

Adaptively Iterating Variance-Based Sensitivity Analysis

Brian W. Bush
National Renewable Energy Laboratory,

Joanne Wendelberger
Los Alamos National Laboratory,

Rebecca Hanes
National Renewable Energy Laboratory

December 21, 2020

Abstract

Inspired by the well-established variance-based methods for global sensitivity analysis, we develop a local total sensitivity index that decomposes the global total sensitivity conditions by independent variables. We employ this local sensitivity index in a new method of experimental design that sequentially and adaptively samples the domain of a multivariate function according to local contributions to the global variance. The method is demonstrated on a nonlinear toy model that has a three-dimensional domain and a three-dimensional codomain, but also on a complex, high-dimensional simulation of the production of bioproducts from biomass.

Keywords:

100 or fewer words.

3 to 6 key-words, that do not appear in the title

Symbol	Definition
$i \in \{1 \dots m\}$	independent-variable index and dimension
$j \in \{1 \dots n\}$	dependent-variable index and dimension
$k \in \{1 \dots N\}$	observation index and dimension
\mathbb{E}_Z	expectation over random variable(s) $Z \in \{X_i, \mathbf{X}_{\sim i}\}$
\mathbb{V}_Z	variance over random variable(s) $Z \in \{X_i, \mathbf{X}_{\sim i}\}$
$X_i \in [0, 1]$	independent variable for dimension i
$\mathbf{X}_{\sim i}$	independent variables except for dimension i
Y_j	dependent variable for dimension j
\mathbf{A}, \mathbf{B}	$N \times k$ design matrices
$\mathbf{A}_{\mathbf{B}_i}$	design matrix \mathbf{A} with its i th column replaced by the i th column of \mathbf{B}
$x_{\mathbf{Z}_i}^{(k)} \in [0, 1]$	i th independent variable for the k th observation in the design matrix $\mathbf{Z} \in \{\mathbf{A}, \mathbf{B}\}$
$y_{\mathbf{Z},j}^{(k)}$	j th dependent variable for the k th observation using the design matrix $\mathbf{Z} \in \{\mathbf{A}, \mathbf{B}, \mathbf{A}_{\mathbf{B}_i}\}$
$\mathbb{1}_c$	indicator function: 1 if the condition c holds, but 0 otherwise
$u \hat{=} v$	v is an estimate for u

Table 1: Notation used in this paper, inspired by and generally consistent with [Sal+10].

1 Introduction

2 Background

3 Method

We consider multivariate functions $\mathbf{Y}(\mathbf{X})$ with m independent variables X_i and n dependent variables Y_j . (Table 1 summarizes the notation used in this paper.) These functions may represent the input-output relationship of computer models (either deterministic or stochastic) or physical experiments.

We approach the problem of developing local versions of variable-based global sensitivity indices by expressing the standard global indices [Sal+10] in forms amenable to local decomposition. The local total sensitivity separately partitions the global sensitivity along each independent variable. We then adopt experimental designs that are suitable for iterative extension. The iterative experimental design samples the space of independent variables proportionally to the local variance indices observed so far in the experiment.

3.1 Global Sensitivity

We start from standard definitions [Sal+10] that use conditional expectations and variances to express the first-order and total sensitivity indices, respectively:

$$S_{i,j} = \frac{\mathbb{V}_{X_i} [\mathbb{E}_{\mathbf{X}_{\sim i}} [Y_j | X_i]]}{\mathbb{V}_{\mathbf{X}} [Y_j]} \quad (1)$$

$$T_{i,j} = \frac{\mathbb{E}_{\mathbf{X}_{\sim i}} [\mathbb{V}_{X_i} [Y_j | \mathbf{X}_{\sim i}]]}{\mathbb{V}_{\mathbf{X}} [Y_j]} \quad (2)$$

The first-order index $S_{i,j}$ quantifies the influence of X_i solely upon the variance of Y_j whereas the total index $T_{i,j}$ quantifies the influence of X_i in combination with other independent variables upon the variance of Y_j . The two indices arise from decomposing the variance Y_j and obey a variety of summation relationships and bounds—see [cite] for detailed discussion.

Practically estimating Eqs. (1) and (2) may be problematic because of computational complexity, finite sample size, and numerical issues. Careful design of experiments alleviates some of these difficulties. An efficient design employs two $N \times k$ design matrices, \mathbf{A} and \mathbf{B} , that are hybridized into matrices $\mathbf{A}_{\mathbf{B}_i}$ that are constructed by replacing the i th column of \mathbf{A} with the i th column of \mathbf{B} . The number of rows N determines the size of the experiment, which requires evaluating $\mathbf{Y}(\mathbf{X})$ for each row of \mathbf{A} , \mathbf{B} , and $\mathbf{A}_{\mathbf{B}_i}$: thus $N(k+2)$ observations must be made. The column-swapping construction facilitates efficient evaluation of the conditional expectations and variances by controlling which independent variables are varied and by organizing the computation.

