Marginal PDF Perturbation Robustness Code Documentation

Joseph Hart[1]

[1]Department of Mathematics, North Carolina State University, Raleigh, NC

1 Introduction

This document overviews the use of the codes contained in

$$Robustness\_Codes/Marginal\_Perturbation$$

which execute the method described in [1]. The folder *Robustness_Codes/Marginal_Perturbation* contains Matlab source code as well as C code generated and precompiled using Matlab Coder. Using the precompiled code is strongly recommended as the algorithm benefits significantly from executing the precompiled C code. The reader is direct to *Robustness_Codes/Manual.pdf* for more information about the precompiled code. Within the directory *Robustness_Codes/Marginal_Perturbation* there are two subdirectories whose names are

src and *Linear_Example*

The user does not need to regard the content of *src* unless he/she would like to modify the algorithm. The subdirectory *Linear_Example* may be copied and edited by the user to generate new examples. Section 2 illustrates the use of the codes in *Linear_Example*. Output of the example is given in Section 3.

2 Example

2.1 Test Problem

Let $\mathbf{X} = (X_1, X_2, X_3)$ where each X_i, $i = 1, 2, 3$, is a independent beta random variable on $[0, 1]$ with shape parameters given in Table 1,

	X_1	X_2	X_3
a	1.2	1.4	1.6
b	1.2	1.4	1.6

Table 1: Beta distribution parameters for X_i, $i = 1, 2, 3$. Each X_i is assumed to be a independent beta random variable on $[0, 1]$ with unnormalized probability density function given by $x^{a-1}(1 - x)^{b-1}$.

and $f : [0, 1]^3 \to \mathbb{R}$ be defined by

$$f(\mathbf{X}) = 1.5X_1 + 1.25X_2 + X_3. \tag{1}$$

This document demonstrates how to use the codes in *Robustness_Codes/Marginal_Perturbation* to compute the first order and total Sobol' indices of (1) and measure their robustness with respect to perturbations of the marginal probability density functions (PDFs). The example is given in

$$Robustness\_Codes/Marginal\_Perturbation/Linear\_Example$$

The example folder *Robustness_Codes/Marginal_Perturbation/Linear_Example* contains a file

Driver.m

which computes the first order and total Sobol' indices and perturbed first order and total Sobol' indices, and a directory

User_Functions

where the user must specifies a collection of Matlab functions. We begin by examining each file in *User_Functions* followed by describing the execution of *Driver.m*. To create a new example, the user is required to implement the functions in *User_Functions*.

2.2 Functions in *User_Functions*

There are three functions in *User_Functions*; one draw samples of $\mathbf{X}$, one evaluates the marginal PDFs of $\mathbf{X}$, and one evaluates f. In what follows we use p to denote the number of inputs (3 in this example) and N to denote the number of samples used for Monte Carlo integration. The support of $\mathbf{X}$ is assumed to be $[0,1]^p$, see [1] for further discussion of this assumption. When developing new examples, the user should transform $\mathbf{X}$ to ensure that it is supported on $[0,1]^p$.

The file *f.m* defines the function f. It inputs a $N \times p$ matrix and returns a vector of length N corresponding to the evaluations of f at each row of the input matrix. If other input arguments are needed to define f, the user may edit *f.m* and must also edit lines 7,8, and 13 of the file *Sobol_Function_Evalution.m* (under *src*) accordingly.

The file *Sample_X.m* defines a function which draws samples of the random vector $\mathbf{X}$. It inputs a integer N, specifying the desired number of samples, and returns a $N \times p$ matrix of samples.

The file *Eval_Phi.m* defines the marginal PDFs of $\mathbf{X}$. It inputs a $N \times p$ matrix and returns a $N \times p$ matrix corresponding to the evaluations of the marginal PDFs at each row of the input matrix. Specifically, the (i,j) entry of the output is the PDF of X_j evaluated at the (i,j) entry of the input.

