Title

Variable selection via a multi-stage strategy

Journal

Journal of Applied Statistics, 42(4)

ISSN

0266-4763

Authors

Chang, Jing

Lee, Herbert KH

Publication Date

2015-04-03

DOI

10.1080/02664763.2014.985640

Peer reviewed

Variable Selection via a Multi-stage Strategy

Jing Chang*and Herbert K. H. Lee†

University of North Carolina, Chapel Hill

and University of California, Santa Cruz

April 20, 2014

Abstract

Variable selection for nonlinear regression is a complex problem, made even more difficult when there are a large number of potential covariates and a limited number of datapoints. We propose herein a multi-stage method that combines state of the art techniques at each stage to best discover the relevant variables. At the first stage, an extension of the Bayesian Additive Regression tree is adopted to reduce the total number of variables to around 30. At the second stage, sensitivity analysis in Treed Gaussian Process is adopted to further reduce the total number of variables. Two stopping rules are designed and sequential design is adopted to make best use of previous information. We demonstrate our approach on two simulated examples and one real dataset.

Key words: Regression Tree, Gaussian Process, Sequential Design, Sensitivity Analysis, Sum of Tree.

*postal address: Anhui Provincial Hospital first staff dormitory, Unit 1, Room 2104 email:jingbb@gmail.com Jing Chang is corresponding author

†postal address: 1156 High Street, Santa Cruz, CA 95064 email: herbie@ams.ucsc.edu

1 Introduction

Variable selection is an important task in a wide variety of problems. Much of the literature focuses on variable selection for linear regression, or for simple parametric models. We are interested in the case of relationships that are potentially highly nonlinear, where we want to allow for nonparametric relationships. Such problems occur across many applications. One particular application that was our original motivation is computer modeling (Santner et al., 2003), where a computer simulation replaces or supplements a physical experiment. In many of these problems, there can be a large number of input variables, and one needs to winnow out the vast number of relatively unimportant variables to focus on the most important ones. Such a reduction can greatly reduce the computational demands and may be a prerequisite for further analysis. It can also be helpful to determine the relative importance of each input variable to find out which of them have crucial influence on the system being studied (Linkletter et al., 2006). Variable selection can help in three key aspects: improving the performance of the predictors, providing more time-efficient and cost-effective predictors, and providing a better understanding of the underlying data-generating processes.

We base our approach on the variable selection literature in the context of computer experiments, as this literature explicitly focuses on nonlinear effects. Linkletter et al. (2006) performs variable selection by approximating the simulator with a Gaussian process using a special correlation structure and prior settings. Then the posterior distribution of the correlation between each variable and the response is used to indicate the relative importance of each variable. They also include an inert variable to act as a reference to differentiate real association from that due to chance. Bondell et al. (2009) breaks down the regression into interpretable main effects and interaction effects. The main effect and interaction effect are modeled by Gaussian processes with different covariance structures. Morris (1991) designs the experiment using individually randomized one-factor-at-a-time designs, then data

is analyzed based on the resulting random sample of *elementary effects*, the changes in an output caused by changes in a certain input. Oakley (2009) adopts the idea of perfect information in decision theory to measure the importance of contributions of the input variables. The Gaussian process emulator is used to estimate partial EVPIs (expected value of perfect information) particularly efficiently using Bayesian quadrature.

However, existing statistical methods for variable selection struggle to effectively identify a few nonlinear variables out of hundreds with high accuracy with only a moderate number of samples. We develop a multi-stage strategy which employs a state-of-the-art technique for each of the stages. By making the best use of each technique, in combination they can accomplish a sophisticated goal that can not be achieved separately. In the first-stage, we use treed models to reduce the total number of variables to around 30. In the second stage, we use a sequential design to sample more data points to search for variables with small but significant contribution to prediction. In the last stage, sensitivity analysis is performed and two stopping rules are designed to reduce the number of variables further as needed.

While our original motivation was for computer experiment applications, and the methodology was developed in the expectation that we would have access to a computer simulator with a large number of inputs that we could run adaptively, the expected simulator did not become available. So instead, we have shifted our application to the more general regression setting. However, many of our modeling choices still reflect this initial motivation of computer simulation experiments, and we think that our methodology will be most useful in that context.

In Section 2 we discuss the methodological components underlying our approach: the sum of trees method, the treed Gaussian Process, and the Monte-Carlo based numerical procedure for calculating sensitivity indices. The novel multi-stage strategy for variable selection is illustrated in Section 3. In Section 4, we analyze two simulated examples with the method proposed, and in Section 5 we analyze a real dataset on crime, followed by some

concluding remarks.

2 Elements of the methodology

Our approach relies upon recent developments in tree models. Chipman et al. (2010) develops a 'sum-of-trees' model, Bayesian Additive Regression Trees (BART). Each tree is constrained by a regularization prior to be a weak learner and is modeled independently. The Bayesian backfitting algorithm is used to generate samples from the posterior distribution. Variables are selected randomly for splitting and a high splitting frequency represents a high association of the variable with the response. Gramacy and Lee (2008) proposes a partition model which divides up the input space and fits independent Gaussian process models to the data in each subregion.

