, BART, with these sampling methods separately. In BART fitting, we skip the first 1000 simulated sum-of-trees as burn-ins and then keep one from every four simulations. By so doing, we obtain 1000 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. The distance function in the maximin method is the same as that in ref. 17. Each sampling method was repeated 20 times. Table 1 exhibits the average sample sizes (of the 20 repetitions) needed to reach the critical $PR^2$ values in predicting the two process performances, GCC and TWOLF, respectively.

Table 1 shows that both methods in MGAS improve 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 a higher $PR^2$ (say, for example, $PR^2 \geq 0.97$). The BGAS1 and BGAS2 methods show negligible difference in this study: BGAS1 seems to be better when the total sample size is large, and the conclusion reverses when the total sample size is small.

The model guided adaptive design can be used with any models. For this example, we also use MART with MGAS, where the prediction variance–covariance matrix is estimated by bootstrap. More specifically, we resample with replacement from the selected design points and fit a MART on the resampled data. From the fitted MART, we predict the unsampled design points and then calculate the variance–covariance matrix. For comparison, we combine MART with the methods in Algorithm 3 Step 2(c)i, noted as 'MGAS1 + MART' and 'MGAS2 + MART', and with SRS and maximin sampling methods separately. Since MART needs a larger sample size to build a model, we start with 50 randomly selected design points and then increase the sample size by 10 each time until we get 200 points. Figure 1 shows the performance of different methods. We observe that MGAS outperforms SRS and maximin with both BART and MART in this example. The two sampling methods described in the algorithm are competitive with BART model, while with MART, Method 1 seems slightly worse when sample size becomes bigger. With each sampling scheme, the predictive model BART is better than MART in this example.

For GCC data, Table 2 presents the average final sampling frequencies (of the $200 \times 20$ samples) for each level of every covariate, based on BGAS1, BGAS2, SRS and maximin separately. The number in each cell is the percentage of times that the corresponding level of the covariate is selected. By SRS and maximin, every level of a covariate has about the same chance of being selected. BGAS chooses samples adaptively and thus results in better predictions as demonstrated in Table 1. Note that BGAS1 and BGAS2 seem to choose very different samples.

It is interesting to see how MGAS performs when $n_2$ varies. Figure 2 shows that for both BGAS1 and BGAS2, the differences in $PR^2$ are negligible when $n_2$ ranges from 5 to 30. These differences become significant when $n_2$ is 50, where $PR^2$ is the smallest.

Theoretically, the predictive performance of BGAS should be better when $n_2$ is smaller. But a smaller $n_2$ would result 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 $n_2$. When designing the BGAS algorithm, our goal is to best approximate the posterior variance so that we can choose multiple design points without rerunning the models or incurring too much error. As in this study, the predictive error is about the same when $n_2$ is between 5 and 30. In practice, we can set up a cost function to study how the costs attributed to time and to the loss of predictive accuracy change with $n_2$, and then search for the best $n_2$.