2.3 Execution of *Driver.m*

The script *Driver.m* is copied below for convenient referencing. This subsection explains the script.

```
restoredefaultpath
clear
clc
addpath(genpath('../src'))
addpath User_Functions/

rng(10123)

% Set constants
N = 5*10^3;
R = (0:.1:1)'*ones(1,3);
tau = 1.5;
r = 60;
m = 50;

[A,B] = Sobol_Sampling(N);
[YA,YB,YC] = Sobol_Function_Evaluation(A,B);
Sobol_Output = Sobol_Indices(A,B,YA,YB,YC);
[sd_S_nom,sd_T_nom] = Sample_std_Estimate(m,Sobol_Output.YA,Sobol_Output.YB,
    Sobol_Output.YC);
Phi_Data = Phi_Data_Generation(Sobol_Output);
save('Preprocessed_Data.mat')
```

```
23 % Perturbation_Output = Marginal_Perturbation_Analysis(Sobol_Output,Phi_Data,r,m,R,
    ↪ sd_S_nom,sd_T_nom);
24 Perturbation_Output = Marginal_Perturbation_Analysis_mex(Sobol_Output,Phi_Data,r,m,
    ↪ R,sd_S_nom,sd_T_nom);

26 % Largest_Perturbation('T',Sobol_Output,Perturbation_Output,tau)
27 % Largest_Perturbation_Variable_i(1,'T',Sobol_Output,Perturbation_Output,tau)

29 save('Output.mat')
```

Lines 1-3 are to ensure a clean workspace and file scope, and Lines 4-5 ensure that the correct functions are in scope. Line 7 sets the random number generator so that the example may be reproduced. This may be edited by the user if he/she desires.

Lines 10-13 are the parameters discussed as inputs of Algorithm 1 in [1]. Line 14 is a parameter specifying the number of random subsamples to generate when estimating the sample standard deviation of the total Sobol' indices.

Lines 16-20 compute the first order and total Sobol' indices and prepares the data needed to perform robustness analysis. Line 16 generates the samples needed to estimate the Sobol' indices. Line 17 is a function which accepts these samples as inputs and evaluates f at each sample. For computationally intensive applications, *Sobol_Function_Evaluation* may be run offline with a computing cluster without effecting any of the robustness codes. Line 18 estimates the first order and total Sobol' indices and Line 19 estimates the sample standard deviation of the estimators. Line 20 precomputes evaluations of the marginal PDFs prior to the robustness analysis. This enables computational savings, and more importantly, enables the robustness codes to be precompiled independently of the specific example a user considers. Line 21 saves this data, this may be omitted if the user desires.

The user may freely edit anything in Lines 1-21 as long as the data

N, R, tau, r, m, A, B, YA, YB, YC, Sobol_Output, sd_nom_S, sd_nom_T, Phi_Data

is stored in the same way as the example. The precompiled robustness codes expect the inputs to be formatted as in this example, but the precompiled codes do not depend on any of the functions in Lines 1-21 so the user is free to edit them.

Line 23 (commented out) executes robustness analysis using the basic Matlab codes and Line 24 executes the robustness analysis using the precompiled codes, which is more efficient. The only difference in the function calls is the additional "*_mex*" included in the function name in Line 24.

Line 26 plots the largest perturbations of either first order or total Sobol' indices, the first argument of the function should be specified as 'S' or 'T' to indicate either the first order or total Sobol' indices. Line 27 plots the perturbations which yield the greatest change to a specific Sobol' index. The first two arguments specify the index. If the first order index for variable i is of interest, the first two arguments should be i,'S'. If the total order index for variable i is of interest, the first two arguments should be i,'T'. In this example, we consider the total Sobol' index for X_1. Line 29 saves the results of the robustness analysis.

3 Example Output

Figure 1 displays the output when *Driver.m* in *Robustness_Codes/Marginal_Perturbation* is executed. The robustness code prints its progress as it executes. Specifically, "Computing Frechet derivative for operator" indicates that the Fréchet derivatives are being computed, the numbers appearing below this indicates which Sobol' index Fréchet derivative is currently being computed (the output 1 indicates that it is computing the Fréchet derivative of S_1 and T_1). The output "Computing perturbed index" indicates that the Fréchet derivatives have been computed and the code is now computing the perturbed Sobol' indices, the numbers below this line indicate which set of perturbed Sobol' indices are currently being computed. There may be up to $2p$ sets of perturbed Sobol' indices, but if multiple perturbations coincide there may be fewer. In this example, there are only 3 sets of perturbed Sobol' indices.

