Local Variable Selection in Sequential optimization

Munir Winkel, Jonathan W Stallings, and Brian J Reich
Department of Statistics, North Carolina State University

Abstract

Optimizing a function using a sequential design is challenging when the function is defined over a high-dimensional design space. Expected improvement algorithms, which balance exploration of the design space with honing in on a global maximum, struggle in high dimensions because estimating the function and its maximum well require a large number of observations. Reducing the dimension of the design space should improve estimation and lead to faster identification of the maximum. However, current variable selection techniques are global; a variable is either in or out of the design matrix. In this paper, we define a measure of local importance to identify which variables are active around regions of local maxima, and we design a method to efficiently search the design space and estimate a global maximum. We present simulation studies involving high-dimensional data and compare the proposed global and local variable selection approach with other methods in terms of their ability to estimate the global maximum. In the simulation study, we show that local variable selection takes fewer steps to estimate a global maximum.

Keywords: Bayesian, Variable Selection, Computer Experiments, Expected Improvement

1. Introduction

Our work is motivated by a material science study where the objective is to find the combination of atomic properties with the optimal tribological properties. Each experiment in that study uses different types of nanoparticles, immersed in different liquids and placed on different substrates on a quartz crystal microbalance (QCM). Varying the nanoparticles, liquids, and substrates can be thought of as adjusting the levels of the corresponding atomic properties; and the quantitative results from each experiment are estimates of the tribological properties. Since each experiment is time-consuming and costly, it is critical to carefully choose the levels of the atomic properties. To that end, we use a sequential design that performs interim analyses and proposes the next experimental setting that maximizes the expected information about the underlying tribological properties.

It is challenging to find the optimal settings for the next experiment, especially when

the number of inputs (the atomic properties of the nanoparticles, substrates, and liquids) is large. Variable selection to reduce the dimension of the design space is appealing both statistically and operationally. We anticipate that a majority of the atomic properties will have some effect on the underlying tribological properties across the entire design space, but that our sequential design approach will eventually propose experimental settings in a local neighborhood around the optimal value, where only a few atomic properties will be influential. This leads to our statistical development of a sequential design algorithm that optimizes a high-dimensional function using both global and local variable selection.

To formalize the problem mathematically, denote $Y(\mathbf{x})$ as the scalar response obtained from conducting an experiment at input $\mathbf{x} = (x_1, ..., x_p)$. We assume

$$Y(\mathbf{x}) = f(\mathbf{x}) + \epsilon \tag{1}$$

for some underlying response surface $f(\cdot)$ and errors $\epsilon \overset{iid}{\sim} N(0, \tau^2)$. The objective is to find the inputs $\mathbf{x}$ that maximize the response surface $f(\mathbf{x})$.

In a sequential design, an initial set of experiments is conducted. Before conducting experiment i, the data from the first $i - 1$ experiments are used to estimate the response surface f and its uncertainty. This information guides which inputs are chosen for the next experiment. For example, Jones [2001] suggests that the response surface can be estimated using basis functions, cubic splines, or Gaussian process models. The experiment that is chosen next maximize an optimality criteria, such as the expected improvement criterion, which balances exploring the design space with honing in on regions near the observed maxima (Močkus [1975], Jones et al. [1998]). See Brochu et al. [2010] for a discussion of other utility functions, including probability of improvement.

Though these tools are powerful, optimizing a function remains challenging in higher dimensions (Shan and Wang [2010]). There is extensive literature available on variable selection. In higher dimensions, several variables can be "inactive," in the sense that varying their inputs does not affect the functional response. Most variable selection procedures are global, where variables are declared to be inactive across the entire $[0, 1]^p$ space. Bai et al. [2014] emphasize that in nonlinear settings, whether a variable is active or not is a "local

concept." Bai et al. [2014] allow for the set and size of locally active variables to vary across the design space and propose two approaches for local variable selection. The first approach assumes a local linear model and uses the magnitude of the partial derivatives as a guide to variable selection. The second approach uses the change in prediction as a measure of local importance, where predictions are made using locally-weighted kernels and the closest design points in each dimension.

In this paper, we introduce a new measure of local importance and as well as a new algorithm for reducing the dimension of the search space. In a simulation study, we show how local variable selection, in conjunction with these other tools, allows scientists to more effectively explore the design space and identify the best input to maximize the unknown underlying function.

2. Gaussian Process Regression

Once data have been collected, surrogate models can be used to make predictions for any arbitrary input. This will guide the choice of input for the next experiment. Following Sacks et al. [1989], we use Gaussian process (GP) regression to build a surrogate model, since GP regression can approximate highly non-linear functions well. We model f as a GP with mean function $E[f(\mathbf{x})] = \mu(\mathbf{x})$ and covariance function $\text{Cov}[f(\mathbf{x}), f(\mathbf{x}')] = \sigma^2 K(\mathbf{x}, \mathbf{x}')$. While it is possible to specify polynomial mean functions, Welch et al. [1992] argues that a constant mean function is sufficient for interpolating the response surface between observed design points, so we set $\mu(\mathbf{x}) \equiv \mu$ for all $\mathbf{x}$. The covariance function dictates how much information to borrow across observations that are nearby. We assume that the true response surface is smooth, and we select the infinitely-differentiable squared exponential covariance function [Sacks et al., 1989]

$$K(\mathbf{x}, \mathbf{x}') = \exp\left\{ -\sum_{k=1}^{p} \gamma_k \left(x_k - x_k'\right)^2 \right\} \tag{2}$$

where $\gamma_1, ..., \gamma_p \geq 0$ are the correlation range parameters. If $\gamma_k = 0$, then the k^{th} input is independent of the response (conditional on all other inputs remaining the same). If γ_k is large then the response surface is sensitive to changes in the k^{th} input variable, since only

observations close by are correlated with each other.