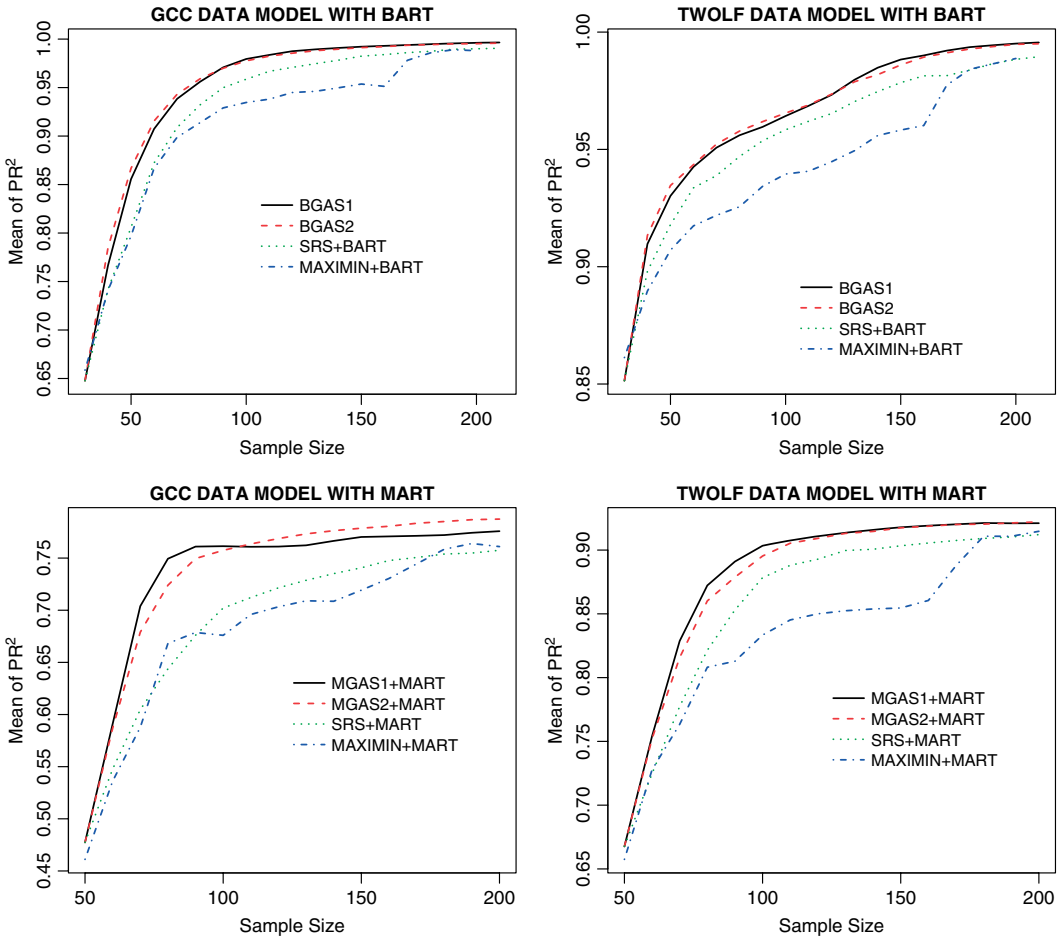


Fig. 1 For the real data set, comparison of MGAS with SRS and maximin. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

**Table 2.** Proportion of the corresponding level of each covariate being selected in the sample.

| | L1CBS | | L1CA | | | | L2CS | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Covariates levels | 32 | 64 | 1 | 2 | 4 | 8 | 256 | 512 | 1024 | 2048 | 4096 |
| BGAS1 | 50 | 50 | 32 | 23 | 20 | 25 | 27 | 19 | 17 | 17 | 20 |
| BGAS2 | 50 | 50 | 29 | 26 | 23 | 23 | 24 | 21 | 19 | 18 | 17 |
| 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 | | | | |
| Covariates levels | 64 | 128 | 8 | 16 | 32 | 64 | 1 | 2 | 4 | 8 | 16 |
| BGAS1 | 51 | 49 | 32 | 23 | 20 | 25 | 34 | 19 | 15 | 14 | 17 |
| BGAS2 | 51 | 49 | 27 | 26 | 24 | 23 | 28 | 21 | 18 | 17 | 17 |
| 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. Simulation 1

The GP is prevalent in computer experiments [5]. Gramacy and Lee [12] adapted the GP to the treed Gaussian process (TGP) extending the Bayesian treed linear model by using a GP model with linear trend independently within each region divided by the Bayesian tree process. They further developed an adaptive sampling method in computer designs [13] (noted as GL) that uses TGP as the predictive model and adaptively chooses design points