We combine tree models with sequential design in which the procedures used for selecting the sample units depend on observations made during the original sampling to make the best use of information contained in the sample (Thompson and Seber, 1996). Previous research relevant to sequential design in computer experiments includes Gramacy and Lee (2009), where data points are sampled in a way to minimize the standard deviation in predicted output or minimize the expected square error averaging over the input space, and Santner et al. (2003) and Taddy et al. (2009), where sequential sampling is performed by taking additional data points maximizing the expected value of an objective function.

2.1 The sum of trees method

BART (Chipman et al., 2010) sums a number of smaller trees to get a flexible and efficient estimator. For a single tree, let T denote a binary tree consisting of a set of interior node decision rules and a set of terminal nodes. Let $M = \{\mu_1, \mu_2, \ldots, \mu_b\}$ denote a set of parameter values ready to be assigned to the b terminal nodes of T. The decision rule is to split the

predictor region into two parts, cutting at a certain point A within the range of a particular variable x, i.e., $x \leq A$ and $x > A$. The splitting rule is used for the whole tree. Each possible input value is assigned to a terminal node. For a given T and M, we use $g(x; T, M)$ to denote the function which assigns a $\mu_i \in M$to x. Thus,

$$Y = g(x; T, M) + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2) \tag{1}$$

where $g(x; T, M)$ assigns $\mu_i = E(Y|x)$ to the terminal node.

This process is repeated independently for m trees. The sum-of-trees model is:

$$Y = \sum_{j=1}^{m} g(x; T_j, M_j) + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2) \tag{2}$$

where T_j is a binary regression tree, M_j is the terminal node, and $g(x; T_j, M_j)$ is the function which assigns $\mu_{ij} \in M_j$ to x. For the sum of trees model, $E(Y|x)$ is the aggregate of all μ_{ij}'s assigned to x by the $g(x; T_j, M_j)$'s. A regularization prior completes the model:

$$p((T_1,M_1), ..., (T_m,M_m), \sigma) = \left[\prod_j p(T_j,M_j)\right] p(\sigma) = \left[\prod_j p(M_j|T_j)p(T_j)\right] p(\sigma)$$

where $p(M_j|T_j) = \prod_i p(\mu_{ij}|T_j)$ with $\mu_{ij} \in M_j$. Each tree component (T_j, M_j) and σ are independent of each other. Thus, the prior choice problem simplifies to the choice of just $p(T_j)$, the priors for the tree structure, $p(\mu_{ij}|T_j)$, the mean values in the leaves, and $p(\sigma)$, the overall variance. The prior of T is specified by three aspects: (i) the probability that a node at depth d is not terminal, given by $\alpha(1 + d)^{-\beta}$, where $\alpha \in (0, 1)$ and $\beta \in [0, \infty)$ are pre-specified shape parameters, (ii) the distribution on the splitting variable assignments at each interior node, (iii) the distribution on the splitting rule assignment in each interior node, conditional on the splitting variable. The priors for (ii) and (iii) are uniform priors, $p(\mu_{ij}|T_j)$ is the conjugate normal distribution $N(\mu_\mu, \sigma_\mu^2)$, where following Chipman et al.

(2010), we choose μ_μ and σ_μ so that $m\mu_\mu - 2\sqrt{m}\sigma_\mu = y_{min}$ and $m\mu_\mu + 2\sqrt{m}\sigma_\mu = y_{max}$. The prior for σ^2 is $\nu\lambda/\chi^2_\nu$ where we take $\nu = 3$ and choose λ such that the 0.9 quantile of the prior for is located at a data-based estimate $\hat{\sigma}^2$. . All $p(T_j)$ and $p(\mu_{ij}|T_j)$ can be considered as having identical forms. The Bayesian backfitting algorithm (Hastie and Tibshirani, 2000) is used when implementing BART, and we use the BayesTree package in R Chipman and McCulloch (2010).

Since the sum of trees model is a nonparametric Bayesian regression approach, it can perform the tasks of regression, prediction, estimation of partial dependence functions and detection of low dimensional structure within high dimensional data. When the response is a binary variable, the sum of trees model can also be used for classification. In either case, the variables highly associated with the responses will tend to have higher splitting frequencies in the model fitting process. By selecting variables with high splitting frequencies, we can select variables highly associated with the response (Chipman et al., 2010).

2.2 Treed Gaussian Process

The standard model in the literature for a computer experiment is a Gaussian process (GP) (Sacks et al., 1989; Santner et al., 2003). A GP is a collection of random variables $Z(x)$ with typically multivariate explanatory variable x, having a jointly Gaussian distribution for any finite subset of explanatory variables. It is specified by a mean function $\mu(x) = E(Z(x))$ and a correlation function $K(x, x') = \frac{1}{\sigma^2}E([Z(x) - \mu(x)][Z(x') - \mu(x')]^T)$. Usually the Gaussian process model is of the form $Z(x) = \xi(x, \beta) + w(x) + \eta(x)$, where $\xi(x, \beta)$ is a simple mean function, $w(x)$ is a random process with mean zero and correlation function $K(x_i, x_j)$ and $\eta(x)$ is Gaussian noise. We use a linear mean trend $\xi(x, \beta) = x\beta$ and an anisotropic separable Gaussian correlation function, $c(x_i, x_j) = \exp[-\Sigma_{k=1}^{d}\frac{(x_{ik}-x_j)^2}{\theta_k}]$, where d is the dimension of the input space and θ_k is the range parameter for each dimension. For a deterministic simulator, $\eta(x)$ is typically omitted, although Gramacy and Lee (2012)

advocates for always including this term, and we follow that approach.