In addition to the covariance of the response surface, we include the nugget term ϵ in (1) to account for random experimental variation unrelated to the mean response $f(\mathbf{x})$. Even for deterministic functions where $Y(\mathbf{x}) = f(\mathbf{x})$, including a nugget effect with $\text{Var}(\epsilon) = \tau^2 > 0$ can be used to stabilize computations by avoiding computational issues involving non-singular matrices [Gramacy and Lee, 2009]. By assuming an error process independent of the underlying function, we have

$$\text{Cov}[Y(\mathbf{x}), Y(\mathbf{x}')] = \sigma^2 K(\mathbf{x}, \mathbf{x}') + \tau^2 1_{\{\mathbf{x}=\mathbf{x}'\}} \equiv V(\mathbf{x}, \mathbf{x}'). \tag{3}$$

The Gaussian process model can be used for prediction at a new input $\mathbf{x}$ given a sample of n observations with inputs $\mathbf{x}_1, ..., \mathbf{x}_n$ and corresponding outcomes $Y(\mathbf{x}_1), ..., Y(\mathbf{x}_n)$. We denote $\mathbf{V_X}$ to be the n x n covariance matrix of the n observations $\mathbf{y} = [Y(\mathbf{x}_1), ..., Y(\mathbf{x}_n)]^T$ corresponding to the observed n x p matrix of inputs $\mathbf{X} = [\mathbf{x}_1^T, ..., \mathbf{x}_n^T]^T$. Denote $\mathbf{v}(\mathbf{x}) \equiv [V(\mathbf{x}_1, \mathbf{x}), ..., V(\mathbf{x}_n, \mathbf{x})]^T$ as the covariance between input $\mathbf{x}$ and the observed inputs. For any input $\mathbf{x}$, it holds that $Y(\mathbf{x}) \mid \mathbf{y}$ is a Gaussian process with prediction mean and variance

$$\begin{aligned} \hat{Y}(\mathbf{x}) &= \mu + \mathbf{v}(\mathbf{x})^T \mathbf{V_X}^{-1}(\mathbf{y} - \mu \mathbf{1}_n), \\ s^2(\mathbf{x}) &= \sigma^2 - \mathbf{v}(\mathbf{x})^T \mathbf{V_X}^{-1} \mathbf{v}(\mathbf{x}). \end{aligned} \tag{4}$$

To complete the Bayesian model we specify priors for the parameters. For prior specification we reparameterize to the total precision (inverse variance), $\eta = 1/(\sigma^2 + \tau^2)$, and proportion of variance from the response surface, $r = \dfrac{\sigma^2}{\sigma^2 + \tau^2}$. The priors are then $\eta \sim \text{Gamma}(a_\eta, b_\eta)$, $r \sim \text{Uniform}(0, 1)$, and $\mu \sim N(0, \sigma_\mu^2)$, where the hyperparameters a_η, b_η, and σ_μ^2 are set to give uninformative priors. The priors for the correlation parameters $\gamma_1, ..., \gamma_p$ are given in Section 3.1.

3. Sequential Optimization using Expected Improvement

Jones et al. [1998] introduced the efficient global optimization (EGO) algorithm which balances exploring the unobserved design space to improve estimates of the response sur-

face, with honing in on areas around an observed maximum. The EGO algorithm was built for deterministic computer simulations, where experiments repeated at the same input settings yield identical responses. As introduced by Močkus [1975], we define the improvement at any arbitrary input as $I(\mathbf{x}) \equiv \max\{Y(\mathbf{x}) - Y(\mathbf{x}_{opt}), 0\}$, where $Y(\mathbf{x}_{opt}) = \max\{Y_1, ..., Y_n\}$. Since the exact improvement is unknown, the EGO algorithm instead uses the surrogate model to compute the expected improvement (EI) $EI(\mathbf{x}) = E[I(\mathbf{x})]$, which Jones et al. [1998] shows can be written as

$$EI(\mathbf{x}) = s(\mathbf{x}) \left\{ Z(\mathbf{x}) \Phi \left[Z(\mathbf{x}) \right] + \phi \left[Z(\mathbf{x}) \right] \right\}, \tag{5}$$

where $Z(\mathbf{x}) \equiv \dfrac{Y(\mathbf{x}) - Y(\mathbf{x}_{opt})}{s(\mathbf{x})}$ and $\Phi(\cdot)$ and $\phi(\cdot)$ are the CDF and PDF of a standard normal distribution, respectively. The next input evaluated is the one that maximizes EI, i.e., $\mathbf{x}_{n+1} \equiv \arg\max_{\mathbf{x}} E[I(\mathbf{x})]$. In the deterministic case, with $\tau = 0$, a input already selected will never be selected again, since its EI is exactly zero.

We will use the Augmented Expected Improvement (AEI) criterion [Huang et al., 2006], which is more appropriate for non-deterministic functions. Using the parameterization in (3) and given an interest in maximization, the AEI criterion is defined as

$$AEI(\mathbf{x}) \equiv E\left[\max\{Y(\mathbf{x}) - \hat{Y}(\mathbf{x}_{opt}), 0\} \right] \left(1 - \frac{\tau}{\sqrt{s^2(\mathbf{x}) + \tau^2}} \right) \tag{6}$$

where $\mathbf{x}_{opt} \equiv \arg\max_{\mathbf{x}_i \in \{\mathbf{x}_1, ..., \mathbf{x}_n\}} \{\hat{Y}(\mathbf{x}_i) - cs(\mathbf{x}_i)\}$ for a given $c \in \mathbb{R}$ and predicted values $\hat{Y}(\mathbf{x}_1), ..., \hat{Y}(\mathbf{x}_n)$ given by (4). Huang et al. [2006] state that the $\mathbf{x}_{opt}$ design point is chosen to reflect the user's degree of risk aversion, where $c = 1$ represents a "willingness to trade 1 unit of predicted objective value for 1 unit of the standard deviation of prediction uncertainty." Because it assumes $\tau^2 > 0$, AEI could propose going to an input that has already been observed. To prevent this from happening too often, the EI is multiplied by a penalty term involving $s^2(\cdot)$ and τ^2. If enough replicates are conducted at the same design point, say $\mathbf{x}_i$, then both $s^2(\mathbf{x}_i) \to 0$ and $AEI(\mathbf{x}_i) \to 0$.