The best-practice recommendations of Saltelli [Sal+10] are to use the formulae in Jansen [Jan99] when computing the variances of conditional expectations and the expectations of conditional variances:

$$\mathbb{V}_{\mathbf{X}} [Y_j] - \mathbb{V}_{X_i} [\mathbb{E}_{\mathbf{X}_{\sim i}} [Y_j | X_i]] \hat{=} \frac{1}{2N} \sum_{k=1}^N \left| y_{\mathbf{B},j}^{(k)} - y_{\mathbf{A}_{\mathbf{B}_i},j}^{(k)} \right|^2 \quad (3)$$

$$\mathbb{E}_{\mathbf{X}_{\sim i}} [\mathbb{V}_{X_i} [Y_j | \mathbf{X}_{\sim i}]] \hat{=} \frac{1}{2N} \sum_{k=1}^N \left| y_{\mathbf{A},j}^{(k)} - y_{\mathbf{A}_{\mathbf{B}_i},j}^{(k)} \right|^2 \quad (4)$$

Here the $y_{\mathbf{A},j}^{(k)}$, $y_{\mathbf{B},j}^{(k)}$, and $y_{\mathbf{A}_{\mathbf{B}_i},j}^{(k)}$ are the observation of Y_j for the k th row of the design matrix \mathbf{A} , \mathbf{B} , and $\mathbf{A}_{\mathbf{B}_i}$, respectively. We supplement these formulae with the analogous formula for computing the overall variance:

$$\mathbb{V}_{\mathbf{X}} [Y_j] \hat{=} \frac{1}{2N} \sum_{k=1}^N \left| y_{\mathbf{A},j}^{(k)} - y_{\mathbf{B},j}^{(k)} \right|^2 \quad (5)$$

Combining Eqs. (3) through (5), we adopt the following estimators for first-order and

Cite
Saltelli
books
and
review
pa-
pers.

total global sensitivities and variance:

$$\hat{S}_{i,j} = 1 - \frac{\sum_{k=1}^N \left| y_{\mathbf{B},j}^{(k)} - y_{\mathbf{A}_{\mathbf{B}_i},j}^{(k)} \right|^2}{\sum_{k=1}^N \left| y_{\mathbf{A},j}^{(k)} - y_{\mathbf{B},j}^{(k)} \right|^2} \quad (6)$$

$$\hat{T}_{i,j} = \frac{\sum_{k=1}^N \left| y_{\mathbf{A},j}^{(k)} - y_{\mathbf{A}_{\mathbf{B}_i},j}^{(k)} \right|^2}{\sum_{k=1}^N \left| y_{\mathbf{A},j}^{(k)} - y_{\mathbf{B},j}^{(k)} \right|^2} \quad (7)$$

$$\hat{V}_j = \frac{1}{2N} \sum_{k=1}^N \left| y_{\mathbf{A},j}^{(k)} - y_{\mathbf{B},j}^{(k)} \right|^2 \quad (8)$$

3.2 Experimental Design

Although any of a number of design methods could be used for creating the \mathbf{A} and \mathbf{B} matrices, quasi-random sequences are well suited for computing the sensitivity indices and Ref. [Sal+10] recommends using Sobol' sequences because of their low discrepancy properties. In this procedure, the \mathbf{A} and \mathbf{B} matrices are placed side-by-side to form an $N \times 2k$ matrix where the N rows are consecutive points in the $2k$ -dimensional Sobol' sequence. An experiment can very simply be enlarged just by appending additional rows of the Sobol' sequence, so this design procedure is well suited for iteratively extending a sensitivity analysis.

Without loss of generality, we require that the domain of each independent variable be $[0, 1]$. This is convenient because the range of the Sobol' sequence is also $[0, 1]$ in each dimension.

3.3 Local Sensitivity

In developing a local version of variance-based sensitivity, one would like to attribute the variance in output to a specific value or interval of the independent variables. Equation (7), for example, involves evaluating Y_j at the points $x_{\mathbf{A}_i}^{(k)}$ and $x_{\mathbf{B}_i}^{(k)}$ to compute the contribution $\left| y_{\mathbf{A},j}^{(k)} - y_{\mathbf{A}_{\mathbf{B}_i},j}^{(k)} \right|^2$ to the global total sensitivity. For lack of a more specific way to assign this contribution to a particular value of the independent variable, we can simply spread this contribution uniformly between $x_{\mathbf{A}_i}^{(k)}$ and $x_{\mathbf{B}_i}^{(k)}$. Thus we define $t_{i,j}^{(\alpha,\epsilon)}(x_i)$ to be the local sensitivity of the output y_j upon the input x_i :

$$t_{i,j}^{(\alpha,\epsilon)}(x) = \sum_{k=1}^N \left| y_{\mathbf{A},j}^{(k)} - y_{\mathbf{A}_{\mathbf{B}_i},j}^{(k)} \right|^\alpha \cdot \frac{\mathbb{I}_{\min\{x_{\mathbf{A}_i}^{(k)}, x_{\mathbf{B}_i}^{(k)}\} - \frac{\epsilon}{2} \leq x \leq \max\{x_{\mathbf{A}_i}^{(k)}, x_{\mathbf{B}_i}^{(k)}\} + \frac{\epsilon}{2}}{\left| x_{\mathbf{A}_i}^{(k)} - x_{\mathbf{B}_i}^{(k)} + \epsilon \right|} \quad (9)$$

Here we have introduced $\epsilon > 0$ to guard against division by zero, which could occur if the two x_i coincide, and we have replaced the exponent 2 by the general parameter α . The parameter α amplifies the difference between the dependent variables, with $\alpha = 2$

corresponding to computing the variance. The parameter ϵ slightly widens the interval over which the contribution to the variance is spread. It is also useful to define a cumulative version of the local sensitivity:

$$T_{i,j}^{(\alpha,\epsilon)}(x) = \frac{1}{2N\hat{V}_j} \int_{-\infty}^x dx' t_{i,j}^{(\alpha,\epsilon)}(x') \quad (10)$$

Note that we can recover the global total sensitivity index, Equation (7), from the limit of Equation (10):

$$\hat{T}_{i,j} = \lim_{x \rightarrow \infty} T_{i,j}^{(2,\epsilon)}(x)$$

In order to sample an independent variable proportionally to $t_{i,j}^{(\alpha,\epsilon)}(x_i)$, we need a normalized version of it, which we call the sensitivity density:

$$\tau_{i,j}^{(\alpha,\epsilon)}(x) = \frac{t_{i,j}^{(\alpha,\epsilon)}(x)}{t_{i,j}^{(0,\epsilon)}(x)} \bigg/ \int_{-\infty}^{\infty} dx' \frac{t_{i,j}^{(\alpha,\epsilon)}(x')}{t_{i,j}^{(0,\epsilon)}(x')} \quad (11)$$