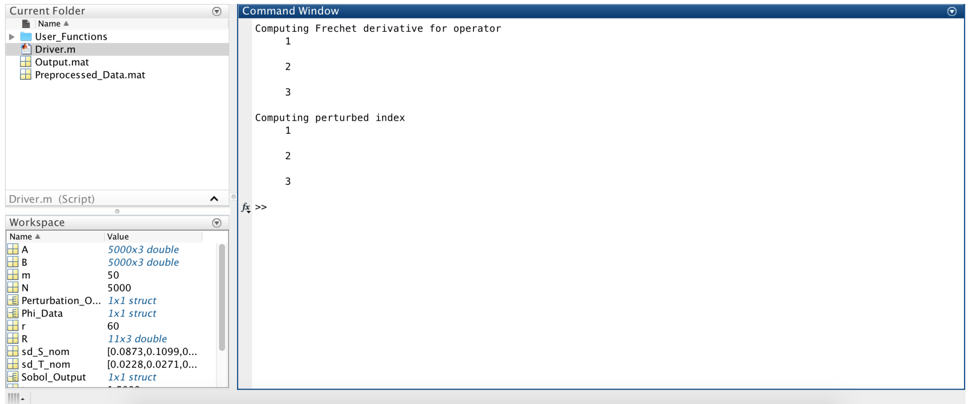

Figure 1: Matlab command line output when *Robustness_Codes/Marginal_Perturbation/Driver.m* is executed.

Figure 2 displays the figures generated by executing

Largest_Perturbation('T',Sobol_Output,Perturbation_Output,tau)

They show the maximum and minimum total Sobol' indices observed over all perturbations. Specially, the blue region indicates the nominal value of the total Sobol' index, the top of the green bar indicates the maximum value (over all perturbations) observed for the total Sobol' index, and the bottom of the magenta bar indicates the minimum value (over all perturbations) observed for the total Sobol' index.

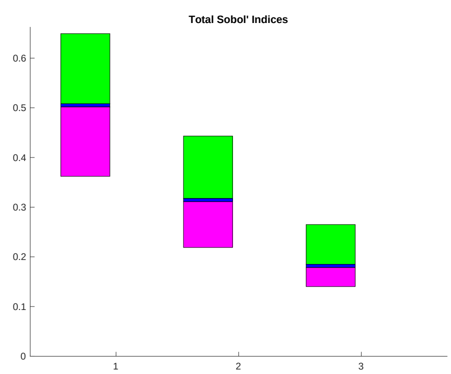

Figure 2: The top of the green bar and bottom of the magenta bar indicate the maximum and minimum values of the total Sobol' indices, computed over all perturbations. The blue region in the centers indicate the values of the nominal total Sobol' indices.

The data depicted in Figure 2 does not correspond to a particular set of perturbed total Sobol' indices, but rather the extreme cases over all sets of perturbed total Sobol' indices. Figure 3 displays the total Sobol' indices (left) and marginal distributions (right) corresponding to the nominal distribution and perturbations which maximum and minimize T_1. Figure 3 was generated by executing

Largest_Perturbation_Variable_i(1,'T',Sobol_Output,Perturbation_Output,tau)

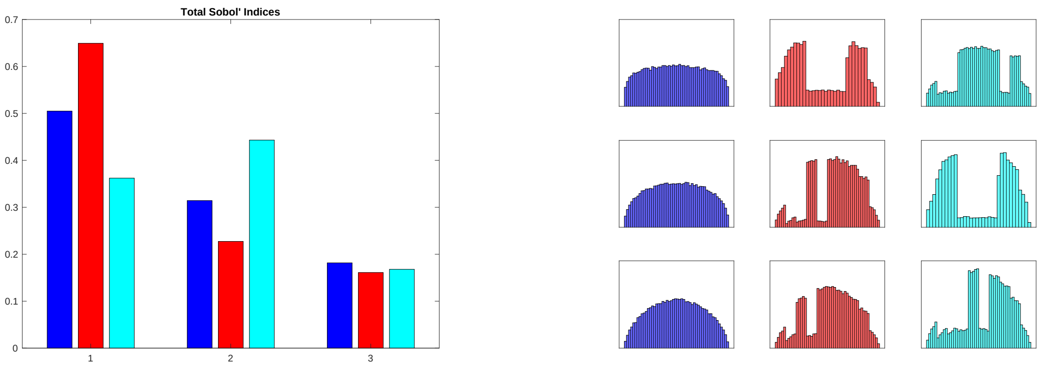

Figure 3: Left: the blue, red, and cyan bars indicates the nominal total Sobol' indices, the perturbed total Sobol' indices when T_1 is maximized, and the perturbed total Sobol' indices when T_1 is minimized, respectively. Right: histograms of samples from the marginals distributions. The top, middle, and bottom rows correspond to X_1, X_2, and X_3, respectively. The left, center, and right columns correspond to the nominal distribution, the perturbed distribution which maximized T_1, and the perturbed distribution which minimized T_1, respectively.

References

[1] Joseph Hart and Pierre Gremaud. Robustness of the Sobol' indices to marginal distribution uncertainty. *In preparation*, 2018.