To maximize AEI, we use its gradient vector $\dfrac{\partial EI(\mathbf{x})}{\partial \mathbf{x}}$. After several simplifying steps

reproduced in Section 7, the gradient of EI is

$$\begin{aligned}\frac{\partial EI(\mathbf{x})}{\partial \mathbf{x}} &= \Phi[Z(\mathbf{x})]\left\{Z(\mathbf{x})\frac{\partial s(\mathbf{x})}{\partial \mathbf{x}} + s(\mathbf{x})\frac{\partial Z(\mathbf{x})}{\partial \mathbf{x}}\right\} + \phi\left[Z(\mathbf{x})\right]\frac{\partial s(\mathbf{x})}{\partial \mathbf{x}} \\ &= \Phi\left[Z(\mathbf{x})\right]\frac{\partial}{\partial \mathbf{x}}\mu(\mathbf{x}) + \phi\left[Z(\mathbf{x})\right]\frac{\partial s(\mathbf{x})}{\partial \mathbf{x}}.\end{aligned} \tag{7}$$

The gradient of the prediction standard error, which depends on the correlation function, is

$$\frac{\partial s(\mathbf{x})}{\partial \mathbf{x}} = \frac{1}{2s(\mathbf{x})}\frac{\partial}{\partial \mathbf{x}}\left[-\mathbf{v}(\mathbf{x})^T {\mathbf{V}_\mathbf{X}}^{-1}\mathbf{v}(\mathbf{x})\right]. \tag{8}$$

Under our parameterization, the k^{th} component involves the scalar

$$\frac{\partial \mathbf{v}(\mathbf{x})^T {\mathbf{V}_\mathbf{X}}^{-1}\mathbf{v}(\mathbf{x})}{\partial \mathbf{x}_k} = -4\gamma_k \mathbf{v}(\mathbf{x})^T {\mathbf{V}_\mathbf{X}}^{-1}\left[(x_k - x_{1k})V(\mathbf{x}_1, \mathbf{x}), ..., (x_k - x_{nk})V(\mathbf{x}_n, \mathbf{x})\right]^T. \tag{9}$$

Finally, the gradient of AEI is

$$\frac{\partial AEI(\mathbf{x})}{\partial \mathbf{x}} = \frac{\tau\ s(\mathbf{x})\ EI(\mathbf{x})}{(s^2(\mathbf{x}) + \tau^2)^{3/2}}\frac{\partial s(\mathbf{x})}{\partial \mathbf{x}} + \left(1 - \frac{\tau}{\sqrt{s^2(\mathbf{x}) + \tau^2}}\right)\frac{\partial EI(\mathbf{x})}{\partial \mathbf{x}}. \tag{10}$$

3.1 Global Variable Selection

If $\gamma_k = 0$, we say that the k^{th} input variable is "globally inactive," since it does not affect the response anywhere in the design space. Following Linkletter et al. [2006] we specify a prior for the GP range parameters that places positive mass on each input variable being independent of the response; specifically, $\gamma_k = u_k b_k$, where $u_k \sim$ Gamma(a_u, b_u) is independent of $b_k \sim$ Bernoulli(θ), and $\theta \sim$ Beta(a_θ, b_θ) is the prior probability of any variable being globally active. The decision to declare an input variable globally active is based on the posterior probability $\bar{b}_k =$ Prob$(b_k = 1 \mid \mathbf{y}) =$ Prob$(\gamma_k > 0 \mid \mathbf{y})$. We drop variable k if $\hat{b}_k < \kappa$ for a pre-specified $\kappa \in (0, 1)$.

As described in the Appendix, we use MCMC to obtain M posterior draws from the joint distribution of the parameters, $\mathbf{\Theta} = \{\eta, r, \mu, b_1, ..., b_p, u_1, ..., u_p\}$, denoted $\mathbf{\Theta}_1, ..., \mathbf{\Theta}_M$. The samples are used to approximate the global variable importance measures $\bar{b}_k$ as well as the local variable important measures described in the next section.

4. Local Variable Selection

4.1 Defining Local Importance

Even after performing global variable selection, it may be possible to further reduce the dimension of the search space around a neighborhood of the maximum and improve convergence to the optimum. One approach to extend the global variable selection methodology in Section 3.1 and perform local variable selection is to specify a prior on the response surface, with the probability that the response surface is zero in certain subregions [e.g. Kang et al., 2016]. However, this is likely too computationally intensive for sequential optimization. Instead, we define a measure of local importance and develop an algorithm that uses local variable selection to find a global maximum.

As a toy example, consider the two-dimensional function

$$f(\mathbf{x}) = 5x_2 \mathbb{1}_{\{x_1 \geq 0.5\}} - \sqrt{x_1 + x_2}\mathbb{1}_{\{x_1 < 0.5\}}.$$

Both x_1 and x_2 are needed to describe the function globally, but once we are in the local region where $x_1 \geq 0.5$, we need only vary x_2 for optimization.

The motivation for our local importance measure is that if the k^{th} input is locally inactive, then setting $\gamma_k = 0$ will not affect the fitted response surface around the maximum, and vice versa. Denote the estimated optimal input as $\hat{\boldsymbol{\chi}} = \arg\max_{\mathbf{x}\in[0,1]^p} \hat{Y}(\mathbf{x})$ and its ϵ-neighborhood as $\mathcal{N} = \{\mathbf{x}; ||\mathbf{x} - \hat{\boldsymbol{\chi}}|| < \epsilon\}$. Then the local importance measure is the squared correlation

$$R_k^2 = \frac{\left[\int_{\mathcal{N}} \{\hat{Y}(\mathbf{x}) - \mu\}\{\hat{Y}^k(\mathbf{x}) - \mu\} d\mathbf{x}\right]^2}{\int_{\mathcal{N}} \{\hat{Y}(\mathbf{x}) - \mu\}^2 d\mathbf{x} \int_{\mathcal{N}} \{\hat{Y}^k(\mathbf{x}) - \mu\}^2 d\mathbf{x}} \tag{11}$$

where $\hat{Y}$ are the baseline predictions and $\hat{Y}^k$ are the alternative predictions made under the assumption that $\gamma_k = 0$. All the other estimated parameters are the same.