A Treed Gaussian Process (TGP) (Gramacy and Lee, 2008) is a more flexible non-stationary process which divides up the input space and fits different base Gaussian Process models to data independently in each subregion. The model has the advantage that it can model discontinuities since the partitioning model is more flexible and it can fit different submodels around the discontinuities and changes in the correlation structure. The second advantage of the partitioning model is that it is more computationally efficient. If the sample size is n, the matrix inversion for a GP requires $O(n^3)$ computation time. By partitioning the whole tree into smaller subregions, matrix inversions are now carried out for smaller sample sizes. The tree grows via recursive binary partitioning, following Chipman et al. (1998, 2010). In each partition, a GP model is fit independently. A tree T recursively divides the data space into R distinct subregions. For each subregion in the tree partition r_v, the hierarchical GP model with linear mean is:

$$Z_v|\beta_v, \sigma_v^2, K_v \sim N_{nv}(F_v\beta_v, \sigma_v^2 K_v) \qquad \beta_0 \sim N_{d+1}(\mu, B)$$

$$\beta_v|\sigma_v^2, \tau_v^2, W, \beta_0 \sim N_{d+1}(\beta_0, \sigma_v^2\tau_v^2 W) \qquad \tau_v^2 \sim IG(\alpha_\tau/2, q_\tau/2)$$

$$\sigma_v^2 \sim IG(\alpha_\sigma/2, q_\sigma/2) \qquad W^{-1} \sim Wish((pV)^{-1}, \rho)$$

with $F_v = (1, X_v)$, and W is a $(d+1)\times(d+1)$ matrix. The N, IG, and $Wish$ are the Normal, Inverse-Gamma, and Wishart distributions, respectively. Hyperparameters μ, B, V, ρ, α_σ, q_σ, α_τ, q_τ are deemed as known. K_v is a $n_v * n_v$ correlation matrix, where n_v is the number of observations in region r_v. The mean function coefficients β_v are modeled hierarchically with a common unknown mean β_0 and variance for each subregion $\sigma_v^2\tau_v^2$ (Gramacy and Lee, 2008). We use the tgp package in R for implementing the Treed Gaussian Process and related analyses (Gramacy, 2007).

2.3 Sensitivity indices

Sensitivity analysis is the study of how uncertainty in the output of a model can be apportioned to different sources of uncertainty in the model input. In other words, it is a technique that systematically changes the parameters of a model to determine the effects of such changes. Sensitivity analysis is useful for computer modeling for several different aims, such as making recommendations for decision makers, increasing understanding or quantification of the system and model development (Saltelli et al., 2008). Usually, sensitivity analysis is measured by the main effects and the total effects. The first order sensitivity indices measure main (linear only) effects while the total sensitivity indices measure total effects, which are the main effects plus interaction effects, quadratic effects, and all higher order effects. A first order main effect is $V[E(Y|X_i)]$. The conditional expectation $E(Y|X_i)$ can be calculated empirically by cutting the X_i domain into slices and averaging the values of $(Y|X_i)$ within the same slice X_i. In this way, if the scatterplot has a pattern, the conditional expectation $E(Y|X_i)$ has a large variation across X_i values and the factor X_i is considered as important. The first order sensitivity index for variable X_i is obtained by dividing this term by the total unconditional variance. The total effect for the input variable X_i is linked to $E[V(Y|X_{-i})]$, which is the remaining variance of Y that would be left, on average, if we could determine the true value of every factor except X_i. Dividing this term by the total unconditional variance yields the total sensitivity index for variable X_i (Saltelli et al., 2008).

Saltelli et al. (2008) and Saltelli et al. (2010) describe a Monte-Carlo based numerical procedure for computing first order and total effect sensitivity indices. Assume the input (N, k) matrix, where N is the number of rows and k is the number of columns, is the

following:

$$A = \begin{pmatrix} x_1^1 & x_2^1 & ... & x_i^1 & ... & x_k^1 \\ x_1^2 & x_2^2 & ... & x_i^2 & ... & x_k^2 \\ ... & ... & ... & ... & & \\ x_1^{N-1} & x_2^{N-1} & ... & x_i^{N-1} & ... & x_k^{N-1} \\ x_1^N & x_2^N & ... & x_i^N & ... & x_k^N \end{pmatrix}$$