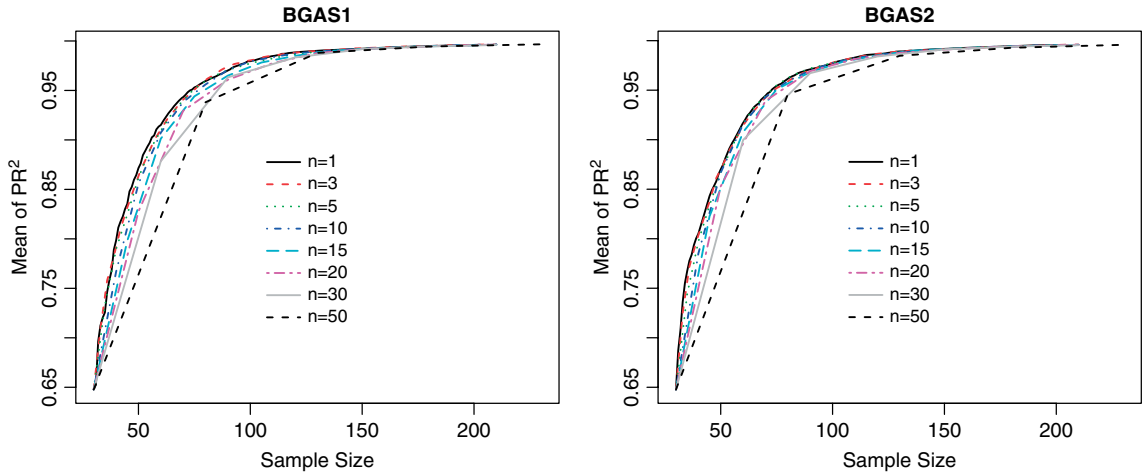


Fig. 2 Comparison of BGAS methods when $n_2$ changes. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

using ALC or ALM. When we use the ALC or ALM criterion in MGAS, our sampling methods differ from GL in two aspects: first, to reduce the correlation among the selected design points, GL uses space-filling designs, such as maximin distance design and latin hypercube sampling (LHS), to select a candidate data set. Therefore, the correlations are controlled by prespecified 'distance'. In contrast, we develop a theory that can be used to estimate the sequential change of the predictive variances of unsampled design points. The correlations among design points are dynamically considered in that sampling at a design point would incur big changes in predictive variances of those points highly correlated with it. Second, to calculate the predictive variance–covariance matrix of the candidate design space, GL uses the GP by assuming that the density over output at a new point has a normal distribution. MGAS has no assumption on the density in general and thus can be used with any nonparametric models. For parametric models, MGAS is also applicable if the calculation of the adjusted variance is too complicated.

In this simulation, we compare the sampling efficiencies of MGAS and GL. For this purpose, we use MGAS guided by the TGP or GP method. We adopt the high-dimensional simulation in GL with a minor adjustment. This is a six-dimensional example, with true response

$$z(x_1, x_2, x_3, x_4, x_5, x_6) = \exp\{\sin\left([0.9(x_1 + 0.48)]^{10}\right)\} + x_2 x_3 + x_4.$$

This function has four active variables. We also allow the inputs to uniformly vary in [0, 1]. The simulation is slightly different from GL in that the responses here are calculated with no noise. Therefore, the response surface is deterministic. For all methods, we start with 30 samples. At each run, we start with an initial set of 1000 configurations from a maximum entropy design, and then choose ten samples from the candidate data sets. The process is repeated until we reach 200 samples. To compare sampling and modeling methods, we start with the same candidate sets. As in GL, ALC works better than ALM in this simulation, and TGP with jumps to the limiting linear models (TGPLLM) is the best. Therefore, we use the ALC criterion and compare the following processes denoted in the form of 'modeling method + sampling methods': 'GP + ALC', 'TGPLLM + ALC', 'GP + MGAS2', 'TGPLLM + MGAS2' and 'BART + MGAS2'. Figure 3 shows the average $R^2$ and its variances as evaluated on 20 random LHSs of size 1000 in $[0, 1]^6$ for 20 repeated modeling and sampling runs.

As is shown Fig. 3, with the same model, GP or TGPLLM, MGAS2 outperforms ALC. Among the three modeling methods, GP, BART and TGPLLM, TGPLLM outperforms GP. BART is slightly worse than TGPLLM in terms of the predictive performance when the sample size is small, but it catches up quickly and becomes much better when the sample size accumulates. As for the predictive variances, GP introduces the lowest variances while TGPLLM has the largest. Furthermore, when using the same model, the variances from MGAS2 are smaller than those from ALC. The variances from 'BART + MGAS2' are comparatively small and appear to fluctuate less as the sample size varies, compared with other processes. When the sample size is 170 or more, 'BART + MGAS2' performs the best and 'GP + ALC' is the worst. These differences are statistically significant. All the computations are carried out in *R*, with 'tgp' package [24] for TGPLLM and GP fitting and 'BayesTrees' package for BART fitting. 'BART + MGAS2' requires much less time than other processes.