We can estimate (11) using m MCMC draws from the posterior $\boldsymbol{\Theta}_\ell$, by constructing vectors of baseline predictions and alternative predictions at q locations. For each $\ell \in \{1, ..., m\}$, we take $\mathbf{q}_\ell$ to be 900 points distributed as a truncated multivariate normal

$N(\hat{\boldsymbol{\chi}}_\ell, \sigma_\epsilon \mathbf{I})$ bounded by $[0,1]^p$ and evaluate

$$R^2_{k\ell} = \text{Corr}\left(\hat{Y}, \hat{Y}^{(k)} \mid \boldsymbol{\Theta}_\ell, \mathbf{X}_\ell, \mathbf{q}_\ell\right)^2 \tag{12}$$

using only the $n_\epsilon < N$ design points $\mathbf{X}_\ell$ that are within a radius ϵ of the local maximum $\hat{\boldsymbol{\chi}}_\ell$, in order to ensure that predictions are not influenced by design points far away. To prevent either too few or too many local design points from being considered, we can adjust ϵ such that $a \leq n_\epsilon \leq b$, for pre-specified $a, b \in \mathbb{Z}^+$.

An R^2 of 1 indicates perfect correlation; if that occurs, we see that setting the spatial range parameter to 0 makes no difference in predictions, suggesting that an input is not locally active. On the other extreme, an R^2 value of 0 suggests that the alternative predictions are very different from the original ones, offering evidence that an input is locally active.

Next, we average across all the m different posterior draws of the location of the maximum, and define the local importance L_k of input k as follows

$$L_k \equiv 1 - \frac{1}{m}\sum_{\ell=1}^{m} R^2_{k\ell}, \tag{13}$$

such that 0 indicates no importance and 1 indicates the greatest importance. We declare a variable to be locally active if $L_k \geq \rho$ for some pre-specified number, $0 < \rho < 1$.

Finally, denote $\mathbb{A}$ to be the set of locally active variables. It may appear troubling that the $L_1, ..., L_p$ measures of "local" importance are based on locations of maxima $\boldsymbol{\chi}_\ell$ that could be spread "globally" across the design space. This is intentional; if different local regions have different sets of locally active variables, we want to optimize over all of them to find a global maximum.

4.2 Selecting the Next Design Point

Since the expected improvement surface in p dimensions could be multi-modal and have large regions where the surface is flat, it is not trivial to find the input $\mathbf{x} \in [0,1]^p$ with the greatest expected improvement. To find a reasonable value, we first evaluate AEI over a matrix of candidate points $\mathbf{C}$.

Denote $\mathbf{C}$ to be the maximinLHS with N_2 candidate points in p dimensional space, with $N_2 >> n$. Let $\mathbf{T}$ be the "targeted" design that involves only the locally active variables: maximinLHS($N_2, |\mathbb{A}|$). Next, we replace the columns of $\mathbf{C}$ corresponding to the locally active variables with $\mathbf{T}$. By doing so, we hedge our bets. We construct a "better" space-filling design in the dimensions we think are locally active, which could help us identify a maximum faster. Not all is lost, however, if we misspecified the set of locally active variables, since we are still varying all of the globally active variables in the matrix of candidate points .

Then we choose the five candidate points with the greatest AEI and do local line searches in the direction of the p-dimensional AEI gradient, g.

We place two restrictions on the local line search. First, we require that the line search lie within a p-dimensional ball of radius ϵ centered at $\mathbf{x}^*$, the starting point.

Second, we require the line searches stay within the p-dimensional hypercube, $[0, 1]^p$. Whenever the line search proposes a coordinate at or outside of the boundary, that coordinate is set to equal the boundary value and the corresponding coordinate of the gradient is set to zero. In two dimensions, if the g_1 (horizontal) gradient hits the boundary, the subsequent design points of that line search would either slide up or down the boundary, depending if $g_2 > 0$ or $g_2 < 0$. This approach was inspired by the more rigorous approach of Rosen [1960].

Once the line searches have been done from each of the 5 candidate design points that had the largest AEI, we choose the one that has the maximum AEI. We repeat this process until 20 new design points have been augmented to the design matrix $\mathbf{X}$ and evaluated using the black-box function. We find that updating the parameter estimates each time a design point is added leads to the best performance in finding a maximum. We summarize this entire process in Algorithm 1.

4.3 Estimating the Optimum Location

To determine the estimated location of the optimum, we use information about the set of locally active variables, and we fix values for the locally inactive variables at the "best" input, as defined as the $\hat{\boldsymbol{\chi}}_\ell$ with the largest predicted value using the posterior median parameter

estimates

$$\hat{\boldsymbol{\chi}} = \arg \max_{\ell=1,\ldots,m} \hat{Y}(\hat{\boldsymbol{\chi}}_\ell \mid \bar{\boldsymbol{\Theta}}). \tag{14}$$

5. Simulation Study

5.1 Design

We conduct a simulation study to evaluate the effects of global and local variable selection on sequential optimization. Each simulated dataset begins with the same randomly generated maximin Latin Hypercube design [Joseph and Hung, 2008] with n_0 observations and p_0 inputs, denoted maximinLHS(n_0, p_0), and we compare five sequential algorithms for selecting the 20 additional design points:

1. **None** does not do any variable selection;
2. **GVS** conducts global variable selection only;
3. **LVS** does global and local variable selection and allows the set of locally active variables to change each time a new design point is added;
4. **Both** conducts global and local variable selection, but once a variable is declared 'locally inactive,' it will never be considered again; and
5. **Oracle** knows which variables are globally active and uses these only these variables throughout the sequential design.

For all methods we use uninformative priors $a_u = 1$ and $b_u = 20$ so that $E(u_k) = 20$ and Var$(u_k) = 400$, $\sigma_\mu = 100$, $a_\eta = b_\eta = 0.1$ and $a_\theta = b_\theta = 1$. We update the posterior of the parameters after each new design point is added to the dataset.