The factor $t_{i,j}^{(0,\epsilon)}(x)$ in the denominators removes the non-uniformity in the sampling of x_i . Finally, we can further summarize the local sensitivity by averaging it over the n dependent variables; we call this the average sensitivity density:

$$\bar{\tau}_i^{(\alpha,\epsilon)}(x) = \frac{1}{n} \sum_{j=1}^n \tau_{i,j}^{(\alpha,\epsilon)}(x) \quad (12)$$

It also has a cumulative version:

$$\bar{T}_i^{(\alpha,\epsilon)}(x) = \int_{-\infty}^x dx' \bar{\tau}_i^{(\alpha,\epsilon)}(x') \quad (13)$$

Unfortunately, the expression of the first-order sensitivity in Eq. (6) as the difference of two terms precludes its analogous decomposition into a local first-order sensitivity index. One could create a local first-order *insensitivity* index based on $1 - \hat{S}_{i,j}$, but the density of insensitivity would not be practical for iteratively adapting experimental designs.

3.4 Adaptively Iterating Sensitivity Analysis

We are now in a position to specify an algorithm for iteratively adapting the experimental design for a sensitivity analysis as that analysis proceeds. Let M be the batch size for the iterations.

1. Evaluate $y_{\mathbf{A},j}^{(k)}$, $y_{\mathbf{B},j}^{(k)}$, and $y_{\mathbf{A}_{\mathbf{B}_i},j}^{(k)}$ for the first M rows of the design matrices \mathbf{A} , \mathbf{B} , and $\mathbf{A}_{\mathbf{B}_i}$.
2. Compute the cumulative average sensitivity densities $\bar{T}_i^{(\alpha,\epsilon)}(x)$ for each independent variable.

3. Append M rows to the design matrices by taking the next M points in the Sobol' sequence, but transform those points according to the average sensitivity density. Namely, the Sobol' point s_ℓ is mapped to $\bar{T}_i^{(\alpha, \epsilon)^{-1}}(s_\ell)$, where $i = 1 + (\ell - 1 \bmod m)$, since $\ell \in \{1 \dots 2m\}$ because the \mathbf{A} and \mathbf{B} matrices are placed side-by-side in the Sobol' space.
4. Evaluate $y_{\mathbf{A},j}^{(k)}$, $y_{\mathbf{B},j}^{(k)}$, and $y_{\mathbf{AB}_i,j}^{(k)}$ for these M new rows of the design matrix.
5. Proceed to step #2 above.

For computational efficiency, the \hat{V}_j and $T_{i,j}^{(\alpha, \epsilon)}(x)$ can be updated during each iteration instead of being computed from scratch. The batch size can be $M = 1$ if an online algorithm is desired. The parameter ϵ should be set to a small positive value. The parameter α allows one to make the sampling completely non-adaptive ($\alpha = 0$) or highly adaptive (large positive α): setting α too high may result in missing sensitivities in the input space if the batch size is too small, since the algorithm may home in on the first observed region of sensitivity, thus missing other areas of high sensitivity.

3.5 Functional PCA

4 Results

We evaluate the adaptive iterative approach to sensitivity analysis first using a toy model with a manageably small number of dimensions, and then with a complex simulation.

4.1 Toy Model

The Appendix A describes a class of toy models that have strong nonlinearities, configurable dimensionality, and specifiable discontinuities in the dependent variables. For convenience in exploring and plotting results, we select three input dimensions ($m = 3$) and three output dimensions ($n = 3$). The model has a zeroth-order discontinuity in the output at $x_1 \doteq 0.59$, a first-order discontinuity in the output at $x_2 \doteq 0.95$, and a second-order discontinuity in the output at $x_3 \doteq 0.10$.

Table 2 shows that the global sensitivities when $N = M = 1000$ exhibit a diversity of sensitive and insensitive first-order and total sensitivity indices. The cumulative sensitivities $T_{i,j}^{(2, \epsilon)}(x)$ in Figure 1 approach the global sensitivities $\hat{T}_{i,j}$ of Table 2 as $x_i \rightarrow 1$. One can also see mild nonlinearities near the $x_1 \doteq 0.59$, and $x_3 \doteq 0.10$ discontinuities of the model. The local sensitivity density in Figure 2 highlights the stronger contributions to the local variance near the model's zeroth- and first-order discontinuities ($x_1 \doteq 0.59$ and $x_2 \doteq 0.95$). Averaging over the output dimensions yields the average local sensitivity density in Figure 3. Figure 4 shows how densely different regions are sampled in the first five batches of $M = 10$ observations when the sensitivity analysis proceeds adaptively: this illustrates how quickly the algorithm detects the areas of higher variance as it transitions from the

Hybridize
the
method
with
func-
tional
PCA.