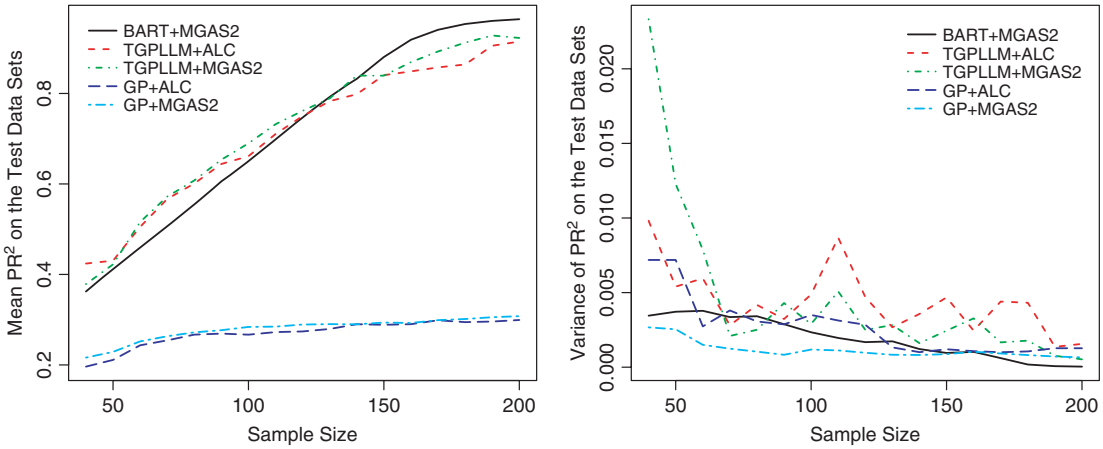


Fig. 3 For simulation 2, left panel compares the mean $R^2$ of the different 'modeling + sampling' process. The right panel compares their variances. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

### 4.3. Simulation 2

In the simulation, we create five covariates. Each covariate has six levels $(0, 0.2, 0.4, 0.6, 0.8, 1)$. Thus the design space is composed of 7776 different design points. The response variable is simulated from the 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 zero 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 7776 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. In addition, BART is compared with the linear models chosen by AIC and BIC, and all models are built on randomly selected samples. The left panel of Fig. 4 shows the mean $PR^2$ of the 20 repetitions. The right panel shows the corresponding variance of the $PR^2$.

Figure 4 shows that the AIC and BIC model selection criteria outperform BART in model building and predictions when the sample size is very small. This is due to the fact that with small sample size, both linear models and BART have high biases, while simpler model usually has lower variances. As the sample size increases, BART shows much better predictive performance since the biases are significantly decreased with the sample size. Combined with BART, BGAS consistently outperforms SRS in that adaptive sampling reduces predictive variances quickly. The right panel of Fig. 4 indicates that the predictive variance for BART is larger than that for linear models (compare 'SRS + BART' with 'SRS + AIC' and 'SRS + BIC'). But the predictive variances of BGAS1 and BGAS2 are consistently smaller than that of SRS with BART. BGAS1 also has consistently smaller predictive variance than the linear regression models. From this simulation, we see that BGAS1 is better than BGAS2 in terms of $PR^2$ and has smaller predictive variance when the sample size becomes large.

We also investigate the difference between the estimated sum of predicted variance from Lemma 1 and the true variance. As is shown in the left panel of Fig. 5, the estimated variance is smaller than the calculated variance from BART simulation. But it becomes stable at around 80% of the calculated predictive variance when the sample size reaches 200. This is partially due to the fact that the calculated predictive variances from BART simulation actually include both the predictive variance and the variance of random errors. The right panel of Fig. 5 shows the proportion of the calculated predictive variance contributes to the sum of squared error. We find that overall, model guided sampling methods result in greater reduction in predictive variance than does the simple random sampling.