We compare results for several combinations of the initial sample size, n_0, and the number of input variables p_0. For each combination of n_0 and p_0 we simulate 100 datasets with response surface f set to the "Simba" function given in the Appendix and plotted in Figure 1. We add error terms ϵ to each observation, such that $\epsilon \sim N(0, \tau^2 = 0.05)$. In this function, the first six inputs are globally active, while only the first three are locally-active

Algorithm 1 Sequential Design using Global and Local Variable Selection

procedure SEQUENTIAL DESIGN
  Create an initial maximinLHS(n, p) design, $\mathbf{X}$
  Evaluate $y = f(\mathbf{X}) + \epsilon$
  Fit a Gaussian process and obtain $\mathbf{\Theta}_1, ..., \mathbf{\Theta}_M$ draws from the joint posterior distribution
  for step $i \in \{1, ..., 20\}$ **do**
    Perform global variable selection, GVS($\mathbf{\Theta}$)
    Construct a set of locally active variables $\mathbb{A}$ using LVS($\mathbf{\Theta}, i$)
    Given $\mathbb{A}$, construct a matrix of candidate points $\mathbf{C}$ using CANDIDATES($\mathbb{A}, \mathbf{\Theta}$)
    Choose the design point $\mathbf{x}^{**}$ that maximizes AEI using AEI($\mathbf{C}, \bar{\mathbf{\Theta}}$)
    Augment $\mathbf{x}^{**}$ to $\mathbf{X}$ and $Y(\mathbf{x}^{**})$ to Y
    Update parameter estimates $\mathbf{\Theta}_1, ..., \mathbf{\Theta}_M$
    Estimate the optimal input $\boldsymbol{\chi}$.

function GVS($\mathbf{\Theta}$)
  for variable $k \in \{1, ..., p\}$ **do**
    Keep X_k in the design matrix $\iff pr(\gamma_k > 0 \mid \mathbf{y}, \bar{\mathbf{\Theta}}) \geq \kappa$

function LVS($\mathbf{\Theta}$, i)
  Randomly sample $m < M$ posterior draws of the optimal design point $\hat{\boldsymbol{\chi}}_1, ..., \hat{\boldsymbol{\chi}}_m$
  for $\ell \in \{1, ..., m\}$ **do**
    Make predictions $\hat{Y} \mid \mathbf{\Theta}_\ell$ at q points using n_ϵ local design points within radius ϵ of $\hat{\boldsymbol{\chi}}_\ell$
    for variable $k \in \{1, .., p\}$ **do**
      Set $\gamma_k = 0$ and make alternative predictions, $\hat{Y}^{(k)}$ at the same q points
      Calculate the local importance, $L_{\ell k} \equiv 1 - \text{Corr}(\hat{Y}, \hat{Y}^{(k)} \mid \mathbf{\Theta}_\ell)^2$
  Summarize across posterior draws and calculate $L_k = \text{mean}(L_{1k}, ..., L_{mk})$ for $k \in \{1, ..., p\}$.
  Let $\mathbb{A}_i = \{k : L_k \geq \rho \mid \rho \in (0, 1)\}$ be the set of locally active variables at step i.
  If no variables meet this criteria, set $\rho = \max\{L_1, ..., L_p\}$ such that at least one variable is considered locally active. **return** $\mathbb{A}$

function CANDIDATES($\mathbb{A}$, $\mathbf{\Theta}$)
  Construct $\mathbf{C}$, a maximinLHS(N_2, p) design
  Fill the columns of $\mathbf{C}$ corresponding to $\mathbb{A}$ with a LHS$(N_2, |\mathbb{A}|)$ design **return** $\mathbf{C}$

function AEI($\mathbb{A}, \mathbf{C}, \bar{\mathbf{\Theta}}$)
  Evaluate AEI at each of the candidate points $\mathbf{C}$
  Set $\mathbf{x}^{(i)} = \arg\max_{\mathbf{x} \in \mathbf{C}} AEI(\mathbf{x})$
  for variable $k \in \{1, .., p\}$ **do**
    compute g_k, the k^{th} component of the gradient of AEI($\mathbf{x}^*$)
  Choose the 5 design points with the largest AEI
  For each of the 5 design points (WLOG: $\mathbf{x}^{(i)}$), do five line searches spanned by $\mathbf{x}^{(i)} + t\mathbf{g}$ for different step multipliers, $t \in \left[0, \epsilon(g_1^2 + ... + g_p^2)^{-1/2}\right]$, where the bounds keep the line searches within a radius ϵ of the starting point, and each subsequent line search beings where the previous one ends. **return** $\mathbf{x}^{**} \equiv \arg\max_t AEI(\mathbf{x}^{(i)} + t\mathbf{g})$

around the true optimum. The true maximum depends linearly on X_1 in a local region defined primarily by X_2 and X_3.

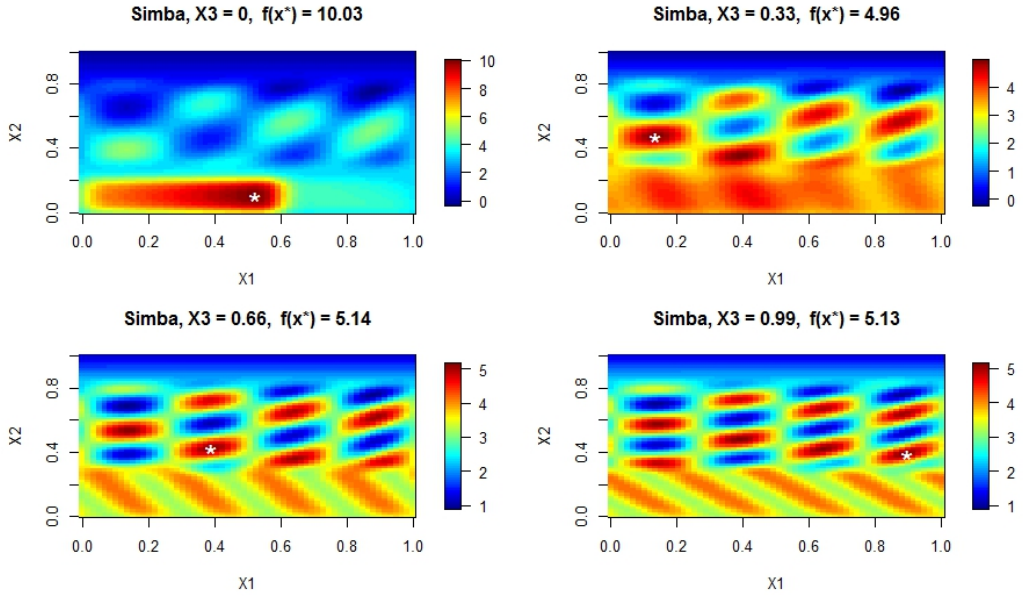

Figure 1: Holding $X_4 = X_5 = X_6$ fixed at 0.28, and varying $X_3 \in \{0, 0.33, 0.66, 0.99\}$, we see how the response surface changes. Additionally, the location and value of the local maxima vary considerably. The white star represents the location of the maximum.