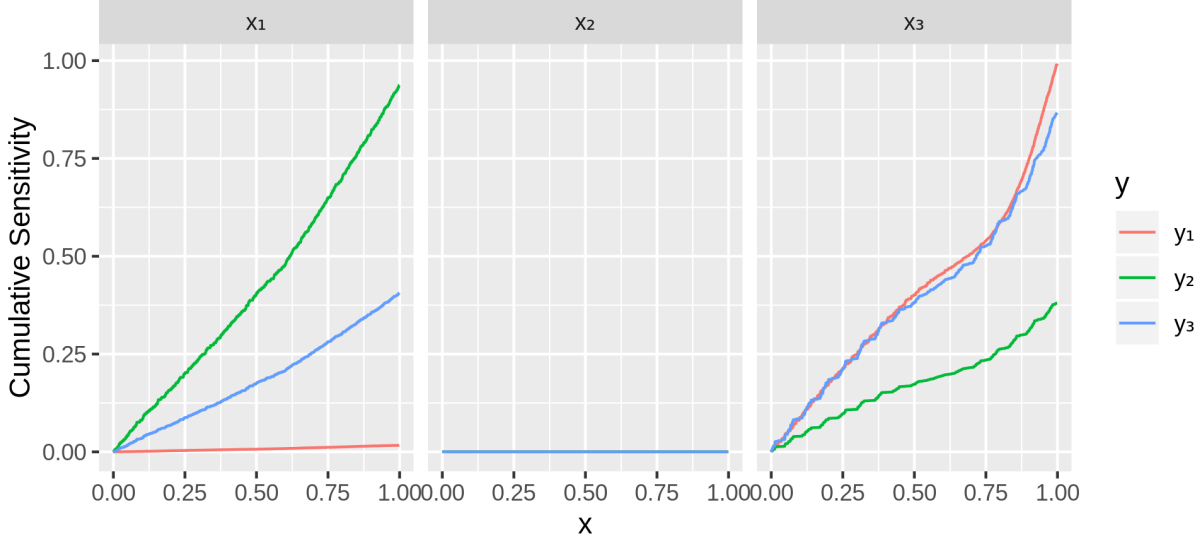


Figure 1: Cumulative local sensitivity $T_{i,j}^{(2,\epsilon)}(x)$ for the toy model, with $\epsilon = 10^{-4}$ and $N = M = 1000$.

$\hat{S}_{i,j}$	x_1	x_2	x_3	$\hat{T}_{i,j}$	x_1	x_2	x_3
y_1	0.01	0.00	0.98	y_1	0.01	0.00	0.99
y_2	0.62	0.00	0.06	y_2	0.94	0.00	0.38
y_3	0.13	0.00	0.59	y_3	0.41	0.00	0.87

Table 2: Global first-order sensitivity (left) and total sensitivities (right) results for the toy model, on a scale from zero (insensitive) to one (sensitive), with one large batch of observations $N = M = 1000$.

initial uniform sampling to the sensitivity-emphasizing sampling. Furthermore, Figure 5 displays two-dimensional projections of actual sampling for a batch size of $M = 10$, with 100 batches so $N = 1000$: one can see that the algorithm explores the whole space while concentrating on areas of high sensitivity.

4.2 Bioproduct Transition Dynamics Model

Figure 6 Figure 7 Figure 8

5 Conclusion

Acknowledgements

This work was authored in part by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under

PCA could be applied on the input space, too, or tensor PCA (maybe functional PCA)

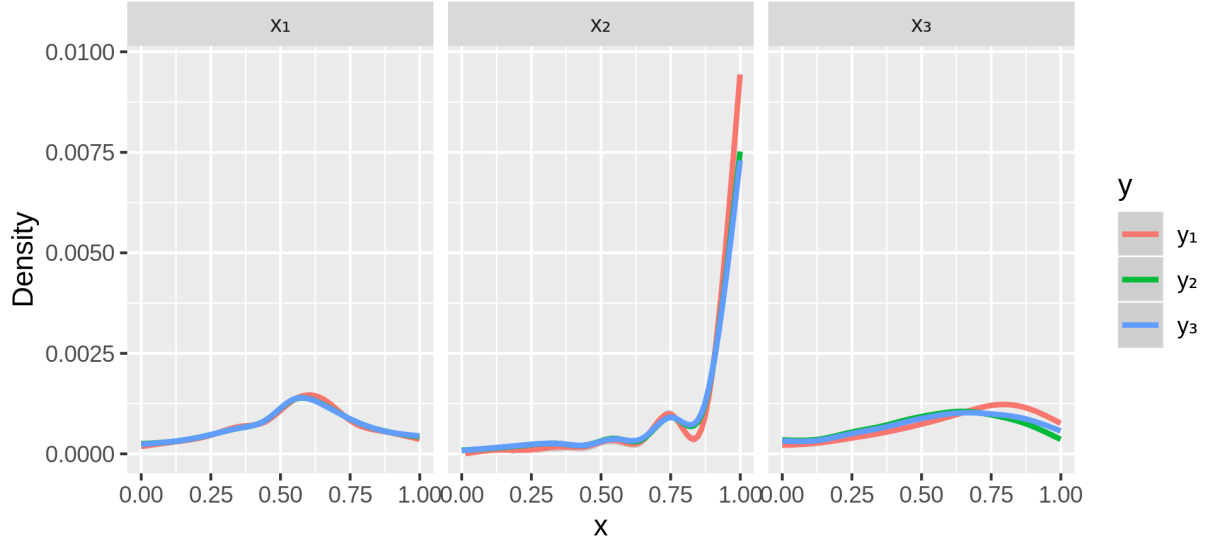


Figure 2: Sensitivity density $\tau_{i,j}^{(2,\epsilon)}(x)$ for the toy model, with $\epsilon = 10^{-4}$ and $N = M = 1000$.