## 5. CONCLUSIONS AND FURTHER RESEARCH

Computer experiments have been popular in exploring huge design spaces without real experiments. However, such exploration has recently become very challenging because of the large number of parameters involved. In this paper, we propose a GMGAS scheme, aiming at efficiently sampling a small proportion of the design space that can provide high predictive accuracy. By combining with modeling processes such as BART, MART, TGPLLM and GP, MGAS has shown its superiority over other sampling schemes. The beauty of MGAS lies in the facts that it is applicable with any modeling process,

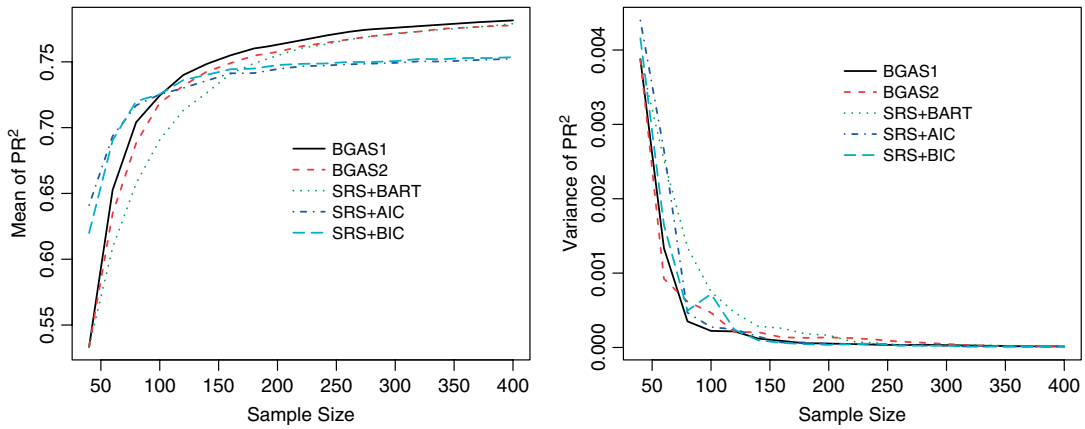


Fig. 4 For simulation study 2, comparison of adaptive experimental design with SRS and BART with linear regression model. Left plot shows the mean predictive R-square of each method as the sample size increases. Right plot shows the variance of predictive R-square of the 20 repetitions. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

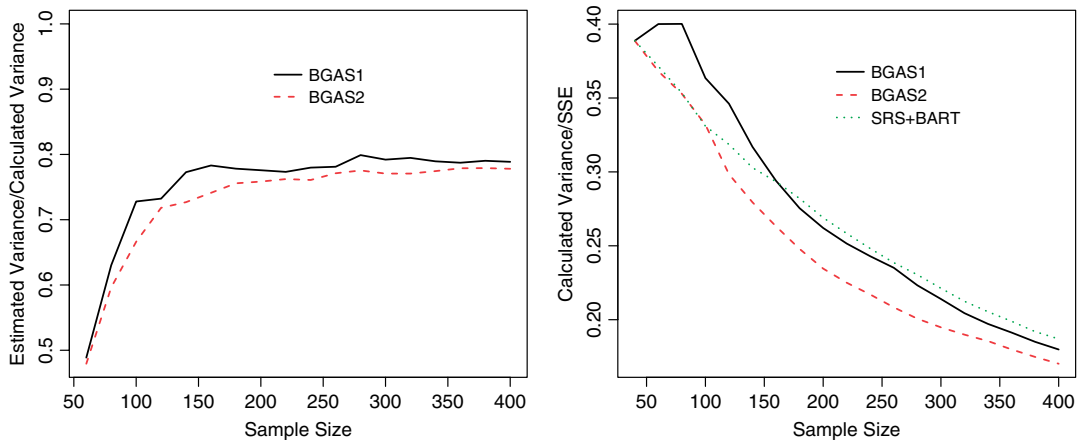


Fig. 5 For simulation study 2, left panel compares the estimated variance with the calculated variance from BART simulation. The right panel shows the proportion of the calculated predictive variance contributed to the sum of squared error. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

parametric or nonparametric, and that its calculation is straightforward.

The proposed method may be applied to find optimal designs from huge spaces. Instead of randomly selecting candidate design points, we choose design points that are predictive of extreme values. By predicting the posterior predictive intervals for the quantity of interest on design points, we could sample all the points whose posterior intervals overlap that of the predicted extreme value. This will be left as further research topic.