A (N, k) resampling matrix is generated which has the same distribution in each dimension as the input

$$B = \begin{pmatrix} x_{k+1}^1 & x_{k+2}^1 & ... & x_{k+i}^1 & ... & x_{2k}^1 \\ x_{k+1}^2 & x_{k+2}^2 & ... & x_{k+i}^2 & ... & x_{2k}^2 \\ ... & ... & ... & ... & & \\ x_{k+1}^{N-1} & x_{k+2}^{N-1} & ... & x_{k+i}^{N-1} & ... & x_{2k}^{N-1} \\ x_{k+1}^N & x_{k+2}^N & ... & x_{k+i}^N & ... & x_{2k}^N \end{pmatrix}$$

A third matrix B_A^i is generated with the i_{th} column of B being replaced by the i_{th} column of A:

$$B_A^i = \begin{pmatrix} x_{k+1}^1 & x_{k+2}^1 & ... & x_i^1 & ... & x_{2k}^1 \\ x_{k+1}^2 & x_{k+2}^2 & ... & x_i^2 & ... & x_{2k}^2 \\ ... & ... & ... & ... & & \\ x_{k+1}^{N-1} & x_{k+2}^{N-1} & ... & x_i^{N-1} & ... & x_{2k}^{N-1} \\ x_{k+1}^N & x_{k+2}^N & ... & x_i^N & ... & x_{2k}^N \end{pmatrix}$$

A fourth matrix A_B^i is generated with the i_{th} column of A being replaced by the i_{th} column of B:

$$A_B^i = \begin{pmatrix} x_1^1 & x_2^1 & ... & x_{k+i}^1 & ... & x_k^1 \\ x_1^2 & x_2^2 & ... & x_{k+i}^2 & ... & x_k^2 \\ ... & ... & ... & ... & & \\ x_1^{N-1} & x_2^{N-1} & ... & x_{k+i}^{N-1} & ... & x_k^{N-1} \\ x_1^N & x_2^N & ... & x_{k+i}^N & ... & x_k^N \end{pmatrix}$$

Four model output vectors are calculated based on these four matrices:

$$y_A = f(A) \qquad y_B = f(B) \qquad y_{BA^i} = f(B_A^i) \qquad y_{AB^i} = f(A_B^i)$$

The formula for calculating first-order sensitivity indices is:

$$S_i = \frac{V[E(Y|X_i)]}{V(Y)} = \frac{(1/N)\Sigma_{j=1}^N y_A^{(j)} y_{BA^i}^{(j)} - f_0^2}{(1/N)\Sigma_{j=1}^N (y_A^{(j)})^2 - f_0^2},$$

where $f_0^2 = (\frac{1}{N}\Sigma_{j=1}^N y_A^{(j)})^2$. The formula for calculating total sensitivity indices is:

$$S_{T_i} = 1 - \frac{V[E(Y|X_{-i})]}{V(Y)} = 1 - \frac{(1/N)\Sigma_{j=1}^N y_A^{(j)} y_{AB^i}^{(j)} - f_0^2}{(1/N)\Sigma_{j=1}^N (y_A^{(j)})^2 - f_0^2}.$$

3 Multi-stage variable selection tool

The computer modeling literature has traditionally dealt with relatively few possible input variables. However, with the rapid increase in the role of computer modeling and simulation in scientific discovery, there can be a large number of input variables. Due to the lack of knowledge of the underlying physical process, a simulation model with high credibility can only be built through rigorous model validation. Variable selection plays an important role in the model validation process. For some complex phenomena, such as those which

simultaneously model multiple physical and chemical processes, the simulators tend to have several hundreds or thousands of input variables. However, in practice only a few (perhaps less than 10) of them are typically significantly related to the factors controlling the underlying mechanism of the physical processes. Complicating the matter, these variables are often non-linearly related to the response. Existing statistical methods for variable selection struggle to effectively identify a few variables out of hundreds with high accuracy with only a moderate number of samples. A multi-stage strategy is designed which employs a state-of-the-art technique for each of the stages. By making the best use of each technique, in combination they can accomplish a sophisticated goal that can not be achieved separately. In the first-stage, an extension of the sum of trees method is developed to reduce the total number of variables to around 30. In the second stage, a sequential sampling strategy is used to sample more data points to search for variables with small but significant contribution to prediction. In the last stage, sensitivity analysis is combined with two stopping rules to reduce the number of variables further as needed.

3.1 The extended sum of trees method

The first stage is based on the BART sum of trees model, and is implemented with the number of trees set to 10. The rough estimate of the standard deviation of σ is the standard deviation of y. During a preliminary study, we found that the sum of trees model performed better for datasets whose input points followed a Latin Hypercube design (Tang, 2008) than datasets with uniform initial distribution. Thus, the Latin Hypercube design is adopted when generating simulated datasets.