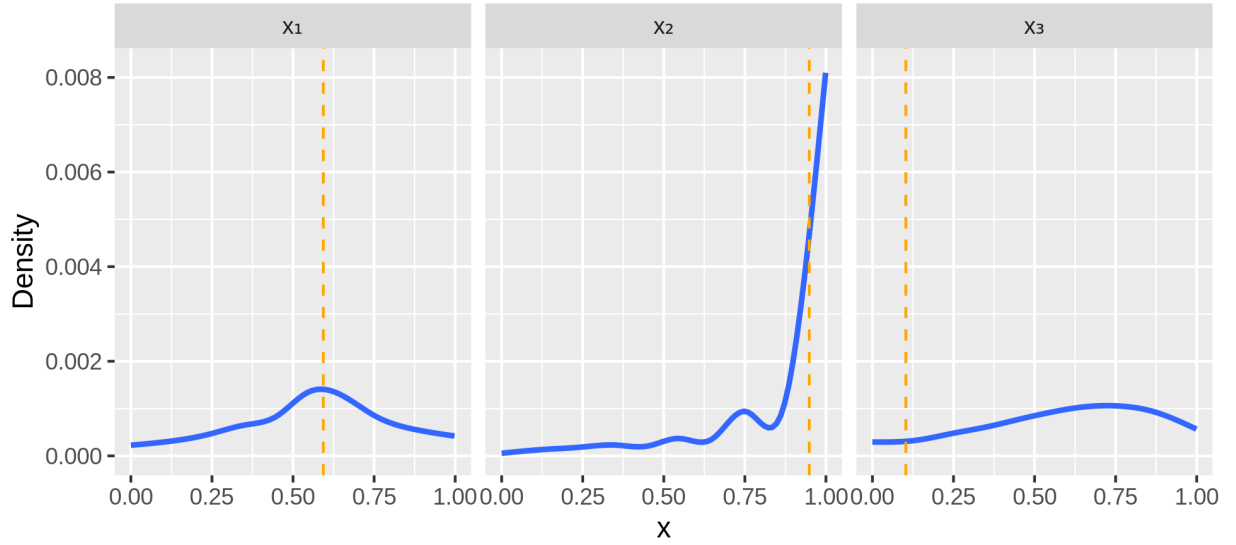


Figure 3: Averaged sensitivity density $\bar{\tau}_i^{(2,\epsilon)}(x)$ for the toy model, with $\epsilon = 10^{-4}$ and $N = M = 1000$. The three dashed orange lines mark locations of the toy model's discontinuities with respect to its three independent variables: the algorithm successfully detects the zeroth- and first-order discontinuities in x_1 and x_2 , respectively, but not the second-order discontinuity in x_3 .

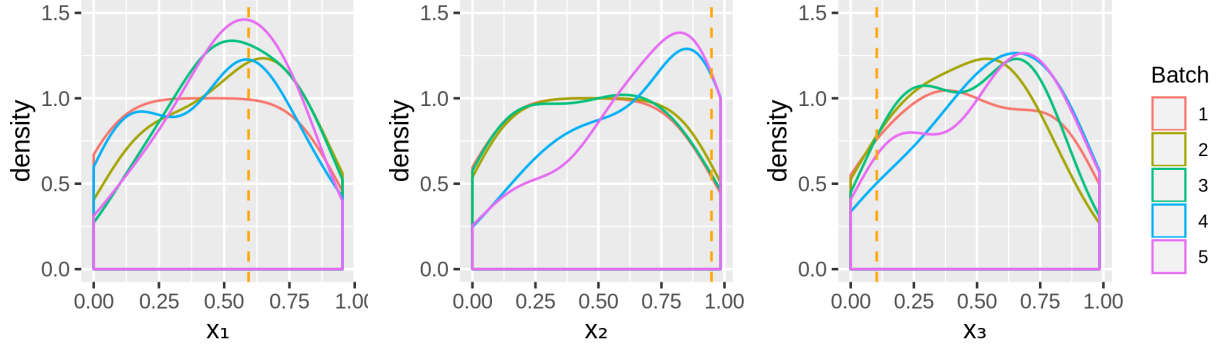


Figure 4: Density of sampling in the first five batches of $M = 10$ observations in adaptive sampling according to $\bar{\tau}_i^{(2,\epsilon)}(x)$ for the toy model, with $\epsilon = 10^{-4}$, starting with a first batch that uniformly samples. The three dashed orange lines mark locations of the toy model's discontinuities with respect to its three independent variables: the algorithm successfully detects the zeroth- and first-order discontinuities in x_1 and x_2 , respectively, but not the second-order discontinuity in x_3 .

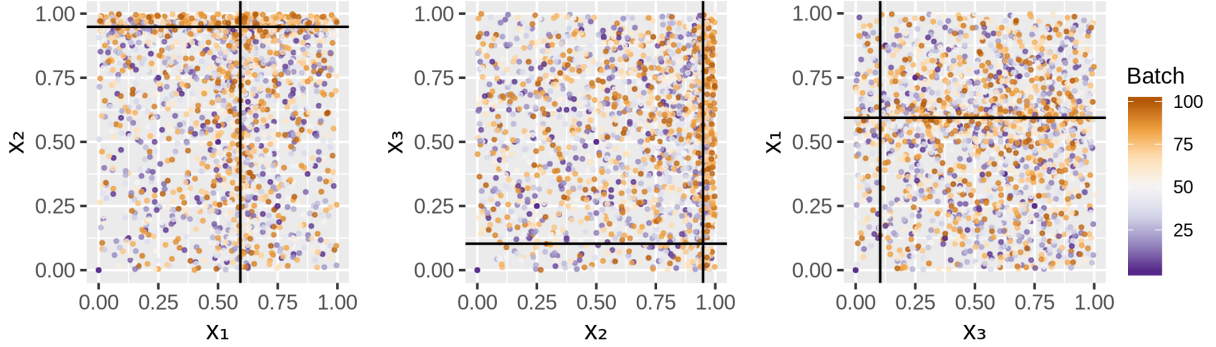


Figure 5: Two-dimensional projections of adaptive sampling according to $\bar{\tau}_i^{(2,\epsilon)}(x)$ for the toy model, with $\epsilon = 10^{-4}$ and for 100 sequential batches of $M = 10$ observations, so $N = 1000$. The solid black lines mark locations of the toy model's discontinuities with respect to its three independent variables: the algorithm successfully detects the zeroth- and first-order discontinuities in x_1 and x_2 , respectively, but not the second-order discontinuity in x_3 .