## ACKNOWLEDGMENTS

The authors thank Dr Hugh Chipman, Ms Patricia Andrews, the editor, associate editor and referees for constructive comments and suggestions that helped to improve the presentation of the paper. This research was supported in part by the NSF Award 1017961.

## APPENDIX: PROOF OF LEMMAS

**Proof:** Proof of Lemma 1
By (A2), we have that $y_k = af_0^{\{k,j\}} + f_k^{\{k,j\}} + \epsilon_k$ and $y_j = bf_0^{\{k,j\}} + f_j^{\{k,j\}} + \epsilon_j$, for unsampled points $\mathbf{x}_j$ and $\mathbf{x}_k$. Note that for deterministic functions, $\text{var}(\epsilon_i)$ are 0. This fact dose not have any effect on the proof.

1. If design point $\mathbf{x}_k$ is sampled, then by (A1), $\text{var}(f_0^{\{i,j\}}|y_k) = 0$. Therefore,

$$\begin{aligned}\text{var}(y_j|y_k) &= \text{var}(bf_0^{\{k,j\}} + f_j^{\{k,j\}} + \epsilon_j|y_k)\\ &= \text{var}(f_j^{\{k,j\}}) + \text{var}(\epsilon_j),\\ \Rightarrow \text{var}(y_j) - \text{var}(y_j|y_k) &= b^2\text{var}(f_0^{\{k,j\}}) + \text{var}(f_j^{\{k,j\}})\\ &\quad + \text{var}(\epsilon_j) - \text{var}(f_j^{\{k,j\}}) - \text{var}(\epsilon_j) = b^2\text{var}(f_0^{\{k,j\}}).\end{aligned}$$

Since $y_j = -\frac{b}{a} f_k^{\{k,j\}} + \frac{b}{a} y_k + f_j^{\{k,j\}} + (\epsilon_j - \frac{b}{a}\epsilon_k)$, which is a linear regression with $-\frac{b}{a} f_k^{\{k,j\}}$ being the intercept, $y_k$ and $f_j^{\{k,j\}}$ being two independent regressors and $(\epsilon_j - \frac{b}{a}\epsilon_k)$ being the random error. Therefore, $\frac{b}{a}$, the slope for $y_k$ regressed on $y_j$, equals $\frac{\sigma_{jk}}{\sigma_k^2}$. Also,

$$\begin{aligned}\mathrm{cov}(y_j, y_k) &= \mathrm{cov}(bf_0^{\{k,j\}} + f_j^{\{k,j\}} + \epsilon_j, af_0^{\{k,j\}} + f_k^{\{k,j\}} + \epsilon_k) \\ &= \mathrm{abcov}(f_0^{\{k,j\}}, f_0^{\{k,j\}}) = \mathrm{abvar}(f_0^{\{k,j\}}).\end{aligned}$$

Therefore,

$$\begin{aligned}\mathrm{var}(y_j) - \mathrm{var}(y_j|y_k) &= \frac{b}{a} \cdot abvar(f_0^{\{k,j\}}) \\ &= \frac{\sigma_{jk}}{\sigma_k^2} \cdot \mathrm{cov}(y_j, y_k) = \frac{\sigma_{jk}^2}{\sigma_k^2}.\end{aligned}$$

2. When design point $\mathbf{x}_k$ is sampled, let $\rho_{ij} = \sigma_{ij}/(\sigma_i\sigma_j)$ be the correlation coefficient between $y_i$ and $y_j$, and $\sigma_{i|k}^2$ the conditional variance of $y_i$ given $y_k$, then

$$\begin{aligned}&\mathrm{cov}(y_i, y_j|y_k) = \rho(y_i, y_j|y_k) \cdot \sigma_{i|k} \cdot \sigma_{j|k} \\ &= \frac{\rho_{ij} - \rho_{ik}\rho_{jk}}{\sqrt{1-\rho_{ik}^2} \cdot \sqrt{1-\rho_{jk}^2}} \cdot \sigma_{i|k} \cdot \sigma_{j|k} \\ &= \frac{\sigma_{ij}\sigma_k^2 - \sigma_{ik}\sigma_{jk}}{\sqrt{\sigma_i^2\sigma_k^2 - \sigma_{ik}^2} \cdot \sqrt{\sigma_j^2\sigma_k^2 - \sigma_{jk}^2}} \cdot \sqrt{\frac{\sigma_i^2\sigma_k^2 - \sigma_{ik}^2}{\sigma_k^2}} \cdot \sqrt{\frac{\sigma_j^2\sigma_k^2 - \sigma_{jk}^2}{\sigma_k^2}} \\ &= \frac{\sigma_{ij}\sigma_k^2 - \sigma_{ik}\sigma_{jk}}{\sigma_k^2} \\ &= \sigma_{ij} - \frac{\sigma_{ik}\sigma_{jk}}{\sigma_k^2}.\end{aligned}$$

