

# Working Paper M11/01

Methodology

## Bayesian Lightweight Emulators

## For Multivariate Computer Models

Antony M. Overstall, David C. Woods

### Abstract

Statistical emulators for the outputs of complex computer codes (simulators) are typically constructed using nonparametric regression methods, such as Gaussian Process (GP) regression. For many simulators, emulators based on parametric models may provide adequate descriptions whilst enabling straightforward and computationally inexpensive fitting, inference and prediction. We place such so-called “lightweight” emulators into the same Bayesian framework as the more usual nonparametric emulators, and provide methodology for their application to two novel examples with multivariate output: an emergency-relief simulator and a low-level atmospheric dispersion simulator. For the former, the inputs to the simulator are both continuous and categorical, and a comparison is made to GP emulators; for the latter, the output is zeroinflated and an appropriate emulator is developed from a Tobit model. In each case, sensitivity analyses are performed to identify the inputs to the simulator that have a substantive impact on the response, using both traditional methods and Bayesian model selection.

Keywords: Bayesian linear regression; Gaussian Process; Markov Chain Monte Carlo Model Composition; Tobit model; Zero-inflated response.

# Bayesian lightweight emulators for multivariate computer models

Antony M. Overstall and David C. Woods\*  
Southampton Statistical Sciences Research Institute,  
University of Southampton,  
Southampton, UK

Statistical emulators for the outputs of complex computer codes (simulators) are typically constructed using nonparametric regression methods, such as Gaussian Process (GP) regression. For many simulators, emulators based on parametric models may provide adequate descriptions whilst enabling straightforward and computationally inexpensive fitting, inference and prediction. We place such so-called “lightweight” emulators into the same Bayesian framework as the more usual nonparametric emulators, and provide methodology for their application to two novel examples with multivariate output: an emergency-relief simulator and a low-level atmospheric dispersion simulator. For the former, the inputs to the simulator are both continuous and categorical, and a comparison is made to GP emulators; for the latter, the output is zero-inflated and an appropriate emulator is developed from a Tobit model. In each case, sensitivity analyses are performed to identify the inputs to the simulator that have a substantive impact on the response, using both traditional methods and Bayesian model selection.

*Keywords:* Bayesian linear regression; Gaussian Process; Markov Chain Monte Carlo Model Composition; Tobit model; Zero-inflated response.

## 1. Introduction

For many systems and processes in science and engineering, conducting a real physical experiment may be infeasible due to it being economically prohibitive, unethical, or even impossible. Some examples include ecosystems, modelling infectious diseases, climate models, and galaxy formation. In such cases, it is increasingly common for the scientist or engineer to develop a *computer model* (also known as a *simulator*) that aims to provide a description of the physical system. The simulator is a mathematical function, which can be deterministic or stochastic, that maps the inputs and outputs of the system. Therefore the input space of the system is the domain of the function and the output space is the range.

However, due to the complexity of the simulator, the function may be computationally expensive to evaluate, making statistical inference time consuming. In which case, a *computer experiment* (Sacks et al., 1989) can be used, with the computer model evaluated for a collection of points within the input space. We then use those evaluations of the simulator to build a statistical model, or *emulator*, to predict the output of the system at any set of input points without having to run the simulator. With the prediction should come some associated measure of uncertainty. The emulator can be used to replace and supplement the simulator in any number of tasks, including optimization, inference, calibration and validation. For more on computer experiments, see Kennedy and O’Hagan (2001), Santner, Williams and Notz (2003) or Fang, Li and Sudjianto (2006).

In this article, we develop and investigate methodology for two examples of emulating computer models:

---

\* *Address for correspondence:* David Woods, School of Mathematics, University of Southampton, Southampton, SO17 1BJ, UK. E-mail: D.Woods@southampton.ac.uk

1. In Section 3, the simulator models a humanitarian relief scenario in Sicily following an eruption of Mount Etna. The simulator output is multivariate, dynamic and continuous.
2. In Section 4, the simulator models the dispersion of particles after a chemical or biological release. Here, the output is multivariate and zero-inflated.

The standard method for emulating a simulator with continuous output is the Gaussian Process (GP) model. However, in this article we discuss the alternative methodology of lightweight emulation, introduced by Rougier (2007). We describe both methods within a Bayesian framework in Sections 2 and 3, and highlight their differences. In Section 4 the lightweight emulator is generalised to zero-inflated output. Some discussion is given in Section 5.

## 2. Statistical emulators

In this Section, we describe the principles of Bayesian emulation that will be applied to simulators in Sections 3 and 4.

Let  $\mathbf{x} = (x_1, \dots, x_p)^T \in \mathcal{X} \subset \mathbb{R}^p$  denote the vector of  $p$  input variables, where  $\mathcal{X}$  is the  $m$ -dimensional input space. We have a vector-valued function

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))^T,$$

which is the  $k \times 1$  vector of output from the simulator at the combination of input variables  $\mathbf{x}$ . We assume the simulator is a black-box function,  $\mathbf{f} : \mathcal{X} \rightarrow \mathcal{Y} \subset \mathbb{R}^k$ , where  $\mathcal{Y}$  is the  $k$ -dimensional output space. As mentioned in Section 1, evaluating  $\mathbf{f}(\cdot)$  may be computationally expensive so we wish to construct an emulator, i.e. a surrogate for  $\mathbf{f}(\cdot)$  that can be used to predict  $\mathbf{f}(\mathbf{x}_0)$  at a combination,  $\mathbf{x}_0$ , of input variables at which we have not previously evaluated  $\mathbf{f}(\cdot)$ .

We begin by specifying a design,  $\zeta = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ , where each treatment, or input point,  $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^T$  defines a combination of values of the input variables. We evaluate the computer model at each  $\mathbf{x}_i$  and obtain  $\mathbf{f}(\mathbf{x}_1), \dots, \mathbf{f}(\mathbf{x}_n)$ . Let

$$\mathbf{Y} = \begin{pmatrix} \mathbf{f}(\mathbf{x}_1)^T \\ \vdots \\ \mathbf{f}(\mathbf{x}_n)^T \end{pmatrix},$$

denote the  $n \times k$  matrix of outputs. We assume some statistical model for  $\mathbf{Y}$ , i.e.

$$\mathbf{Y}|\boldsymbol{\theta} \sim \text{Model}(\boldsymbol{\theta}),$$

where  $\boldsymbol{\theta} \in \Theta$  is a  $d \times 1$  vector of unknown model parameters and  $\Theta$  is the parameter space. In this article, we use a Bayesian approach and so incorporate any available prior information on  $\boldsymbol{\theta}$  in the prior distribution, with probability distribution function (pdf)  $\pi(\boldsymbol{\theta})$ .

Here, the emulator will be the posterior predictive distribution (see, for example, O'Hagan and Forster, 2004, pg 89) of  $\mathbf{y}_0 = \mathbf{f}(\mathbf{x}_0)$ , given by

$$\pi(\mathbf{y}_0|\mathbf{Y}) = \int_{\Theta} \pi(\mathbf{y}_0|\boldsymbol{\theta}, \mathbf{Y})\pi(\boldsymbol{\theta}|\mathbf{Y})d\boldsymbol{\theta}, \quad (1)$$

where  $\pi(\boldsymbol{\theta}|\mathbf{Y})$  denotes the pdf of the posterior distribution of  $\boldsymbol{\theta}$ , found using Bayes theorem, and  $\pi(\mathbf{y}_0|\boldsymbol{\theta}, \mathbf{Y})$  denotes the distribution of  $\mathbf{y}_0$  conditional on  $\boldsymbol{\theta}$  and  $\mathbf{Y}$ . Suppose  $k = 1$ , so

that  $\mathbf{f}(\mathbf{x}) = f(\mathbf{x})$  is scalar, and that the output is continuous over the real line. In this case, the standard emulator is the Gaussian process model, in which it is assumed that  $\mathbf{Y}$  ( $n \times 1$ ) is multivariate normal and that the elements of  $\mathbf{Y}$  are correlated according to the distance between their respective input points. The Gaussian process model can be generalised to  $k > 1$  (Conti and O’Hagan, 2010) and we describe this emulator in Section 3.