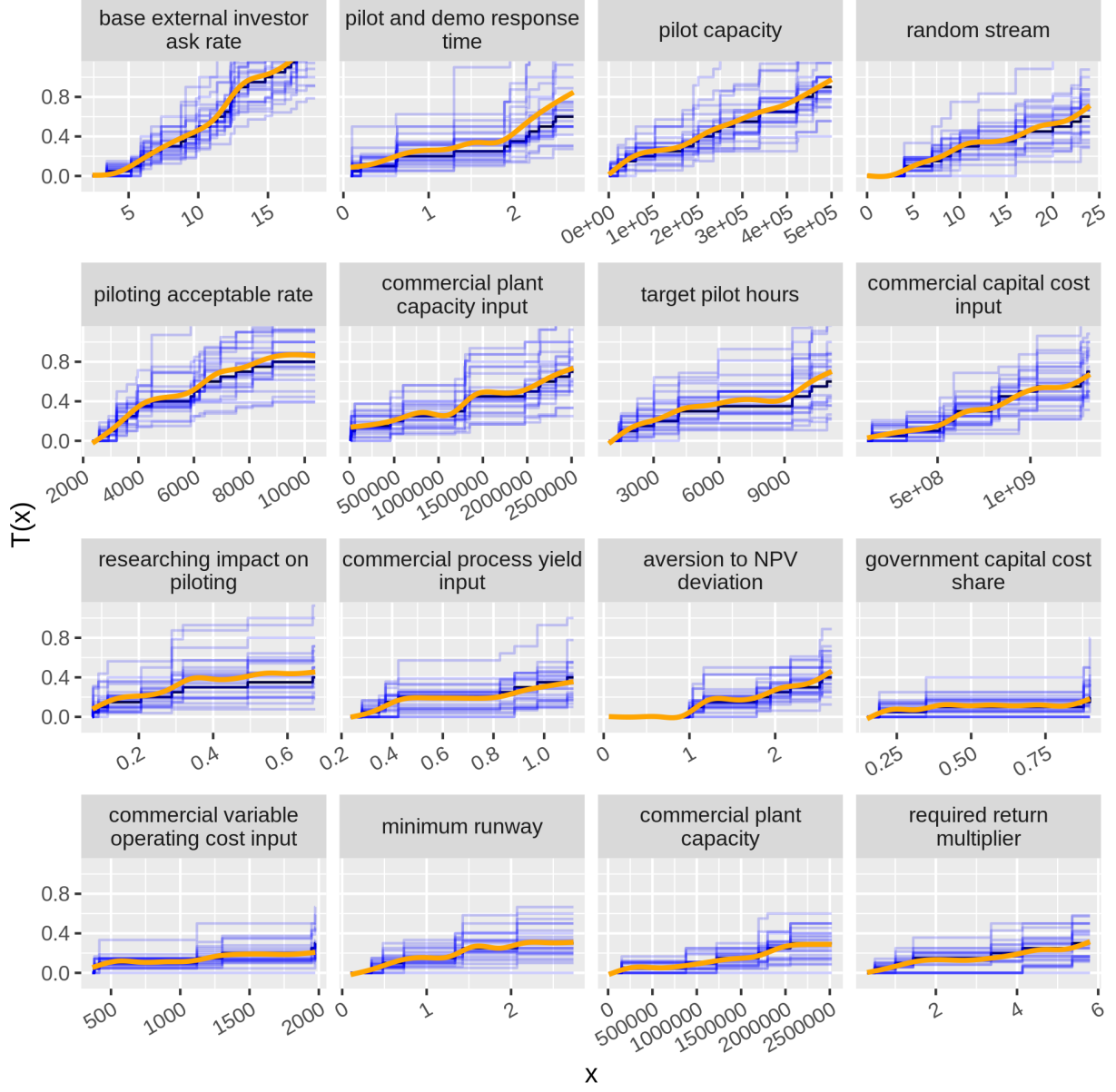


Figure 6: Cumulative local sensitivity $T_{i,j}^{(2,\epsilon)}(x)$ for the BTM model, with $j = \text{"demo plant construction"}$ and $N = M = 2500$.