■

## REFERENCES


[1] D. R. Jones, M. Schonlau, and W. J. Welch, Efficient global optimization of expensive black-box functions, J Glob Optim 13 (1998), 455–492.

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

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

[4] P. Joseph, K. Vaswani, and M. Thazhuthaveetil, Use of linear regression models for processor performance analysis, In Proceedings of 12th IEEE Symposium on High Performance Computer Architecture (HPCA-12), 2006, 99–108.

[5] S. Seo, M. Wallat, T. Graepel, and K. Obermayer, Gaussian process regression: active data selection and test point rejection, In Proceedings of the International Joint Conference on Neural Networks (IJCNN), IEEE, 2000, 241–246.

[6] Q. Yu, B. Li, Z. Fang, and L. Peng, An adaptive sampling scheme guided by BART - with an application to predict processor performance, Can J Stat 38(1) (2010), 136–152.

[7] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, Design and analysis of computer experiments, Stat Sci 4 (1989), 409–435.

[8] T. J. Santner, B. J. Williams, and W. I. Notz, The Design and Analysis of Computer Experiments, New York, Springer Verlag, 2003.

[9] D. J. C. Mackay, Information-based objective functions for active data selection, Neural Comput 4(4) (1992), 589–603.

[10] D. A. Cohn, Neural network exploration using optimal experiment design, Neural Netw 9(6) (1994), 1071–1083.

[11] P. Kim, and Y. Ding, Optimal engineering system design guided by data-mining methods, Technometrics, 47(3) (2005), 336–348.

[12] R. B. Gramacy, and H. K. H. Lee, Bayesian treed Gaussian process models with an application to computer modeling, J Am St Assoc 103 (2008), 1119–1130.

[13] R. B. Gramacy and H. K. H. Lee, Adaptive design and analysis of supercomputer experiments, Technometrics 51 (2009), 130–145.

[14] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, Information, prediction, and query by committee, In Proceedings of Advances in Neural Information Processing Systems, 1993, 483–490.

[15] K. Sung and P. Niyogi, Active learning for function approximation, Proc Adv Neural Inf Process Syst 7 (1995), 593–600.

[16] M. Saar-Tsechansky, and F. Provost, Active learning for class probability estimation and ranking, In Proceedings of 17th International Joint Conference on Artificial Intelligence, 2001, 911–920.

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

[18] J. E. Oakley and A. O'Hagan, Probabilistic sensitivity analysis of complex models: a Bayesian approach, J R Stat Soc [Ser B] 66 (2004), 751–769.

[19] B. Efron and R. J. Tibshirani, An Introduction to the Bootstrap, Boca Raton, FL, CRC Press, 1994.

[20] H. A. Chipman, E. I. George, and R. E. McCulloch, BART: Bayesian additive regression trees, Ann Appl Stat 4(1) (2010), 266–298.

[21] X. Zhang, T. Y. Shih, and P. Muller, A spatially-adjusted Bayesian additive regression tree model to merge two datasets, Bayesian Anal 2(3) (2007), 611–634.

[22] Q. Yu, S. N. MacEachern, and M. Peruggia, Bayesian synthesis: Combining subjective analysis, with an application to ozone data. The Ann Appl Stat 5(2B) (2011), 1678–1698.

[23] J. H. Friedman, Greedy function approximation: a gradient boosting machine, Ann Stat 29 (2001), 1189–1232.

[24] R. B. Gramacy, tgp: An R package for Bayesian nonstationary, semiparametric nonlinear regression and design by treed Gaussian process models, J Stat Softw 19(9) (2007), 1–46.