As we shall see in Sections 3 and 4 the key to a *lightweight emulator* is that, conditional on  $\boldsymbol{\theta}$ , the distribution of  $\mathbf{y}_0$  is independent of  $\mathbf{Y}$ . Therefore, for a lightweight emulator, (1) can be expressed as

$$\pi(\mathbf{y}_0|\mathbf{Y}) = \int_{\Theta} \pi(\mathbf{y}_0|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\mathbf{Y})d\boldsymbol{\theta}, \quad (2)$$

where the integrand in (2) is the product of the likelihood of  $\mathbf{y}_0$  and the posterior distribution of  $\boldsymbol{\theta}$ . In Section 3, we demonstrate that modelling the correlation between the rows of  $\mathbf{Y}$ , and between  $\mathbf{Y}$  and  $\mathbf{y}_0$  is non-trivial. By assuming they are independent we introduce the lightweight concept and remove the need to model these correlations.

In Sections 3 and 4 we develop emulators appropriate for our examples, and describe their application.

### 3. A humanitarian relief scenario: a simulator with multivariate dynamic output

#### 3.1. Introduction

The simulator we consider in this section is an operational/campaign level simulation model for emergency planning and peace support operations. The primary focus of the simulator is to assess the effectiveness of variations in force mixes, in scales of effort and in differing command and control structures. The model is mission-based, using a command and control structure to deconstruct high level plans into objectives and thence to missions which are allocated to individual entities. It is truly multi-sided, with no limit on the number of parties, full functionality being available to all parties. Parties may represent a wide range of agents: military forces in non-warfighting roles, non-governmental organisations, recruited indigenous forces, civilians, etc. Variations to possible courses of action, including rates of repair (or degradation) to the infrastructure can be incorporated to determine how these affect the outcome.

The scenario we consider here simulates a humanitarian relief mission to Sicily after an eruption of Mount Etna, which damages the food supply and housing (shelter) at the cities of Giarre and Catania. An non-governmental organisation (NGO) launches a humanitarian relief mission which has the following components

- *Food Aid Mission*  
To supply food to Catania and Giarre, by using helicopters to transport food from the NGO base.
- *Repair Mission*  
To transport engineers from the NGO base to Giarre and Catania, where they repair the food supply infrastructure and/or the shelter.

As an illustration, we consider a scenario where the NGO has three helicopters, two engineering teams and a food depot. Two of the helicopters are assigned to the food aid mission and one to transporting the engineers for the repair mission.

The output of the computer model is the number of civilian deaths that have occurred in the previous 24 hours (in 100,000's) from the end of day two, to the end of day six, where the days are counted from the moment of the eruption.

The input variables are the features of the humanitarian relief mission. There are five continuous input variables and two categorical, each with two levels. The continuous input variables are:

- *Weighting of the engineer toolbox,  $x_1$ .*  
The two engineers on the repair mission have a weighting that assigns the amount of repair to the food supply the engineers perform relative to the amount of shelter repair. It can vary in the interval  $(0, 1)$ .
- *Planning time for the humanitarian mission,  $x_2$ .*  
This input variable defines the time from the eruption to when the humanitarian mission starts. It can vary in the interval  $(36, 60)$  hours.
- *Speed of the helicopters,  $x_3$ .*  
This input variable defines the speed of the NGO helicopters. It can vary in the interval  $(220, 270)$  km/hr.
- *Capacity of the helicopters,  $x_4$ .*  
This input variable defines the capacity of the NGO helicopters on the food aid mission. It can vary in the interval  $(7000, 7500)$ . The units of this input variable are specific to this computer model.
- *Speed of engineers,  $x_5$ .*  
This input variable defines the ground speed of the engineering team. It can vary in the interval  $(0, 10)$  km/hr.

All of the continuous input variables are scaled to lie in the interval  $[0, 1]$ . The categorical input variables are:

- *Recipient of the food aid mission,  $z_1$ .*  
Level one denotes the situation when one helicopter on the food aid mission supplies food to Giarre and the other to Catania. Level two denotes when both helicopters supply food to Catania. The rationale for just supplying food to Catania is that here there is a larger shortfall between the available food and shelter, and the required food and shelter.
- *Location of the NGO base,  $z_2$ .*  
Level one denotes the situation when the NGO base is located in continental Europe. Level two denotes when the NGO base is located at the taskforce base (a fleet of ships in the sea between Sicily and Italy).

For the categorical input variables, level one is coded as 0 and level two as 1.

## 3.2. Methodology

**3.2.1. Design** The most common design used with Gaussian Process emulators is the Latin Hypercube (McKay, Conover, and Beckman, 1979) and its extensions (see, for example, Tang,

1993, and Morris and Mitchell, 1995). Such space-filling designs provide low-dimensional uniformity in the input variables, hence achieving good projection properties, and allow the estimation of nonparametric regression models. They are also an attractive choice for lightweight emulation, as the exact form of the emulator will be unknown in advance of the data collection and a flexible design that allows the fitting of many different parametric models is required.

The design,  $\zeta = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ , for this study needs to combine both continuous and categorical input variables. We employed a sliced space-filling design as proposed by Qian and Wu (2009) with  $n = 64$  runs. Such a design, constructed from a particular orthogonal array, has not only good space-filling properties overall but also for the projection into the continuous variables for each combination of values of the categorical input variables.

**3.2.2. Multivariate emulator** We now extend the multi-dimensional Gaussian process model, proposed by Conti and O’Hagan (2010), to include both continuous and categorical input variables using the correlation structures of Qian, Wu and Wu (2008). Initially suppose the input variables are continuous. Let

$$\mathbf{H} = \begin{pmatrix} h(\mathbf{x}_1)^T \\ \vdots \\ h(\mathbf{x}_n)^T \end{pmatrix}$$

be the  $n \times m$  model matrix of regressors where  $h : [0, 1]^p \rightarrow \mathcal{S} \subset \mathbb{R}^m$ . We assume that

$$\mathbf{Y}|\mathbf{B}, \boldsymbol{\Sigma}, \mathbf{A} \sim \text{MN}_{n,k}(\mathbf{H}\mathbf{B}, \boldsymbol{\Sigma}, \mathbf{A}), \quad (3)$$

where  $\text{MN}_{n,k}(\mathbf{M}, \mathbf{C}, \mathbf{R})$  denotes the  $n \times k$  matrix-normal distribution (Dawid, 1981) with mean  $\mathbf{M}$  (an  $n \times k$  matrix), a column covariance matrix  $\mathbf{C}$  (a  $k \times k$  matrix) and a row correlation matrix  $\mathbf{R}$  (an  $n \times n$  matrix). Note that

$$\text{vec}(\mathbf{Y})|\mathbf{B}, \boldsymbol{\Sigma}, \mathbf{A} \sim \text{N}_{nk}(\text{vec}(\mathbf{H}\mathbf{B}), \boldsymbol{\Sigma} \otimes \mathbf{A}),$$

is a multivariate normal distribution, where  $\text{vec}(\cdot)$  denotes the vectorisation function that stacks columns of a matrix, and  $\otimes$  denotes the Kronecker product.

In (3),  $\mathbf{B}$  is an  $m \times k$  matrix of unknown regression parameters,  $\boldsymbol{\Sigma}$  is an unknown  $k \times k$  column covariance matrix and  $\mathbf{A}$  is an  $n \times n$  row covariance matrix. The  $ij$ th element of  $\mathbf{A}$  gives the correlation between the  $i$ th and  $j$ th runs of the computer experiment, denoted as  $a(i, j)$ . The  $(r, s)$ th element of  $\boldsymbol{\Sigma}$  gives the covariance between the  $r$ th and  $s$ th elements of  $\mathbf{y}(\mathbf{x})$ , i.e. the covariance between different elements of the output vector for the same input variables.