We extended the sum of trees model to make use of large sample theory. For each run of BART, when using a small number of trees, the splitting frequencies will be high for those variables strongly associated with the response. However, there is a random variation among different runs in the splitting frequencies for the same variable. Therefore, the variables

selected might be inconsistent between the first run and the second run. By the property of large sample theory, if BART is run many times, the mean/sum of the splitting frequencies of each variable will reach its underlying true value. If the ranks of these means/sums are used instead of the ranks of the splitting frequencies of a single run, the accuracy in ranking will be remarkably improved (Chipman et al., 2010). Consequently, the extended method of the sum of trees is proposed to repeatedly run BART 10 times to keep the top 50 variables with the highest splitting frequencies each time. Then cumulative splitting frequencies from those runs are reckoned for each variable. The top 25 variables with the highest cumulative splitting frequencies are singled out together with an extra 5 variables showing the strongest linear associations (because BART is more focused on finding nonlinear relationships, so simple linear relationships can get overlooked). By combining the sum of trees with linear regression, the extended method can select variables with either a strong linear association or a complicated non-linear relationship. We keep the top 30 variables at this stage to try to ensure that we include all potentially important variables, as we do not expect to have more than that many; should more be expected, then more should be retained.

3.2 Adaptive sampling

In some cases, we may be able to collect more data through a sequential experiment. A computer simulation experiment is a classic example where additional runs can be obtained iteratively after analyzing the existing data. In such cases, we want to carefully select the additional runs to maximize the information gained. We design a strategy to sample more data points efficiently to best identify which of the remaining variables truly are relevant. We bring in ideas from machine learning to find non-linear effects.

The ALM criterion (MacKay, 1992; Seo et al., 2000) chooses the data point with the greatest standard deviation in the predicted output. With a nonlinear model, in practice we sample by selecting the data point with the largest 95% predictive interval, as that approach

is more robust for irregular predictive distributions. Sampling that location will provide the most information available about the response surface. We use BART to compute the posterior distribution of the predicted output. 10,000 candidate data points are generated from a Latin Hypercube design, and the one with the longest 95% predictive interval is chosen and a new datapoint is collected at that set of inputs. The sampled data point is then added to the original observations for making new predictions in BART. A moderate number of data points, such as 50-150, are sampled in this manner. BART provides a computationally efficient model to enable quick fitting after each update.

3.3 Two Stopping rules

After selecting 30 variables out of the initial set, we use sensitivity analysis to rank the 30 variables selected. But a question remaining is when to stop removing variables? How many of the variables are true predictors? We provide two stopping rules here based on the sensitivity analysis. To obtain an accurate sensitivity analysis including nonlinear effects, we use the fit from the TGP model, and the sensitivity analysis from the tgp package, function `sens()` (Gramacy and M., 2010). We found that we obtain a more accurate sensitivity analysis using TGP modeling than using BART modeling.

First stopping rule: The first stopping rule used here is to run the sensitivity analysis in the TGP model 10 times and select those variables which consistently rank highly in these 10 runs. The first stopping rule is to pick variables satisfying the following criteria: (1) over the 10 runs, the maximum rank minus the minimum rank is less than 5; (2) disregarding the lowest 2 ranks, the maximum of the remaining 8 ranks minus the minimum of the remaining 8 ranks is less than 4; (3) disregarding the lowest 3 ranks, the maximum of the remaining 7 ranks minus the minimum of the remaining 7 ranks is less than 3. A variable is considered as significant if any one of the three conditions is satisfied. The last two criteria are similar

to the first criterion in principle, but they assume that there are at most two (the second criterion) or three (the third criterion) outliers.

Second stopping rule: The second rule is based on Linkletter et al. (2006), substituting one input variable with an inert variable each time and comparing the change in the posterior distribution of the sensitivity indices before and after the substitution. The differences of the mean of the posterior distributions of the sensitivity indices are computed. Those variables which yield a difference in mean posterior sensitivities bigger than 0.008 are considered as the true predictors.

In both cases, the values were determined empirically, after much experience with a variety of examples.

4 Simulated Examples

A dataset was generated for testing the performance of our proposed method, where the identity of the true variables was kept unknown to the user as the method was run (hence the irregular numbering):

$$y = 2x_1^3 - 3\sin[5\pi x_6] - 2(1 - x_9) \times (1 - x_{10}) + 4(x_{43} - 0.5)^2 - x_{88} + x_{89}$$

The dataset has 100 variables and 150 observations in the initial run. The input values were chosen as a random Latin hybercube. No noise was added, in order to focus on the complex function itself (and also keeping with the standard approach in deterministic computer code emulation). The extension of the sum of trees method is used to reduce the total number of variables to 30. Then sensitivity analysis in the tgp package is used to rank the

30 variables selected and the stopping rules are used. The results with and without adaptive sampling are compared and summarized as follows.

For this first test dataset, the first 30 variables selected are (highest rank first, the bolded ones are the truly significant ones):

Ordered by main effects : **88 9 89 1** 93 96 17 3 94 48 47 14 24 51 **6** 77 50 87 62 **10** **43** 15 35 72 26 74 49 39 8 40
Ordered by total effects: **88 9 1 89** 93 87 17 96 47 94 74 49 77 24 62 72 8 3 **43** 14 35 50 39 51 48 15 40 26 **6 10**

Figure 1 illustrates the boxplots for the posterior distribution of sensitivity indices generated by the sensitivity analysis in the tgp package for this dataset. When using the sensitivity indices for the total effects, the first stopping rule selects variables 88,9,1,89,93,17,96 (highest rank first). The second stopping rule selects variables 1,9,17,62,88,89,93. So far, we are missing predictors 6, 10, and 43, and erroneously including 9,17,62,93 and 96.