Since all methods begin with the same n_0 observations, our performance metric is the improvement

$$f(\hat{\boldsymbol{\chi}}_i \mid \bar{\boldsymbol{\Theta}}_i) - f(\hat{\boldsymbol{\chi}}_0 \mid \bar{\boldsymbol{\Theta}}_0) \tag{15}$$

for sequential step $i \in \{1, ..., 20\}$, where the true function evaluated at the best input, based on parameter estimates from the initial LHS design, is denoted as $f(\hat{\boldsymbol{\chi}}_0 \mid \bar{\boldsymbol{\Theta}}_0)$. Algorithms with large improvement are preferred.

5.2 Results

We present the improvement over iteration, averaged across all 100 simulated datasets, for each setting and approach in Figure 2. As expected, the Oracle approach has the largest improvement in all cases. The methods with the smallest improvement is often ”None” or ”Both,” where the former has to optimize more dimensions than necessary, and the latter is often stuck in sub-optimal regions of the design space. LVS and GVS are often similar, but

LVS has larger improvement than GVS in three of the four simulation settings.

Improvement over iteration is choppy, partially because this is a difficult test function with noisy observations, and also because the approaches are re-estimating where the location of the optimum $\mathbf{x}_{opt}$ is each time a design point is added. While the average improvement (Figure 2) is around 2, improvement for any given simulated dataset and iteration could vary greatly, from -25 to 8. The large negative values often occur in the beginning of the sequential design, when parameter estimates are based on a limited number of observations in high dimensional space.

We see in Figure 3 that our measure of local importance, defined in Section 4.1, can accurately distinguish between variables that are locally active and locally inactive. The first three truly locally-active inputs generally have the largest L_k values, where $0 \leq L_k \leq 1$.

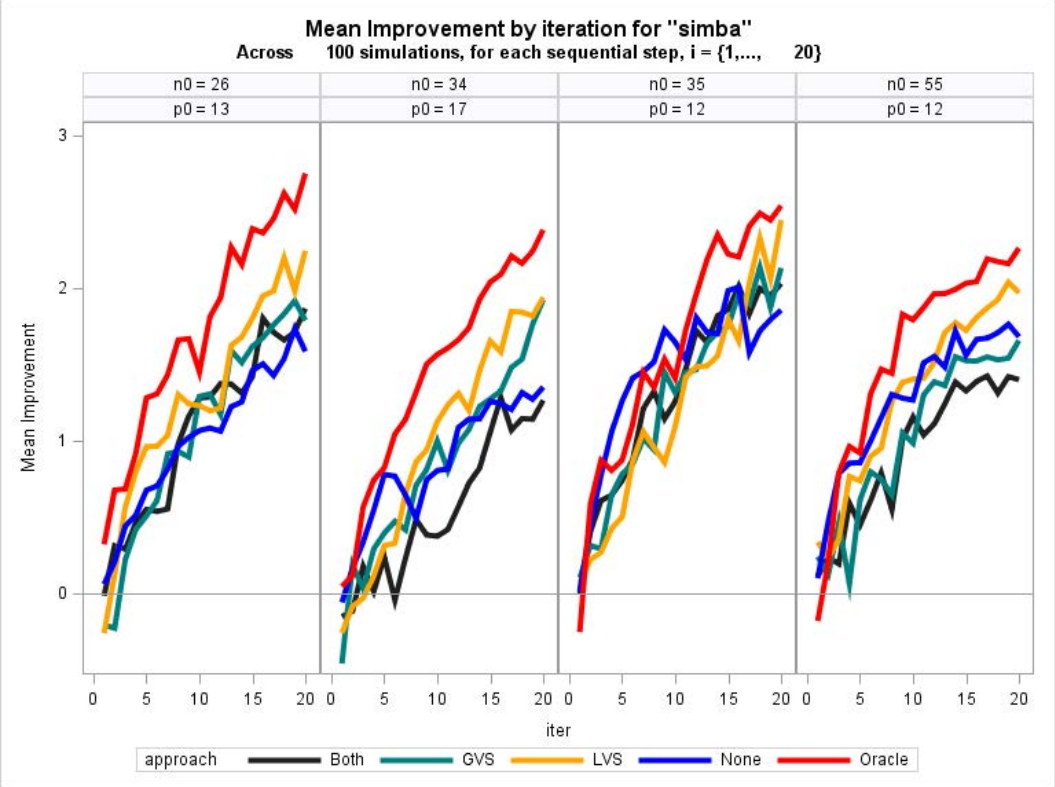

Figure 2: We evaluate the "Simba" test function across a range of initial design sizes n_0 and dimensions p_0, and calculate the change in estimated maximum as improvement$(i) = f(\hat{\boldsymbol{\chi}}_i \mid \boldsymbol{\Theta}_i) - f(\hat{\boldsymbol{\chi}}_0 \mid \boldsymbol{\Theta}_0)$. Oracle (in red) performs the best across all settings, since it is optimizing over only the six globally active variables.