We use the conditionally conjugate (given  $\mathbf{A}$ ) matrix-normal-inverse-Wishart (MNIW) prior distribution for  $\mathbf{B}$  and  $\boldsymbol{\Sigma}$ , denoted by  $\text{MNIW}_{m,k}(\mathbf{M}, \boldsymbol{\Omega}, \mathbf{S}, \delta)$ , where

$$\mathbf{B}|\boldsymbol{\Sigma} \sim \text{MN}_{m,k}(\mathbf{M}, \boldsymbol{\Sigma}, \boldsymbol{\Omega}), \quad (4)$$

$$\boldsymbol{\Sigma} \sim \text{IW}_k(\mathbf{S}, \delta). \quad (5)$$

Here,  $\text{IW}_k$  denotes the inverse-Wishart distribution for  $k \times k$  positive-definite matrices,  $\mathbf{S}$  is the  $k \times k$  scale matrix and  $\delta > 0$  is the degrees of freedom. We use the parameterisation of the inverse-Wishart distribution described by Rougier (2007).

Using this prior distribution the conditional posterior distribution given  $A$  is

$$\mathbf{B}, \boldsymbol{\Sigma} | \mathbf{Y}, \mathbf{A} \sim \text{MNIW}_{m,k} \left( \hat{\mathbf{M}}, \hat{\boldsymbol{\Omega}}, \hat{\mathbf{S}}, \hat{\delta} \right),$$

where

$$\begin{aligned} \hat{\boldsymbol{\Omega}} &= (\mathbf{H}^T \mathbf{A}^{-1} \mathbf{H} + \boldsymbol{\Omega}^{-1})^{-1}, \\ \hat{\mathbf{M}} &= \hat{\boldsymbol{\Omega}} (\mathbf{H}^T \mathbf{A}^{-1} \mathbf{Y} + \boldsymbol{\Omega}^{-1} \mathbf{M}), \\ \hat{\mathbf{S}} &= \mathbf{Y}^T \mathbf{A}^{-1} \mathbf{Y} + \mathbf{M}^T \boldsymbol{\Omega}^{-1} \mathbf{M} + \mathbf{S} - \hat{\mathbf{M}}^T \hat{\boldsymbol{\Omega}}^{-1} \hat{\mathbf{M}}, \\ \hat{\delta} &= \delta + n. \end{aligned}$$

As mentioned in Section 2, the emulator is the posterior predictive distribution of  $\mathbf{y}_0$  at unobserved inputs,  $\mathbf{x}_0$ , given by (1). First let  $c(\mathbf{x}_i, \mathbf{x}_j)$  be a function describing the correlation between different runs of the simulator such that  $c : \mathcal{X}^2 \rightarrow [0, 1]$  and  $c(\mathbf{x}_i, \mathbf{x}_i) = 1$ . Note that

$$\begin{pmatrix} \mathbf{Y} \\ \mathbf{y}_0 \end{pmatrix} \Big| \mathbf{B}, \boldsymbol{\Sigma}, \mathbf{A}, \mathbf{t}, a \sim \text{MN}_{n+1,k} \left( \begin{pmatrix} \mathbf{H} \\ \mathbf{h}_0 \end{pmatrix} \mathbf{B}, \boldsymbol{\Sigma}, \begin{pmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{t}^T & a \end{pmatrix} \right),$$

where  $\mathbf{h}_0 = h(\mathbf{x}_0)$ ,  $a = c(\mathbf{x}_0, \mathbf{x}_0) = 1$ ,  $\mathbf{t} = (t_1, \dots, t_n)^T$ , and  $t_j = c(\mathbf{x}_j, \mathbf{x}_0)$ , for  $j = 1, \dots, n$ . Then

$$\mathbf{y}_0 | \mathbf{B}, \boldsymbol{\Sigma}, \mathbf{A}, \mathbf{t}, c, \mathbf{Y} \sim \text{N}(\mathbf{m}^*, a^* \boldsymbol{\Sigma}), \quad (6)$$

where

$$\begin{aligned} \mathbf{m}^* &= \mathbf{B}^T (\mathbf{h}_0 - \mathbf{H}^T \mathbf{A}^{-1} \mathbf{t}) + \mathbf{Y}^T \mathbf{A}^{-1} \mathbf{t}, \\ a^* &= a - \mathbf{t}^T \mathbf{A}^{-1} \mathbf{t}. \end{aligned}$$

The distribution in (6) is equivalent to  $\pi(\mathbf{y}_0 | \boldsymbol{\theta}, \mathbf{Y})$  in (1). Now, conditional on  $\mathbf{A}$ ,  $\mathbf{t}$  and  $a$ , the posterior predictive distribution of  $\mathbf{y}_0$  is

$$\mathbf{y}_0 | \mathbf{A}, \mathbf{t}, c, \mathbf{Y} \sim t_k \left( \mathbf{m}^{**}, \frac{a^{**} \hat{\mathbf{S}}}{\hat{\delta}}, \hat{\delta} \right), \quad (7)$$

where

$$\mathbf{m}^{**} = \hat{\mathbf{M}}^T (\mathbf{h}_0 - \mathbf{H}^T \mathbf{A}^{-1} \mathbf{t}) + \mathbf{Y}^T \mathbf{A}^{-1} \mathbf{t}, \quad (8)$$

$$a^{**} = a^* + (\mathbf{h}_0 - \mathbf{H}^T \mathbf{A}^{-1} \mathbf{t})^T \hat{\boldsymbol{\Omega}} (\mathbf{h}_0 - \mathbf{H}^T \mathbf{A}^{-1} \mathbf{t}). \quad (9)$$

In (7),  $t_k(\boldsymbol{\mu}, \mathbf{R}, \nu)$  denotes the  $k$ -variate t-distribution with location  $\boldsymbol{\mu}$ , scale matrix  $\mathbf{R}$  and  $\nu$  degrees of freedom, (see Kotz and Nadarajah, 2004). Note that (7) is the same distribution derived by Conti and O'Hagan (2010) with a non-informative joint prior distribution used for  $\mathbf{B}$  and  $\boldsymbol{\Sigma}$ .

To make use of (7), we need to know  $\mathbf{A}$ ,  $\mathbf{t}$  and  $a$ , or, equivalently, the function  $c(\cdot, \cdot)$ . The Gaussian process proceeds by setting  $c(\cdot, \cdot)$  to be dependent on the 'distance' between the input variables, i.e.

$$c(\mathbf{x}_i, \mathbf{x}_j) = s(|\mathbf{x}_i - \mathbf{x}_j|).$$

A common form for  $c(\cdot, \cdot)$  is

$$c(\mathbf{x}_i, \mathbf{x}_j) = \exp \left( - \sum_{l=1}^p r_l |x_{il} - x_{jl}|^\rho \right), \quad (10)$$

for  $r_l > 0$  and  $0 < \rho \leq 2$  (see, for example, Fang et al., 2006, pg 145). Typically,  $\rho = 2$ , resulting in a process with infinitely differentiable paths, and this is the correlation function used throughout this article.

Let  $\mathbf{r} = (r_1, \dots, r_p)^T$  be the  $p \times 1$  vector of correlation parameters and denote the prior distribution of  $\mathbf{r}$  as  $\pi_{\mathbf{r}}(\mathbf{r})$ . There does not exist a conjugate prior distribution for  $\mathbf{r}$  but we can generate a sample from the marginal posterior distribution,  $\mathbf{r}|\mathbf{Y}$ , with pdf

$$\pi(\mathbf{r}|\mathbf{Y}) \propto \pi_{\mathbf{r}}(\mathbf{r}) |\mathbf{A}|^{-\frac{k}{2}} |\hat{\mathbf{\Omega}}|^{\frac{k}{2}} |\hat{\mathbf{S}}|^{-\frac{\delta+k-1}{2}},$$

see Conti and O'Hagan (2010), and then evaluate the posterior predictive distribution. This is the fully Bayesian approach.

However, this is a computationally intensive approach. Conti and O'Hagan (2010), amongst others, suggested substituting a 'plug-in' estimate,  $\hat{\mathbf{r}}$ , for  $\mathbf{r}$  in (7), where  $\hat{\mathbf{r}}$  is some representative value of  $\mathbf{r}$  relative to its marginal posterior distribution. Sensible values for  $\hat{\mathbf{r}}$  are the mean or element-wise median from a posterior sample, or the posterior mode.