We then conduct sequential sampling to better refine our variable selection. 100 additional data points are sampled via the ALM algorithm. For each new data point, 10,000 candidates are searched. Table 1 presents our results. After obtaining 100 additional points, the first criterion of the first stopping rule selects variables 1, 10, 43, 88, and 89, and the second criterion adds in variable 9. We have found these results to be repeatable, in that if a different random seed and different starting Latin Hypercube Design are used, this sequential process will usually select the same sets of variables as that in Table 1.

We are pleased that the sequential process is able to filter out the unimportant variables and identify nearly all of important variables. Variable 43 is difficult to find because its first order effect is zero. Variable 6 is the one variable that we miss, as it also has a zero first order

effect, and has a small total effect, and was designed to be difficult to find. Significantly more data would be needed to detect this variable.

We now consider a second test dataset with 15 relevant variables out of 200, again with the truth kept hidden from the person implementing the algorithm. The true function is:

$$\begin{aligned} y &= 3x_3 + 4/(x_{17}+1) + 2\log(x_{23}+1) - 3(x_{29}+.2)(x_{34}+.2)(x_{181}+.3) + 2x_{43} - 4x_{64}^2 \quad (3) \\ &\quad -2x_{87} + 2.1\sin(x_{92}\times\pi\times 1.2) - 2.8\sin(x_{100}\times\pi) - (x_{110}+.7)^2 \quad (4) \\ &\quad -2.8I(x_{141}<.6) + 2.5I(x_{155}>.6) - 2.7\times x_{199} \quad (5) \end{aligned}$$

where $I(\cdot)$ is the indicator function which takes value one when its argument is true and value zero otherwise. No noise was added as we were attempting to simulate a deterministic computer experiment. The first stage of our analysis identifies these 30 variables as potentially important and worthy of further exploration (highest rank first, the bolded ones are the truly significant ones):

Ordered by main effects: **141 155 64 199 3 87 110 92 43 34 181 17 23** 125 130 146 105 165 62 172 **29** 137 163 52 186 162 103 **100** 180 73

Ordered by total effects: **141 155 64 199 87 3 43 110 34 92 181 23 17** 125 62 146 52 105 163 186 **100** 180 172 165 73 137 103 **29** 130 162

Figure 2 is similar to Figure 1 except that it is generated for the second test dataset. The first stopping rule, which uses the sensitivity indices for the total effects, selects variables 141, 155, 64, 199, 3, 87, 110, 43, 34, 17, 92, 181, 23, 62 (highest rank first).The second stopping rule uses the sensitivity indices for the main effects and selects variables 3, 17, 23, 29, 34, 43, 64, 87, 92, 100, 110, 141, 155, 181, 186 and 199. It picks all the correct variables

and erroneously includes variable 186.

Sequential sampling via the ALM algorithm is used to obtain an additional 100 data points. Then the first stopping rule is used for analyzing the dataset, picking out variables: 3, 17, 23, 29, 34, 43, 64, 87, 92, 110, 141, 155, 181, 199. The ranking of the variables picked is indicated in Table 2. All the other variables are picked by the first criterion of the first stopping rule except variable 92, which is picked by the second criterion of the stopping rule. In the column corresponding to variable 92, after rank 24 is disregarded, for the remaining nine ranks, the maximum rank (23) minus the minimum rank (18) is 5. After adaptive sampling, variable 29 is now clearly labeled as relevant. It had a lower ranking picked originally since it is only part of a three way interaction effect with variable 34 and variable 181, which is difficult to detect. The extraneous variable 186 is now excluded. The results are almost perfect except for the omission of variable 100, which is associated with a sine function. Like in the first example, this mean-zero periodic effect is difficult to detect without yet more data.

5 Analyzing the Community and Crime dataset

We demonstrate the main part of the methodology on a crime dataset. The Communities and Crime dataset was taken from UC-Irvine Machine Learning Repository (http://archive.ics.uci.edu/ml/datasets.html). The data combines socio-economic data from the 1990 US Census, law enforcement data from the 1990 US LEMAS survey, and crime data from the 1995 FBI UCR. The data has 128 attributes and 1994 observations. The 128th variable is the total number of violent crimes per 100K population, which is the goal attribute to be predicted. The other 127 variables are treated as predictors.

Using the extension of the sum of trees method, the 30 variables selected are: 1, 8, 9, 11, 17, 21, 23, 30, 33, 38, 39, 43, 45, 46, 48, 49, 50, 51, 54 , 55, 74, 76, 77, 79, 82, 84, 86, 91,

94, and 95. These 30 variables are analyzed by the sensitivity analysis in the TGP model, shown in Figure 3, and the first stopping rule is applied. Overall five variables are selected. The 33_{th}, 55_{th} and 77_{th} are clearly significant, and the 8_{th} and the 50_{th} are also found to be significant. The following are the social meaning of these five variables:

8: percentage of population that is African American.