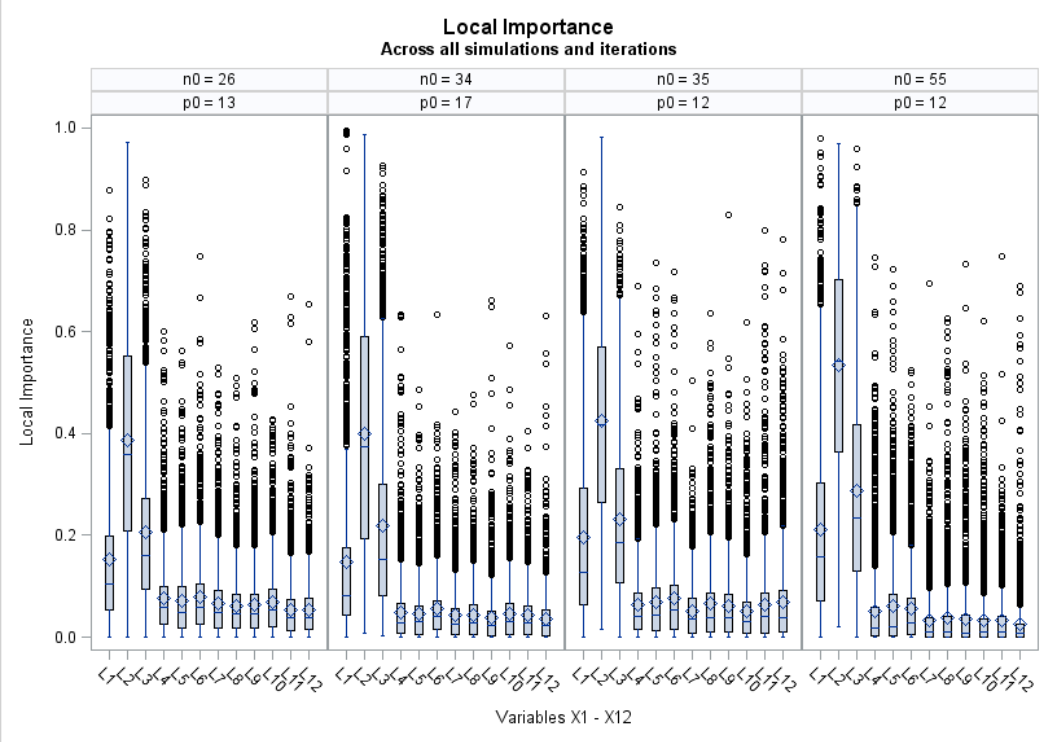

Figure 3: Variables X1, X2, X3 are truly locally active. Our measure of local importance captures this information across a variety of simulation settings, where values closer to 1 indicate greater local importance.

6. Discussion

We proposed a new method for using global and local variable selection for finding the maximum of an unknown function that takes in a large number of inputs. From our simulation study, we observed that the approach that uses only the important inputs performed far better than the approach that used all of the inputs. This leads us to conclude that combining Augmented Expected Improvement with global variable selection leads to better estimation of the optimal input. We also find that our measure of local variable importance effectively identifies the correct subset of locally-active variables. Finally, we see some evidence that using local variable information in creating a targeted search in a low-dimensional space can lead to faster and greater improvement than doing global variable selection alone.

Acknowledgements

This work was supported by National Science Foundation grants DGE-1633587 and DMR-1535082.

7. Appendix

7.1 MCMC details

We use Metropolis-Hastings within Gibbs sampling to obtain posterior samples of $\mathbf{\Theta}$. Using the parameterization in Section 2, let

$$V(\mathbf{x}, \mathbf{x}') = \frac{1}{\eta}\left[rK(\mathbf{x}, \mathbf{x}') + (1-r)1_{\{\mathbf{x}=\mathbf{x}'\}}\right] \equiv \frac{1}{\eta}W(\mathbf{x}, \mathbf{x}'). \tag{16}$$

Denote $\frac{1}{\eta}\mathbf{W}_{\mathbf{X}}$ as the n x n covariance matrix corresponding to $\mathbf{X}$. The log likelihood is

$$\log L\left(\mathbf{y} \mid \mathbf{\Theta}, \mathbf{X}\right) = -\frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln|\frac{1}{\eta}\mathbf{W}_{\mathbf{X}}| - \frac{\eta}{2}(\mathbf{y}-\mu\mathbf{1}_n)^T\mathbf{W}_X^{-1}(\mathbf{y}-\mu\mathbf{1}_n). \tag{17}$$

The full conditional distributions of μ, η, θ, and b_k are conjugate, and so these parameters are updated by sampling from their full conditional distributions

$$\begin{aligned}
\eta \mid \text{rest} &\sim \text{Gamma}\left(\frac{n}{2} + a_\eta \ , \ b_\eta + \frac{1}{2}\left[(\mathbf{y}-\mu\mathbf{1}_n)^T{\mathbf{W}_{\mathbf{X}}}^{-1}(\mathbf{y}-\mu\mathbf{1}_n)\right]\right) \\
\mu \mid \text{rest} &\sim \text{Normal}\left(\frac{\eta\mathbf{1}_n^T\mathbf{W}_X^{-1}\mathbf{y}}{\sigma_\mu^{-2}+\eta w} \ , \ \frac{1}{\sigma_\mu^{-2}+\eta w}\right) \\
\theta \mid \text{rest} &\sim \text{Beta}\left(\alpha_\theta + \sum_{k=1}^{p} b_k \ , \ b_\theta + p - \sum_{k=1}^{p} b_k\right) \\
b_k \mid \text{rest} &\sim \text{Bernoulli}\left(\frac{p_{k1}}{p_{k1}+p_{k0}}\right)
\end{aligned} \tag{18}$$

where $w = \mathbf{1}_n^T\mathbf{W}_X^{-1}\mathbf{1}_n$ and $p_{k\ell} \equiv p(\mathbf{y} \mid b_k = \ell, \mathbf{\Theta}_{(-k)})p(b_k = \ell \mid \theta)$ for $\mathbf{\Theta}_{(-k)} = \mathbf{\Theta}_k/\{b_k\}$.

We implement the Metropolis-Hastings algorithm [Hastings, 1970] to update r and $u_1, ..., u_p$. The variance ratio r is sampled using the Metropolis-Hastings algorithm with a Beta$(10, 1)$ proposal distribution. For each $k \in \{1, ..., p\}$, u_k is updated from its prior if

$b_k = 0$ and if $b_k = 1$ it is updated using a Metropolis-Hastings step with a uniform candidate distribution (conditioned on the current value of u_k) Uniform $\left(\max\{0, u_k - \epsilon(u_k)\}\ ,\ u_k + \epsilon(u_k)\right)$, where

$$\epsilon(u_k) \equiv \begin{cases} \max\{100, hu_k\} & u_k \geq 0.30 \\ 0.95 & 0 \leq u_k < 0.30 \end{cases} \tag{19}$$

and $h \sim$ Unif$(1/2, 2)$. This proposal distribution is used because ensures candidates are positive and its candidate distribution's variance increase with the current value of u_k.