An alternative approach to the Gaussian process is the lightweight emulator (LWE) proposed by Rougier (2007). In this case, we assume that the correlation structure is completely specified, i.e. we know  $\mathbf{A}$  a-priori. Typically we assume the runs of the computer model are independent, so that

$$c(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} 1, & \text{if } \mathbf{x}_i = \mathbf{x}_j, \\ 0, & \text{otherwise,} \end{cases}$$

and hence  $\mathbf{A} = \mathbf{I}_n$ ,  $\mathbf{t} = \mathbf{0}$  and  $c = 1$ . Then the distribution in (6) is independent of  $\mathbf{Y}$  given  $\mathbf{B}$  and  $\mathbf{\Sigma}$ , i.e. is equivalent to  $\pi(\mathbf{y}_0|\boldsymbol{\theta})$  given in (2). Therefore, the posterior predictive distribution of  $\mathbf{y}_0$  is

$$\mathbf{y}_0|\mathbf{Y} \sim t_k \left( \hat{\mathbf{M}}^T \mathbf{h}_0, \frac{1 + \mathbf{h}_0^T \hat{\mathbf{\Omega}} \mathbf{h}_0}{\hat{\delta}} \hat{\mathbf{S}}, \hat{\delta} \right), \quad (11)$$

and the LWE is a multivariate Bayesian linear regression (see, e.g., O'Hagan and Forster, 2004, Ch. 11).

In the Gaussian process, we typically use a very simple  $h(\cdot)$ , for example,  $h(\mathbf{x}) = 1$  with  $m = 1$  or  $h(\mathbf{x}) = (1, \mathbf{x}^T)^T$  with  $m = p + 1$ . Most of the modelling effort in a Gaussian process is through the correlation structure. The reverse is true for the LWE, where there is no correlation structure so all of our modelling effort is in the mean structure. So for the LWE,  $h(\cdot)$  will be more complex, typically a polynomial function of several orders, and must be chosen and assessed carefully.

The Gaussian process will interpolate the observed output, i.e. at an observed input,  $\mathbf{x}_i$ , the mean of the posterior predictive distribution will be  $\mathbf{y}(\mathbf{x}_i)$  and the variance will be zero. To see this, suppose, without loss of generality, that  $\mathbf{x}_0 = \mathbf{x}_i$ , for some  $i = 1, \dots, n$ . In this case,  $\mathbf{t}$  will be the  $i$ th row (and column) of  $\mathbf{A}$ . Therefore  $\mathbf{A}^{-1}\mathbf{t}$  will be a vector of zeros except for the  $i$ th element which is one. It follows that  $\mathbf{h}_0 - \mathbf{H}^T \mathbf{A}^{-1}\mathbf{t} = \mathbf{0}$  and  $\mathbf{t}^T \mathbf{A}^{-1}\mathbf{t} = 1$ . Therefore  $\mathbf{m}^{**}$  will be the  $i$ th row of  $\mathbf{Y}$  corresponding to  $\mathbf{y}(\mathbf{x}_i)$  and  $c^{**} = 0$  resulting in the variance of

the distribution given by (7) being  $\mathbf{0}$ . The LWE does not have this property. Rougier (2007) defends this shortcoming by suggesting that this will make little practical difference if the predictive variance is small and that it is often the case that we are only interested in a subset of the input variables (known as active inputs) and therefore we will have unexplained variance due to variation in the inactive inputs.

We can immediately apply the LWE to emulate the computer model in this application. To cope with the categorical input variables we just define dummy variables within  $h(\cdot)$  as with linear regression.

The Gaussian process described above is only defined for continuous input variables. Qian et al. (2008) consider several approaches for defining ‘distances’ between categorical input variables at different levels for single output Gaussian processes. Suppose that  $\mathbf{x}_i$  denotes the input point for the  $i$ th run. Let

$$\mathbf{x}_i = \left( \mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)} \right),$$

where  $\mathbf{x}_i^{(1)}$  denotes the  $p_1$  continuous input variables and  $\mathbf{x}_i^{(2)}$  denotes the  $p_2$  categorical input variables, so that  $p_1 + p_2 = p$ . Then the exchangeable correlation structure proposed by Qian et al. (2008) has a correlation function given by

$$c(\mathbf{x}_i, \mathbf{x}_j) = \exp \left( - \sum_{l=1}^{p_1} r_l^{(1)} |x_{il}^{(1)} - x_{jl}^{(1)}|^\rho - \sum_{l=1}^{p_2} r_l^{(2)} \mathbb{I}[x_{il} \neq x_{jl}] \right),$$

where  $\mathbf{r} = (r_l^{(1)}, r_l^{(2)})$  is the vector of correlation parameters. Qian et al. (2008) also considered multiplicative, group and ordinal correlation structures but all these approaches collapse to a common approach when the categorical input variables only have two levels, as in our application. When the categorical input variables have two levels, coded 0 and 1, this correlation function reduces to (10).

As the elements of  $\mathbf{y}(\mathbf{x})$  in this application correspond to the number of civilian deaths at  $k$  different time points,  $t_1, \dots, t_k$ , we can follow Conti and O’Hagan (2010) and consider a time-input Gaussian process. In this case, the output is one-dimensional but we include time as an extra input. The main computational burden of the multi-output Gaussian process is the repeated inversion of the  $n \times n$  matrix,  $\mathbf{A}$ . In the time-input Gaussian process this matrix is  $nk \times nk$  (in this application, a  $320 \times 320$  matrix). However, since the time input variables are the same for each run, the inverse of the  $\mathbf{A}$  matrix can be written as a kronecker product of the inverse of an  $n \times n$  matrix and the inverse of a  $k \times k$  matrix (Rougier, 2008). So the computational burden is still higher for the time-input Gaussian process but not by as much as may have been first thought.

### 3.3. Results

We apply three broad approaches to emulating the computer model in this application: multi-output Gaussian process (MO), time-input Gaussian process (TI) and the lightweight emulator (LWE).

To compare these approaches, we generate an independent test data set of 64 runs. There are four unique combinations of the categorical input variables so we assign 16 runs to each unique combination. For each of the 16 runs in each unique combination, we generate the values of the five continuous input variables from the uniform distribution on  $[0, 1]$ . We can

then compare the posterior predictive distribution at each run of the test design to the true output.

In the terminology of Section 3.2, for the LWE and MO model  $n_{LWE} = n_{MO} = 64$  and  $k_{LWE} = k_{MO} = 5$ . Whereas for the TI model,  $n_{TI} = 320$  and  $k_{TI} = 1$ . The number of input variables for the LWE and MO model are  $p_{LWE} = p_{MO} = 7$  and, for the TI model,  $p_{TI} = 8$ .

For the MO and TI emulators, we use a set of regressors that includes an intercept and linear terms in each variable, and hence  $m_{MO} = 8$  and  $m_{TI} = 9$ . For the LWE, where it is necessary to use a more detailed model for the mean response, the regressors are the intercept, linear terms in each variable, two-way interactions between all the variables and quadratic terms in the continuous variables. Therefore,  $m_{LWE} = 34$ . In each case, the regressors for the training data set are held in the matrix  $\mathbf{H}$ .

We specify joint non-informative prior distribution for  $\mathbf{B}$  and  $\Sigma$  for all three approaches, where in (4) and (5)

$$\begin{aligned} \mathbf{M} &= \mathbf{0}, \\ \mathbf{\Omega}^{-1} &= \mathbf{0}, \\ \mathbf{S} &= \mathbf{0}, \\ \delta &= -2k. \end{aligned}$$

Although these values specify a improper prior distribution, the posterior distribution, conditional on  $\mathbf{A}$ , will remain proper (Berger et al. 2001).

For the prior distribution of the correlation parameters  $\mathbf{r}$  under the Gaussian process models, MO and TI, we specify

$$\pi_{\mathbf{r}}(\mathbf{r}) = \prod_{j=1}^p (1 + r_j^2)^{-1}, \quad (12)$$

see also Conti and O’Hagan (2010).