33: number of people under the poverty level.

50: percentage of kids in family housing with two parents.

55: number of kids born to parents who were never married.

77: number of vacant households.

If the 30 variables selected by the extension of the sum of trees method are used as explanatory variables, fitting a standard linear regression model yields residuals with variance 0.018. When following our analysis and reducing the inputs to the five variables listed above, fits with the TGP and BART packages yield MSEs of residuals of 0.0191 and 0.0185 respectively. We also performed 10-fold cross validation with the same five explanatory variables and the BART package, which is the package used when implementing the 10-fold cross validations throughout this section. For each of the 10 iterations, the variance of the prediction errors is calculated and the average of the 10 variances is reported, which is 0.0211.

We compared our results to several other available variable selection techniques. When using the command 'bootFreq' in the R package 'MMIX' (Morfin and Makowski, 2012) to analyze the data with the number of samples is set to 50, the top five variables selected are:

8: percentage of population that is African American.

17: percentage of people living in areas classified as urban.

96: number of homeless people counted in the street.

88: rental housing-lower quartile rent.

54: percentage of moms of kids under 18 in labor force.

When using these five variables as explanatory variables, 10-fold cross validation yields BART

fits with mean variance of prediction errors of 0.0249. We see that our method performs better in selecting the significant variables with potentially nonlinear effects.

When using the R package 'spikeslab' (Ishwaran et al., 2013) to analyze the data, the top five variables selected are:

8: percentage of population that is African American.

56: percentage of kids born to parents who were never married.

50: percentage of kids in family housing with two parents.

88: rental housing-lower quartile rent.

44: percentage of males who are divorced.

When using these five variables as explanatory variables, 10-fold cross validation yields mean variance of prediction error of 0.0218, again larger than from our proposed method. Furthermore, spikeslab does not specify how many variables are the truly significant ones, so our proposed method has obvious advantages over 'spikeslab'.

When using the R package 'foba' (Zhang, 2008) to analyze the data, the top five variables selected are:

8: percentage of population that is African American.

50: percentage of kids in family housing with two parents.

70: mean persons per household.

77: number of vacant households.

97: percent of people foreign born.

When using these five variables as explanatory variables, 10-fold cross validation yields mean variance of prediction error equaling 0.0195, which is actually slightly better than that of our proposed method. However, the 'foba' package does not specify the number of significant variables, so having a specific selection of variables is an advantage of our method.

6 Discussion

The extension of the sum of trees method performs well in picking the top 30 variables which have the highest association with the response, with very high probability of including all of the true predictors. An adaptive sampling design using the ALM criterion can detect a broad range of linear and nonlinear effects. The two stopping rules based on sensitivity analysis both work very well in determining the total number of true predictors. Their performance benefits from the high accuracy of ranking the sensitivities of the variables by the sensitivity analysis in the TGP model.

For future research work, we can focus on how to improve the accuracy of the sensitivity analysis so that it can detect smaller effects without the adaptive sampling step. Another direction is to design a test for comparing the equality of two posterior distributions. This test can be used in the second stopping rule. Other adaptive sampling criteria can be explored to improve the learning during sequential sampling.

Table 1: The ranking of the variables picked by the first stopping rule for the test dataset of sample size of 150 and dimension of 100 by the ALM strategy using 10,000 candidates for each sampling points.

variable	1	9	10	43	88	89
1	30	21	25	27	28	29
2	30	20	27	25	28	29
3	30	26	23	25	28	29
4	30	26	23	27	28	29
5	30	25	27	26	28	29
6	30	25	27	26	28	29
7	30	26	27	25	28	29
8	30	25	26	27	28	29
9	30	13	27	26	28	29
10	30	26	24	27	28	29

Table 2: The ranking of the variables picked by the first stopping rule for the test dataset with sample size of 300 and dimension of 200 by the ALM strategy

variable	3	17	23	29	34	43	64	87	92	110	141	155	181	199
1	26	23	18	17	20	19	29	24	21	25	30	28	22	27
2	27	22	18	17	21	23	29	24	20	25	30	28	19	26
3	26	22	18	17	20	21	29	24	23	25	30	28	19	27
4	26	23	17	16	19	21	29	25	18	24	30	28	20	27
5	26	21	17	16	22	19	28	25	23	24	30	29	20	27
6	26	23	18	17	22	19	29	25	20	24	30	28	21	27
7	27	21	18	16	19	22	29	23	24	25	30	28	20	26
8	27	23	18	17	20	22	28	25	19	24	30	29	21	26
9	26	22	17	16	19	21	28	25	23	24	30	29	20	27
10	26	21	17	16	20	22	29	25	23	24	30	28	18	27