7.2 Simba test function

Below is the functional form of "Simba," written in R code.

```
# upper bumps, involving all variables
Y = 3.14749 + sin(  2*pi*(x1^2 - 2*x2*(1 + x3) ) )*
  ( pnorm( 30 *(x2 -.3))  + pnorm(30*(.8 - x2)) - 1 )*
  2*sin( 4*pi*x1 + 3*pi*(1+x3) + 2*pi*(x4 + x5) + 3*pi*(1 + x6))  +

  ### Simba lion king when x3 near 0.2
  (  4 + 6*(x1) )*
  ((  pnorm( 30*(x2 - .0)) + pnorm(30*(.2 - x2))  ) - 1  )*
  ((  pnorm( 30*(x1 - .0)) + pnorm(30*(.6 - x1))  ) - 1  )*
  (   pnorm( 10*(.2 - x3))  ) +

  ### other area below simba lion king
  ### Simba lion king when x3 near 0.2
  (  1 - 8*(x1 + x2 - x4 - x5 - x6)^2 )*
  ((  pnorm( 40*(x2 - .0)) + pnorm(40*(.2 - x2))  ) - 1  )*
  ((  pnorm( 40*(x1 - .6)) + pnorm(40*(1 - x1))  ) - 1  )*
  (   pnorm( 10*(.2 - x3))  ) +
```

```
####  locally active region when x3 > .2
.5*(  1 - sin( 8*pi*x1 + 7*pi*x2*x3 - 4*pi*x4*x5*x6) )*
((  pnorm( 30*(x2 - .0)) + pnorm(30*(.3 - x2))  ) - 1  )*
(   pnorm( 8*(x3 - .3))  ) +

#### deceptive maximum, when x2 > .8, x3 > .2, involving
#### all variables
(  5*cos( 2*(x2 + .5)*(-x4 + .5)*(-x5+.5)^2 )*(-x6 - .5) -
     .02*((1-x2)^2 +
            (1-x1)^2 +
            (1-x3  - .3*x4)^2  +
            (1-x5  + .5*x4)^2 +
            (.8-x6 - .4*x4)^2 )   )*
( pnorm(5*(x2 - 1) + pnorm(10*(.5 - x3))))
```

7.3 Gradient of Expected Improvement

We show the derivation for the gradient of EI, using the concise notation $\dot{s} \equiv \frac{\partial s(\mathbf{x})}{\partial \mathbf{x}}$ and $\dot{z} \equiv \frac{\partial Z(\mathbf{x})}{\partial \mathbf{x}}$.

$$\begin{aligned} EI(\mathbf{x}) &= s\left\{z\Phi(z) + \phi(z)\right\} \\ \frac{\partial EI(\mathbf{x})}{\partial \mathbf{x}} &= s\left\{z\dot{\Phi}(z) + \Phi(z)\dot{z} + \dot{\phi}(z)\right\} + \dot{s}\left\{z\Phi(z) + \phi(z)\right\} \\ &= s\left\{z\dot{\phi}(z)\dot{z} + \Phi(z)\dot{z} - \phi(z)z\dot{z}\right\} + \dot{s}\left\{z\Phi(z) + \phi(z)\right\} \\ &= \Phi(z)\left\{s\dot{z} + \dot{s}z\right\} + \dot{s}\left\{\phi(z)\right\}. \end{aligned} \tag{20}$$

References

Er-Wei Bai, Kang Li, Wen-Xiao Zhao, and Weiyu Xu. Kernel based approaches to local nonlinear non-parametric variable selection. *Automatica*, 50(1):100–113, 2014. doi: 10.1016/j.automatica.2013.10.010. URL http://dx.doi.org/10.1016/j.automatica.2013.10.010.

Eric Brochu, Vlad M. Cora, and Nando de Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. 2010.

Robert B. Gramacy and Herbert K. H. Lee. Adaptive design and analysis of supercomputer experiments. *Technometrics*, 51(2):130–145, 2009. ISSN 00401706. URL http://www.jstor.org/stable/40586591.

Wilfred Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–109, 1970.

D. Huang, T. T. Allen, W. I. Notz, and N. Zeng. Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of Global Optimization*, 34(3):441–466, 2006. ISSN 1573-2916. doi: 10.1007/s10898-005-2454-3. URL http://dx.doi.org/10.1007/s10898-005-2454-3.

Donald R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21:345–383, 2001.

Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

V Roshan Joseph and Ying Hung. Orthogonal-maximin latin hypercube designs. *Statistica Sinica*, pages 171–186, 2008.

Jian Kang, Brian J Reich, and Ana-Maria Staicu. Scalar-on-image regression via the soft-thresholded gaussian process. *arXiv preprint arXiv:1604.03192*, 2016.

Crystal Linkletter, Derek Bingham, Nicolas W. Hengartner, David Higdon, and Kenny Q. Ye. Variable selection for gaussian process models in computer experiments. *Technometrics*, 48(4):478–490, 2006. doi: 10.1198/004017006000000228. URL http://dx.doi.org/10.1198/004017006000000228.

J. Močkus. On bayesian methods for seeking the extremum. In G. I. Marchuk, editor, *Optimization Techniques IFIP Technical Conference Novosibirsk July 1–7, 1974*, pages

400–404. Springer Berlin Heidelberg, Berlin, Heidelberg, 1975. ISBN 978-3-540-37497-8.

J. B. Rosen. The gradient projection method for nonlinear programming. part I. linear constraints. 8(1):181–217, March 1960. ISSN 0368-4245 (print), 1095-712X (electronic).

Jerome Sacks, Susannah B. Schiller, and William J. Welch. Designs for computer experiments. *TECHNOMETRICS*, 31(1):41–47, February 1989.

Songqing Shan and G. Gary Wang. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and Multidisciplinary Optimization*, 41(2):219–241, 2010. ISSN 1615-1488. doi: 10.1007/s00158-009-0420-2. URL http://dx.doi.org/10.1007/s00158-009-0420-2.

William J. Welch, Robert. J. Buck, Jerome Sacks, Henry P. Wynn, Toby J. Mitchell, and Max D. Morris. Screening, predicting, and computer experiments. *Technometrics*, 34(1):15–25, 1992. ISSN 00401706. URL http://www.jstor.org/stable/1269548.