For the Gaussian process models, MO and TI, we find the marginal posterior mode of  $\mathbf{r}_{MO}$  and  $\mathbf{r}_{TI}$ , respectively, using a quasi-Newton method. These posterior modes, denoted by  $\hat{\mathbf{r}}_{MO}$  and  $\hat{\mathbf{r}}_{TI}$ , respectively, are then used in the plug-in approach described in Section 3.2.

We evaluate the posterior predictive distribution for each of the three approaches at each run of the test data set. To compare the different approaches we employ the diagnostic tools from Conti and O’Hagan (2010), using residuals from the test data.

The raw residuals for the test data are defined to be the true output at a particular time point minus the posterior predictive means. Standardised residuals are obtained from the raw residuals by dividing by the posterior predictive standard deviations. Under the model assumptions, for each time point, the standardised residuals should be a sample of size 64 from the central t-distribution with  $\hat{\delta}$  degrees of freedom. This property can be assessed by using the usual quantile-quantile (QQ) plot. Figure 1 shows the QQ plots for each of the five time points and each of the three methods. All three methods exhibit heavy-tailed standardised residuals, which according to Conti and O’Hagan (2010), is also common in single-output emulators.

Table 3.3 shows the average root mean square prediction error, based on the standardised residuals over the 64 runs and five time points. This value, under the model assumptions, has expected value of one (Conti and O’Hagan, 2010). The table also gives the frequentist coverage of the 95% probability intervals from the posterior predictive distribution, the relative mean

Figure 1: Quantile-quantile plots of the standardised residuals for the five different time points and for the three different methods.

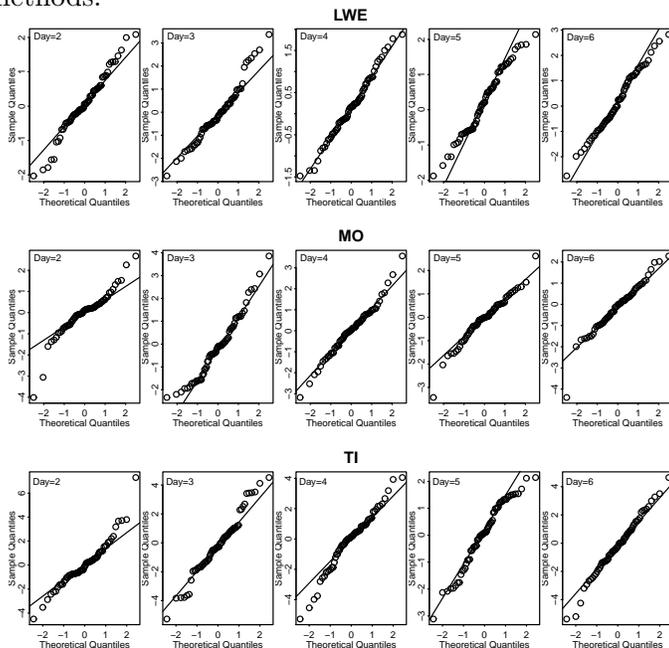


Table 1: Root mean square prediction error (RMSPE), coverage and relative mean width of the 95% probability intervals and relative mean squared error (MSE) of the three different methods.

Method	RMSPE	Coverage	Relative Mean Width ( $10^{-3}$ )	Relative MSE ( $10^{-6}$ )
LWE	1.02	0.950	12.9	17.3
MO	1.12	0.925	6.81	3.89
TI	1.81	0.766	5.19	6.92

width of these intervals (relative to the true output) and the relative mean squared error of the posterior predictive mean and the true output, relative to the true output.

From Table 3.3 it is clear that the multivariate emulators (MO and LWE) perform well in this application. The RMSPE is near one and the coverage of the predictive probability intervals is close to the nominal value of 0.95. The mean width of these intervals is largest for the LWE indicating higher predictive variance under this model. Although the TI model has the lowest mean width, the RMSPE is furthest from one and there is under-coverage of the predictive probability intervals.

### 3.4. Sensitivity Analysis

In this section we determine the sensitivity of the number of civilian deaths to changes in the input variables using the LWE as a surrogate for the computer model. We follow the probabilistic sensitivity analysis outlined by Oakley and O’Hagan (2004) and consider each dimension of output separately. Since we are using the LWE, some of the intractable integrals

in Oakley and O'Hagan (2004) are evaluated as zero, which simplifies the necessary calculations.

Let  $G$  denote the distribution of the input variables and define  $\mathbf{x}_s$  to be a vector of values of a subset of the input variables where  $\mathbf{x}_{-s}$  denotes the input variables not in  $\mathbf{x}_s$ . The posterior mean of the effect of  $\mathbf{x}_s$  is given by the  $k \times 1$  vector

$$\begin{aligned} \mathbf{z}_s(\mathbf{x}_s) &= \mathbb{E}(\mathbb{E}(\mathbf{f}(\mathbf{x})|\mathbf{x}_s) | \mathbf{Y}) - \mathbb{E}(\mathbb{E}(\mathbf{f}(\mathbf{x})) | \mathbf{Y}), \\ &= \hat{\mathbf{M}}^T (\mathbf{R}_s(\mathbf{x}_s) - \mathbf{R}), \end{aligned}$$

where

$$\mathbf{R}_s(\mathbf{x}_s) = \int_{\mathcal{X}_{-s}} h(\mathbf{x}) dG_{-s|s}(\mathbf{x}_{-s}|\mathbf{x}_s), \quad (13)$$

$$\mathbf{R} = \int_{\mathcal{X}} h(\mathbf{x}) dG(\mathbf{x}), \quad (14)$$

and  $G_{-s|s}(\cdot|\mathbf{x}_s)$  is the conditional distribution of  $\mathbf{x}_{-s}$  given  $\mathbf{x}_s$ .

Since  $h(\cdot)$  defines a 2nd order polynomial in the input variables, the  $j$ th element of  $h(\mathbf{x})$  can be written

$$h(\mathbf{x})_j = \prod_{i=1}^7 x_i^{a_{ij}},$$

for  $j = 1, \dots, m$ , where  $\sum_{i=1}^7 a_{ij} \leq 2$  and  $a_{ij} \in \{0, 1, 2\}$ . Therefore the  $j$ th elements of  $\mathbf{R}_s(\mathbf{x}_s)$  and  $\mathbf{R}$  are

$$\mathbf{R}_s(\mathbf{x}_s)_j = \int_{\mathcal{X}_{-s}} \prod_{i=1}^7 x_i^{a_{ij}} dG_{-s|s}(\mathbf{x}_{-s}|\mathbf{x}_s), \quad (15)$$

$$\mathbf{R}_j = \int_{\mathcal{X}} \prod_{i=1}^7 x_i^{a_{ij}} dG(\mathbf{x}), \quad (16)$$

respectively.

A special case is when  $\mathbf{x}_s = x_l$ , i.e. the  $l$ th input variable, for  $l = 1, \dots, p$ , so that  $\mathbf{z}_l(x_l)$  is the posterior mean of the ‘‘main effect’’ of the  $l$ th input variable.