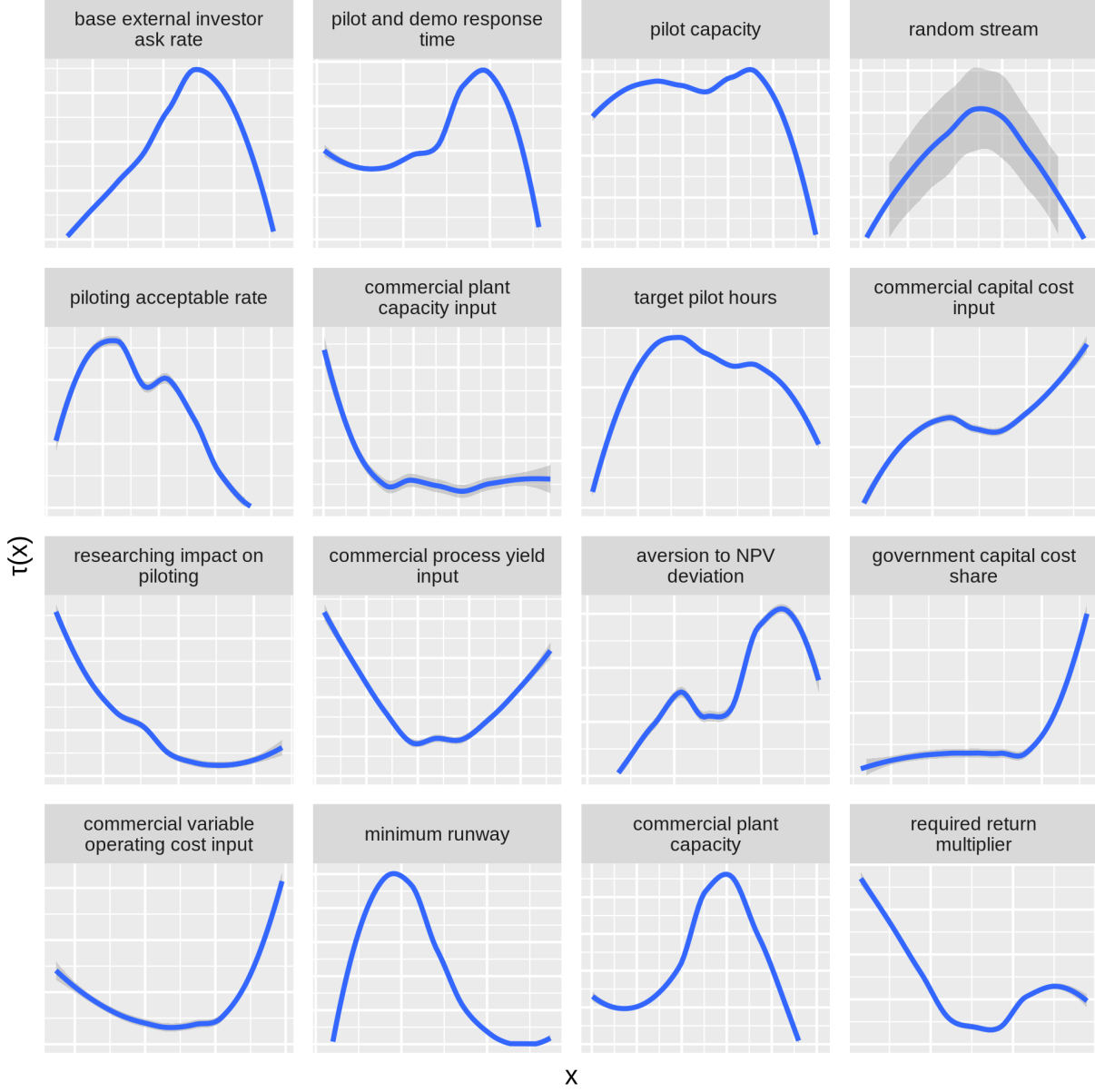


Figure 7: Sensitivity density $\tau_{i,j}^{(2,\epsilon)}(x)$ for the BTD model, with $j = \text{"demo plant construction"}$ and $N = M = 2500$.