Figure 1: Boxplots of the posterior distribution of sensitivity indices for the main effects (a) and total effects (b) for the sensitivity analysis in the tgp package for the first test dataset.From left to right, the 30 variables are: (1_{st},3_{rd},6_{th},8_{th},9_{th},10_{th},14_{th},15_{th},17_{th}, 24_{th},26_{th},35_{th},39_{th},40_{th},43_{th},47_{th},48_{th},49_{th},50_{th},51_{th},62_{th},72_{th},74_{th},77_{th},87_{th},88_{th},89_{th},93_{th},94_{th},96_{th})

Figure 2: Boxplots of posterior distribution of sensitivity indices for the main effects (a) and total effects (b) for the sensitivity analysis in the tgp package for the second test dataset(from left to right, the 30 variables are: 3_{rd},17_{th},23_{th},29_{th},34_{th},43_{th},52_{th},62_{th},64_{th},73_{th},87_{th},92_{th},100_{th},103_{th},105_{th},110_{th},125_{th},130_{th},137_{th}, 141_{th},146_{th},155_{th},162_{th},163_{th},165_{th},172_{th},180_{th},181_{th},186_{th},199_{th}).

Figure 3: Boxplots of posterior distribution of sensitivity indices for the main effects (a) and total effects (b) for the sensitivity analysis in the tgp package for the communities and crime dataset.

References

Bondell, H., Reich, B., and Storlie, B. (2009), "Variable selection in Bayesian Smoothing Spline ANOVA models: application to deterministic computer codes." *Technometrics*, 51, 110–119.

Chipman, H. and McCulloch, R. (2010), *BayesTree: Bayesian Methods for Tree Based Models*, R package version 0.3-1.1.

Chipman, H., George, E., and McCulloch, R. (1998), "Bayesian CART model Search," *Journal of the American Statistical Association*, 93, 935–960.

Chipman, H., George, E., and McCulloch, R. (2010), "BART: Bayesian Additive Regression Trees," *Annals of Applied Statistics*, 4, 266–298.

Gramacy, R. and Lee, H. K. H. (2012), "Cases for the nugget in modeling computer experiments," *Statistics and Computing*, 22, 713–722.

Gramacy, R. B. (2007), "tgp: An R Package for Bayesian Nonstationary, Semiparametric Nonlinear Regression and Design by Treed Gaussian Process Models," *Journal of Statistical Software*, 19, 1–46.

Gramacy, R. B. and Lee, H. K. H. (2008), "Bayesian Treed Gaussian Process model with an application to Computer Modeling," *Journal of the American Statistical Association*, 103, 1119–1130.

Gramacy, R. B. and Lee, H. K. H. (2009), "Adaptive design of supercomputer experiments," *Technometrics*, 51, 130–145.

Gramacy, R. B. and M., T. (2010), "Categorical Inputs, Sensitivity Analysis, Optimization and Importance Tempering with tgp Version 2, an R Package for Treed Gaussian Process Models," *Journal of Statistical Software*, 33, 1–48.

Hastie, T. and Tibshirani, R. (2000), "Bayesian Backfitting," *Statistics Science*, 15, 196–223.

Ishwaran, H., Rao, J. S., and Kogalur, U. B. (2013), *spikeslab : Prediction and variable selection using spike and slab regression*, R package version 1.1.5.

Linkletter, C., Bingham, D., Hengartner, N., Higdon, D., and Ye, K. (2006), "Variable selection for Gaussian process models in computer experiments," *Technometrics*, 48, 478–490.

MacKay, D. J. C. (1992), "Information-based Objective Functions for Active Data Selection," *Neural Computation*, 4, 589–611.

Morfin, M. and Makowski, D. (2012), *MMIX: Model selection uncertainty and model mixing*, R package version 1.2.

Morris, M. (1991), "Factorial Sampling Plans for preliminary computational experiment," *Technometrics*, 33, 161–174.

Oakley, J. (2009), "Decision-Theoretic Sensitivity Analysis for complex Computer Models," *Technometrics*, 51, 121–129.

Sacks, J., Welch, W., Mitchell, T., and Wynn, H. (1989), "Design and Analysis of Computer Experiments," *Statistical Science*, 4, 409–435.

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S. (2008), *Global sensitivity analysis: The primer*, John Wiley and Sons.

Saltelli, A., Annoni, P., I., A., Campolongo, F., Ratto, M., and Tarantola, S. (2010), "Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index," *Computer Physics Communications*, 181, 259–270.

Santner, T., Williams, B., and Notz, W. (2003), *The design and analysis of computer experiments*, Springer, New York.

Seo, S., Wallat, M., Graepel, T., and Obermayer, K. (2000), "Gaussian Process Regression: Active Data Selection and Test Point Rejection," in *Proceedings of the International Joint Conference on Neural Networks*, vol. III, pp. 241–246, IEEE.

Taddy, M., Lee, H. K. H., Gray, G. A., and Griffin, J. D. (2009), "Bayesian guided pattern search for robust local optimization," *Technometrics*, 51, 130–145.

Tang, B. (2008), *Latin Hypercube Designs*, Wiley online library.

Thompson, S. and Seber, G. (1996), *Adaptive Sampling*, John Wiley and Sons.

Zhang, T. (2008), *foba: greedy variable selection*, R package version 0.1.