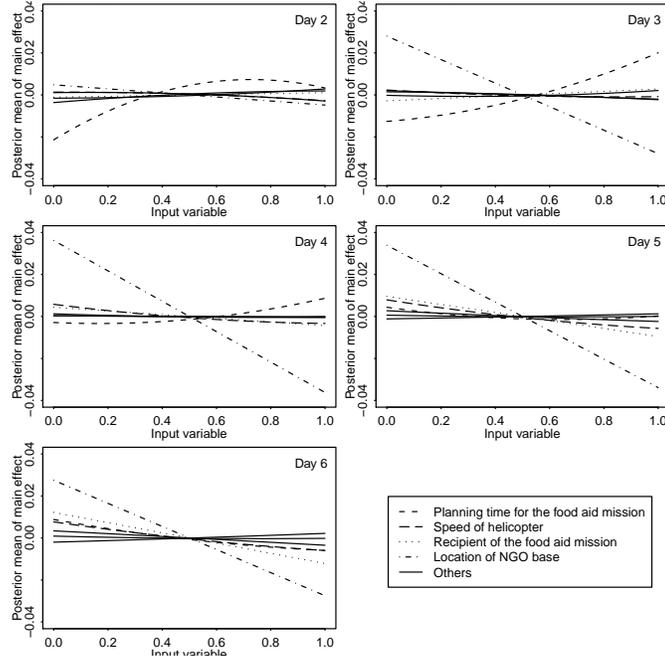
We specify a  $G$  such that  $x_i \stackrel{\text{iid}}{\sim} U[0, 1]$ , for  $i = 1, \dots, 5$ , and  $x_i \stackrel{\text{iid}}{\sim} \text{Bernoulli}(0.5)$ , for  $i = 6, 7$ . Therefore (15) and (16) simplify to

$$\begin{aligned} \mathbf{R}_s(\mathbf{x}_s)_j &= \prod_{i \in S} x_i^{a_{ij}} / \prod_{i \notin S} (a_{ij} + 1), \\ \mathbf{R}_j &= 1 / \prod_{i=1}^7 (a_{ij} + 1), \end{aligned}$$

respectively, where  $S$  denotes the indices of the elements of  $\mathbf{x}_s$ .

Figure 2 shows the posterior mean of the main effect of each of the input variables for each of the time points. We have only uniquely identified the planning time for the food aid

Figure 2: Posterior mean of the main effects for each of the input variables and each time point, plotted against the input variable. We have only uniquely identified planning time for the food aid mission, speed of the helicopter, and the two categorical input variables.



mission, the speed of the helicopter and the two categorical input variables as the output is most sensitive to these four input variables. Figure 3 shows  $\mathbf{z}_l(x_l, x_6, x_7)$  plotted against  $x_l$  for  $l = 1, \dots, p$ , for each time point and for each unique combination of the categorical input variables,  $x_6$  and  $x_7$ . We can see from Figure 3 that there appears to be interaction between the planning time for the food aid mission,  $x_2$ , and the categorical input variables.

For the LWE, the most important input variables can also be determined using Bayesian model selection methods (see, for example, O’Hagan and Forster, 2004, Chapter 7). Each choice of the function  $h(\cdot)$ , say  $h_v(\cdot)$ , defines a model,  $v \in \mathcal{V}$ . The posterior model probability of this model is given by

$$\pi(v|\mathbf{Y}) \propto \pi(\mathbf{Y}|v)\pi(v),$$

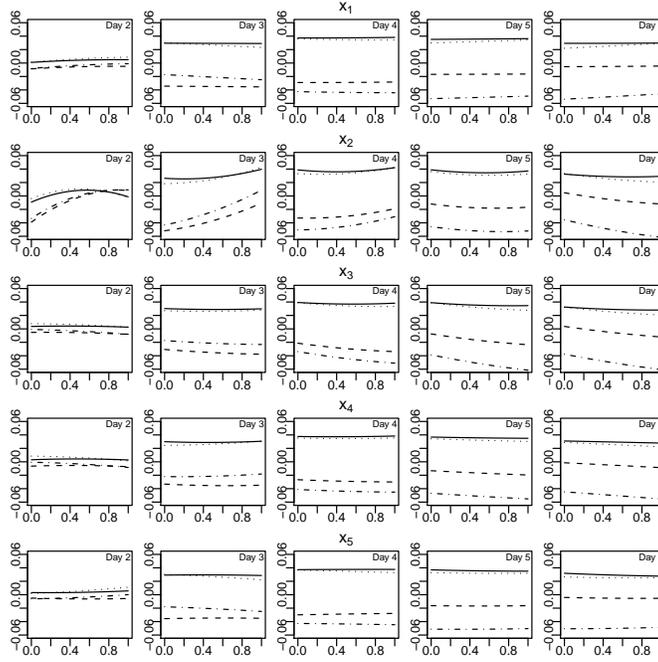
where  $\pi(\mathbf{Y}|v)$  is the marginal likelihood and  $\pi(v)$  is the prior model probability. In the case of a LWE, the marginal likelihood is available in closed form as

$$\pi(\mathbf{Y}|v) = \frac{\Gamma\left(\frac{\delta_v+n+k-1}{2}\right) |\hat{\boldsymbol{\Omega}}_v|^{\frac{k}{2}} |\mathbf{S}_v|^{\frac{\delta_v+k-1}{2}}}{\pi^{\frac{nk}{2}} \Gamma\left(\frac{\delta_v+k-1}{2}\right) |\boldsymbol{\Omega}_v|^{\frac{k}{2}} |\hat{\mathbf{S}}_v|^{\frac{\delta_v+n+k-1}{2}}},$$

where  $\delta_v$ ,  $\hat{\boldsymbol{\Omega}}_v$ ,  $\boldsymbol{\Omega}_v$ ,  $\hat{\mathbf{S}}_v$ , and  $\mathbf{S}_v$  are the prior and updated posterior hyperparameters associated with model  $v$ , as defined in Section 3.2.

We assume that the function  $h(\cdot)$  used in Section 3.3 defines the most complicated model that we are prepared to consider. The marginality principle (McCullagh and Nelder, 1989, ch. 3) is followed so that for an interaction to be included, both linear effects must also be present in the model. Under this principle, there are approximately 72 million possible models and so even though the marginal likelihoods are available in closed form, calculating each one is

Figure 3:  $\mathbf{z}_l(\mathbf{x}_l)$  plotted against  $x_s$  when  $\mathbf{x} = (x_s, x_6, x_7)$  for each input variable,  $x_s$ , and each time point, for each unique value of the categorical input variables  $x_6$  and  $x_7$ .



infeasible. Instead we use Markov Chain Monte Carlo Model Composition (MC<sup>3</sup>, Madigan and York, 1995). This produces a dependent sample which, if large enough, can be viewed as a sample from  $v|\mathbf{Y}$ .

Suppose the chain is in model  $v$  and a move to model  $v' \in \mathcal{V}$  is proposed, where  $v'$  either corresponds to retaining the same  $h(\cdot)$  (i.e.  $v' = v$ ), adding a term, or removing a term, where term refers to a linear effect of a variable or an interaction between two variables. This move is then accepted with probability  $\rho = \min(1, \alpha)$  where

$$\alpha = \frac{\pi(v'|\mathbf{Y})}{\pi(v|\mathbf{Y})} = \frac{\pi(\mathbf{Y}|v')\pi(v')r_v}{\pi(\mathbf{Y}|v)\pi(v)r_{v'}},$$

where  $r_v$  gives the number of models we can propose a move to, given that the current model is  $v$ . If we specify that  $\mathbf{S}_v = \mathbf{S}_{v'} = \mathbf{S}$  and  $\delta_v = \delta_{v'} = \delta$ , for all  $v, v' \in \mathcal{V}$ , then

$$\alpha = \frac{|\hat{\boldsymbol{\Omega}}_{v'}|^{\frac{k}{2}} |\boldsymbol{\Omega}_v|^{\frac{k}{2}} |\hat{\mathbf{S}}|_v^{\frac{\delta+n+k-1}{2}} \pi(v')r_v}{|\boldsymbol{\Omega}_{v'}|^{\frac{k}{2}} |\hat{\boldsymbol{\Omega}}_v|^{\frac{k}{2}} |\hat{\mathbf{S}}|_{v'}^{\frac{\delta+n+k-1}{2}} \pi(v)r_{v'}}.$$

We set  $\mathbf{S} = \mathbf{0}$  and  $\delta = -2k$ , i.e. the same non-informative values we used in Section 3.3. We also set  $\mathbf{M}_v = \mathbf{0}$ . All that remains is to specify the value for  $\boldsymbol{\Omega}_v$  which controls the prior variance of  $\mathbf{B}_v|\boldsymbol{\Sigma}_v$ . We use the convenient g-prior of Zellner (1983),

$$\boldsymbol{\Omega}_v^{-1} = g_v \mathbf{H}_v^T \mathbf{H}_v,$$

where  $g_v > 0$ . The value of  $g_v$  now controls the prior variance of  $\mathbf{B}_v|\boldsymbol{\Sigma}_v$ . It is known that posterior model probabilities are sensitive to the prior variance of the model parameters not common to all models. In particular, as  $g_v \rightarrow 0$ , by Lindley's paradox (see, for example,

Table 2: Marginal inclusion probabilities for the possible terms in  $h(\cdot)$  under the LWE.