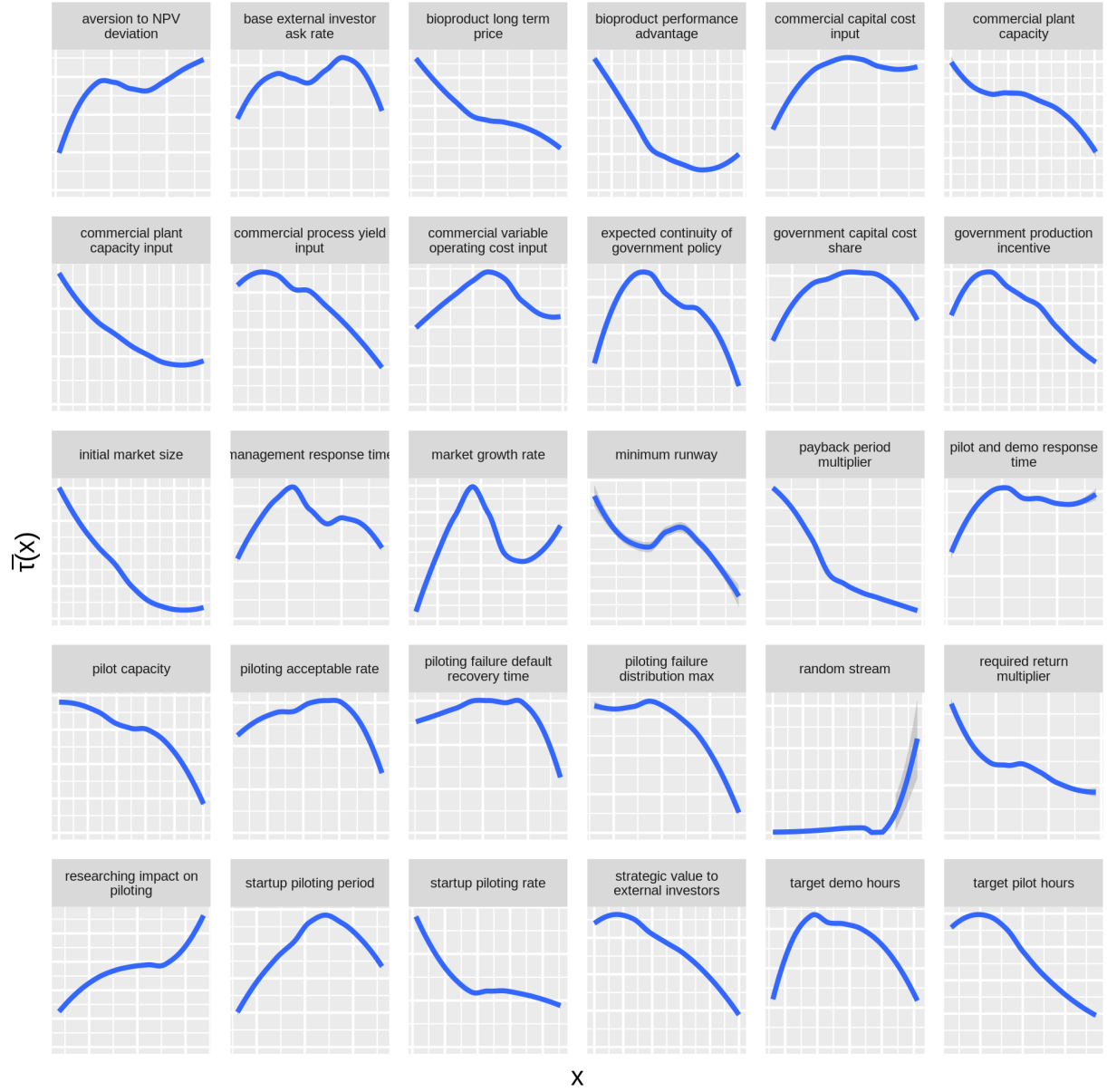


Figure 8: Average sensitivity density $\bar{\tau}_i^{(2,\epsilon)}(x)$ for the BTD model.

Contract No. DE-AC36-08GO28308. Funding provided by the Department of Energy Office of Energy Efficiency and Renewable Energy Bioenergy Technologies Office. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

References

- [Jan99] Michiel J. W. Jansen. “Analysis of variance designs for model output”. en. In: *Computer Physics Communications* 117.1 (Mar. 1999), pp. 35–43. ISSN: 0010-4655. DOI: 10.1016/S0010-4655(98)00154-4. (Visited on 09/20/2020).
- [Sal+10] Andrea Saltelli et al. “Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index”. en. In: *Computer Physics Communications* 181.2 (Feb. 2010), pp. 259–270. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2009.09.018. (Visited on 08/11/2020).

A Toy Model

This toy model consists of a nonlinear ordinary differential equation for y_j as a function of x_i , with $i \in \{1 \dots m\}$ and $j \in \{1 \dots n\}$:

$$\frac{dy}{dt} = \kappa \cdot y \, z \, (1 - \sigma \cdot y \, z),$$

where

$$z_j = \sum_{i=1}^m \mathbb{1}_{j \in L_i} \mathbb{1}_{x_i \geq \xi_i} \zeta_i (x_i - \xi_i)^{\delta_i}.$$

The parameters ξ_i define locations of discontinuities and the parameters δ_i specify the degree of those discontinuities. The parameters L_i affect the mixing of the input variables into the output, as do the matrices κ and σ . The model was designed to exhibit nontrivial and nonlinear behavior with tunable discontinuous behavior. It can be used as a timeseries generator or just evaluated at a specified final time.

The particular instantiation of the toy model studied in this paper uses the following

parameter values:

$$\begin{aligned}
 m &= 3 \\
 n &= 3 \\
 t &\in [0, 10] \\
 L &= [\{2, 3\} \quad \{1, 2\} \quad \{1, 2, 3\}] \\
 \delta &= [0 \quad 1 \quad 2] \\
 \xi &= [0.5933 \quad 0.9485 \quad 0.1030] \\
 \zeta &= [0.8788 \quad 0.2668 \quad 0.6661] \\
 y(0) &= [-0.1900 \quad 0.5145 \quad 0.4094] \\
 \kappa &= \begin{bmatrix} 0.7054 & 0.2921 & 0.7361 \\ -0.1151 & 0.5206 & -0.0707 \\ 0.3475 & -0.0579 & -0.2229 \end{bmatrix} \\
 \sigma &= \begin{bmatrix} 0.0294 & 0.1668 & 0.5788 \\ 0.1046 & 0.1705 & 0.2749 \\ -0.1258 & -0.0712 & 0.7372 \end{bmatrix}
 \end{aligned}$$

The following R code implements this toy model:

```

# Create a multivariate function with specified properties:
#   tmax: maximum time
#   multiplicities: number of correlations each parameter has
#   degrees: polynomial degree of each parameter
#   dimension: the dimension of the output
#   returns a multivariate function of the vector of parameters and time
makeGenerator <- function(multiplicities, degrees, dimension) {
  single <- function(degree) {
    x0 <- runif(1)
    z0 <- runif(1)
    print(paste("Critical point at x = ", x0, sep = ""))
    function(x) {
      if (x < x0)
        0
      else
        z0 * (x - x0)^degree
    }
  }
  locations <- lapply(
    multiplicities,
    function(m) sample(1:dimension, m)
  )
  functions <- lapply(degrees, single)
  start <- runif(dimension, -0.25, 0.75)
  coefs <- matrix(

```

```

        runif(dimension^2, -0.25, 0.75),
        dimension,
        dimension
    )
    shift <- matrix(
        runif(dimension^2, -0.25, 0.75),
        dimension,
        dimension
    )
    function(x, ts) {
        z <- rep(0, dimension)
        for (i in 1:length(locations))
            for (j in locations[[i]])
                z[j] <- z[j] + functions[[i]](x[i])
        ode(start, ts, function(t, y, params) {list((coefs %*% y) * z *
            (1 - ((shift %*% y) * z)))})
    }
}

```

Here is a simple example of the use of this function:

```

# Use reproducible random numbers.
RNGkind("Mersenne-Twister", "Inversion", "Rejection")
set.seed(46)

# Instantiate the model.
f <- makeGenerator(c(2, 2, 3), c(0, 1, 2), 3)
# "Critical point at x = 0.593385165324435"
# "Critical point at x = 0.948547213338315"
# "Critical point at x = 0.102978735696524"

# Evaluate it at x = (0.1, 0.2, 0.3) for t = 0, 5, 10.
f(c(0.1, 0.2, 0.3), c(0, 5, 10))
# time      1      2      3
# 0      -0.1900320  0.5144967  0.4093612
# 5      -0.1478757  0.5489932  0.3864914
#10      -0.1024813  0.5854096  0.3659173

```