Term	Marginal inclusion probability
Planning time for food aid mission	1.000
Speed of helicopter	0.445
Planning time for food aid mission (quadratic)	0.998
Recipient of food aid mission	1.000
Recipient of food aid mission $\times$ Planning time for food aid mission	0.650
Location of NGO base	1.000
Location of NGO base $\times$ Planning time for food aid mission	1.000
Recipient of food aid mission $\times$ Location of NGO base	1.000

O’Hagan and Forster, 2004, pp 77-78), the posterior model probability of the simplest model, i.e.  $h(\mathbf{x}) = 1$ , will tend to one.

We set  $g_j = g = n^{-1}$  which results in the unit-information prior proposed by Kass and Wasserman (1995) for linear models. It amounts to the prior distribution for  $\mathbf{B}_v | \Sigma_v$  providing the same amount of information as one run of the computer model. Non-informative prior model probabilities are specified as  $\pi(v) \propto 1$ .

We run the MC<sup>3</sup> algorithm for  $2.5 \times 10^4$  iterations, after a burn-in phase of  $10^4$  iterations and starting at the most complicated model. The marginal inclusion probability of a term is defined to be the posterior probability of including that particular term in  $h(\cdot)$ , estimated using the MC<sup>3</sup> sample from  $v | \mathbf{Y}$  as the proportion of models in the chain that include that term. Table 3.4 shows marginal inclusion probabilities greater than 0.05 for the 33 possible terms. These probabilities support the conclusions from the traditional sensitivity analysis above, i.e. the input is sensitive to the planning time for the food aid mission, the speed of the helicopter and the two categorical input variables.

### 3.5. Discussion of example 1

From Section 3.3, it appears that the MO method outperforms the TI method in terms of predictive ability in this application. Conti and O’Hagan (2010) came to the same conclusion when they compared the two methods on an example where  $k = 60$  and there were ten continuous input variables.

In addition to the Gaussian process models, we applied the LWE. Although this model was inferior to the MO model in terms of predictive ability, it can be fitted in fractions of a second as we do not need to find a value to use for  $\hat{\mathbf{r}}$ .

In Section 3.4 we use the LWE as a surrogate for the computer model. In the traditional probabilistic sensitivity analysis, many of the integrals in Oakley and O’Hagan (2004) are zero, simplifying the approach significantly. We also used Bayesian model determination to identify the most important input variables. Both approaches concluded that the output was most

Table 3: Input variables for the dispersion model.

Continuous input variables		Categorical input variables	
Name	Range	Name	Levels
Cloud cover, $x_1$	[0, 1]	Material-Munition, $x_5$	$\{\alpha_D, \lambda_D, \mu_C, \mu_D, \mu_{2C}\}$
Temperature, $x_2$	[280, 310]		
Wind speed, $x_3$	[0, 16]		
Release mass, $x_4$	[10, 5000]		
$x$ -release location, $x_0$	[1, 128]		
$y$ -release location, $y_0$	[1, 128]		
Wind direction, $\omega$	[0, $2\pi$ ]		

sensitive to the planning time for the food aid mission, the speed of the helicopter, and the two categorical input variables. There is also strong evidence of interactions between the planning time for the food aid mission and the categorical inputs.

With the LWE a further approach would be to undertake the Bayesian model determination to obtain the posterior model probabilities. We could then use Bayesian model averaging to find the posterior predictive distribution, i.e. the emulator. The emulator would then be a mixture of multivariate t-distributions where the weights are given by the posterior model probabilities. However we would then lose the “light weightness” of the LWE over the Gaussian process.

#### 4. A dispersion model: a computer model with multivariate zero-inflated output

##### 4.1. Introduction

The simulator considered in this section is called a dispersion model. It models the dispersion of particles, over a given terrain, after a chemical or biological particles release. Here, the terrain is represented as a  $128 \times 128$  grid.

Let  $\mathbf{Y}_i = \mathbf{f}(\mathbf{x}_i)$  denote the  $k \times 1$  vector of output of the computer model given the input variables  $\mathbf{x}_i$ , where  $k = 128^2 = 16384$ . Each element,  $Y_{ij}$ , of  $\mathbf{Y}_i$  for  $j = 1, \dots, k$ , gives the dosage observed at a particular location, indexed by  $j$ , on the grid. Table 4.1 shows the input variables for this computer model which are both continuous and categorical.

Let

$$\mathbf{Y} = \begin{pmatrix} \mathbf{Y}_1^T \\ \vdots \\ \mathbf{Y}_n^T \end{pmatrix},$$

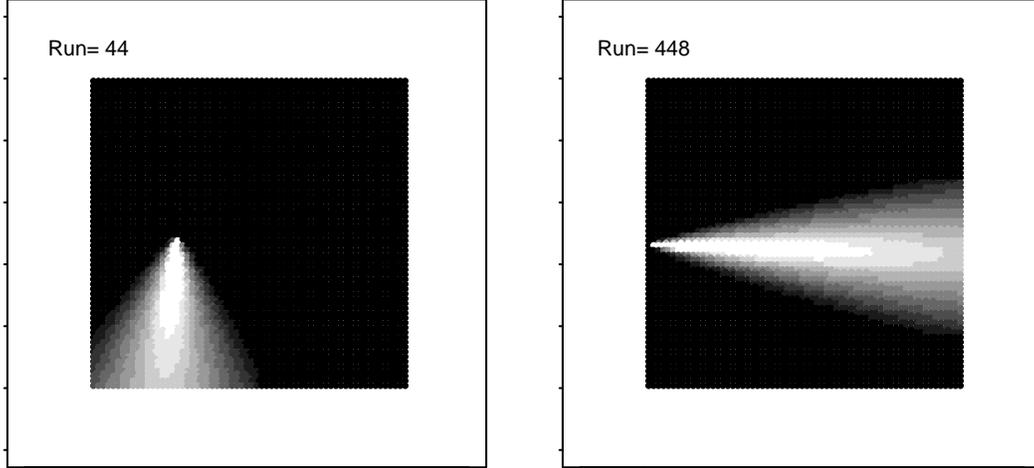
be the  $n \times k$  matrix of outputs.

There are two challenges to overcome when emulating this computer model. The first is the high dimensionality,  $k$ , of the output. The second is that the output is zero-inflated.

Regarding the high dimensionality, Woods and Lewis (2009) emulated this computer model using the approach of Higdon et al. (2008). Consider the singular value decomposition (SVD)

$$\mathbf{Y}^T = \mathbf{U}\mathbf{D}\mathbf{V}^T,$$

Figure 4: Plumes for two runs of the computer model.



where  $\mathbf{U}$  is a  $k \times n$  orthogonal matrix,  $\mathbf{D}$  is an  $n \times n$  diagonal matrix of the singular values and  $\mathbf{V}$  is an  $n \times n$  orthonormal matrix. A  $q$ -dimensional principal component basis,  $\mathbf{b}_1, \dots, \mathbf{b}_q$ , is given by the first  $q \leq n$  columns of  $n^{-\frac{1}{2}}\mathbf{U}\mathbf{D}$  with weights,  $w_j(\mathbf{x}_i)$ , given by the  $j$ th entry of  $n^{\frac{1}{2}}\mathbf{V}^T$ . We relate the  $q$   $n \times 1$  vectors,  $\mathbf{w}_1, \dots, \mathbf{w}_q$ , where  $\mathbf{w}_j = (w_j(\mathbf{x}_1), \dots, w_j(\mathbf{x}_n))$ , using  $q$  Gaussian process models. We then predict  $\mathbf{y}_0 = \mathbf{f}(\mathbf{x}_0)$  using

$$\mathbf{y}_0 = \sum_{j=1}^q \mathbf{b}_j \hat{w}_j(\mathbf{x}_0),$$

where  $\hat{w}_j(\mathbf{x}_0)$  is the prediction of the  $j$ th Gaussian process at  $\mathbf{x}_0$ . For more details on this approach see Woods and Lewis (2009). However, in this section we instead use a lightweight emulator.

## 4.2. Methodology

**4.2.1. Emulator** We have output from 474 runs of the simulator with which to construct and test the emulator. These runs are historical data rather than a designed experiment.

Figure 4 shows the output from two runs, 44 and 448, of the computer model. Black signifies that no dosage was observed at that location and white signifies that a high dosage was observed. The shapes in Figure 4 are called plumes.

It is not feasible to use the LWE described in Section 3.2 where  $k = 16384$ . Also this method is not appropriate for zero-inflated output.

To reduce the dimensionality we consider the output to be scalar, with the location where the dosage is observed as extra input variables. This is a generalisation of the time-input Gaussian process, considered in Section 3.2, to spatial inputs. Associated with each  $Y_{ij}$  is the location variables,  $(\mathcal{X}_{ij}, \mathcal{Y}_{ij})$ , where  $Y_{ij}$  was observed, with  $1 \leq \mathcal{X}_{ij}, \mathcal{Y}_{ij} \leq 128$ . Note that  $\mathcal{X}_{ij} = \mathcal{X}_{lj}$  and  $\mathcal{Y}_{ij} = \mathcal{Y}_{lj}$ , for all  $l, i = 1, \dots, N$  and  $j = 1, \dots, k$ .

We transform the location variables for the  $i$ th run, equivalent to rotating and shifting the grid so that the release is always at the origin and the wind direction is always  $\pi$ . We also convert the location variables to polar co-ordinates,  $r_{ij}$  and  $\theta_{ij}$ , i.e.

$$r_{ij}^2 = (\mathcal{X}_{ij} - x_{0i})^2 + (\mathcal{Y}_{ij} - y_{0i})^2, \quad (17)$$

$$\theta_{ij} = \begin{cases} \arctan\left(\frac{\mathcal{Y}_{ij} - y_{0i}}{\mathcal{X}_{ij} - x_{0i}}\right) + \pi - \omega & \text{if } \omega \geq 0, \\ \arctan\left(\frac{\mathcal{Y}_{ij} - y_{0i}}{\mathcal{X}_{ij} - x_{0i}}\right) - \pi - \omega & \text{if } \omega < 0. \end{cases} \quad (18)$$

Now associated with each  $Y_{ij}$  is the  $p \times 1$  vector of input variables  $\mathbf{x}_{ij}^* = (x_{1i}, x_{2i}, x_{3i}, x_{4i}, x_{5i}, r_{ij}, \theta_{ij})$  where  $p = 7$ . Note that wind direction and release location are now constant for all  $(i, j)$  so no longer feature in the vector of input variables. Studying dispersion where the release location and the wind direction are constant follows from Clarke (1979), albeit in Cartesian co-ordinates as opposed to polar.

We use the odd-numbered runs to build the emulator and the even-numbered runs to test. Therefore  $n = 237$ . The vector of output now has  $nk \approx 3.9$  million elements. This will be too large for sensible computation. We make the assumption that if the location is upwind of the release, then the observed dosage will be zero. Therefore we remove all observations such that  $\theta_{ij} \notin [-\frac{\pi}{2}, \frac{\pi}{2}]$ . This leaves approximately 2.7 million observations. We subsample, without replacement,  $n_i$  observations within each run of the computer model so that we have  $n^* = \sum_{i=1}^n n_i$  observations, labelled  $\mathbf{Y}^*$ .

We therefore have  $n^*$  output values which are zero-inflated. A standard statistical model for analysing zero-inflated continuous responses is the Tobit regression model (Tobin, 1958). Chib (1992) considered Bayesian inference for the Tobit model deriving a method to obtain a sample from the posterior distribution of the model parameters using Gibbs sampling.

We assume that the latent variables,  $Z_i$ , have independent normal distributions

$$Z_i \sim N(h(\mathbf{x}_i)^T \boldsymbol{\beta}, \sigma^2),$$

for  $i = 1, \dots, n^*$  and we observe  $Y_i^* = \max(Z_i, 0)$ . We specify the following conditionally conjugate prior distributions:

$$\begin{aligned} \boldsymbol{\beta} | \sigma^2 &\sim N(\mathbf{m}, \sigma^2 \boldsymbol{\Omega}), \\ \sigma^2 &\sim \text{IG}\left(\frac{\delta}{2}, \frac{s}{2}\right). \end{aligned}$$

Let  $C = \{i : Y_i^* = 0, i = 1, \dots, n^*\}$  identify the output values that are identically zero. The Gibbs sampling scheme for the Tobit model is:

1. Let the current model parameters be  $\boldsymbol{\beta}^{(j)}$  and  $\sigma^{2(j)}$ .
2. For  $i \in C$ , generate a latent variable  $Z_i^{(j+1)}$  from  $N(h(\mathbf{x}_i^*)^T \boldsymbol{\beta}^{(j)}, \sigma^{2(j)})$ , truncated to  $(-\infty, 0)$ . Let

$$R_i = \begin{cases} Y_i^* & \text{if } i \notin C, \\ Z_i^{(j+1)} & \text{if } i \in C, \end{cases}$$

and  $\mathbf{R} = (R_1, \dots, R_{n^*})^T$ .

3. Let

$$\begin{aligned}\hat{\Omega} &= (\mathbf{H}^T \mathbf{H} + \Omega^{-1})^{-1}, \\ \hat{\mathbf{m}} &= \hat{\Omega} (\mathbf{H}^T \mathbf{R} + \Omega^{-1} \mathbf{m}), \\ \hat{\delta} &= \delta + n^*, \\ \hat{s} &= s + \mathbf{R}^T \mathbf{R} + \mathbf{m}^T \Omega^{-1} \mathbf{m} - \hat{\mathbf{m}}^T \hat{\Omega}^{-1} \hat{\mathbf{m}}.\end{aligned}$$

4. Generate  $\sigma^{2(j+1)}$  from  $\text{IG}\left(\frac{\hat{\delta}}{2}, \frac{\hat{s}}{2}\right)$ .

5. Generate  $\beta^{(j+1)}$  from  $\text{N}\left(\hat{\mathbf{m}}, \sigma^{2(j+1)} \hat{\Omega}\right)$ .

Although the posterior distributions are not available in closed form, the above algorithm is still computationally inexpensive since the full conditional distributions are known, so therefore we do not need to use any rejection sampling methods.

We now have a sample,

$$\left\{ \left( \beta^{(1)}, \sigma^{2(1)} \right), \dots, \left( \beta^{(B)}, \sigma^{2(B)} \right) \right\},$$

from the posterior distribution of  $\beta$  and  $\sigma^2$ . To evaluate the posterior predictive distribution,  $\mathbf{y}_0 | \mathbf{Y}^*$ , where  $\mathbf{y}_0 = \mathbf{f}(\mathbf{x}_0)$ , let  $y_{0j}$  be the  $j$ th element of  $\mathbf{y}_0$ , for  $j = 1, \dots, k$ . We can find a posterior sample from  $y_{0j} | \mathbf{Y}^*$  by generating

$$z_{0j}^{(t)} \sim \text{N}\left(h(\mathbf{x}_{0j}^*)^T \beta^{(t)}, \sigma^{2(t)}\right),$$

where  $\mathbf{x}_{0j}^* = (x_{10}, x_{20}, x_{30}, x_{40}, x_{50}, r_{0j}, \theta_{0j})$ , and then setting  $y_{0j}^{(t)} = \max(0, z_{0j}^{(t)})$ , for  $t = 1, \dots, B$ . This is the fully Bayesian approach. The alternative plug-in approach is less computationally expensive and is the approach taken in this paper. Let  $\hat{\beta}$  and  $\hat{\sigma}^2$  be representative values of  $\beta$  and  $\sigma^2$ , respectively, relative to  $\beta, \sigma^2 | \mathbf{Y}^*$ . The plug-in approach is that  $y_{0j} | \mathbf{Y}^*$  is the censored normal distribution, formed by taking  $y_{0j} = \max(0, z_{0j})$ , with

$$z_{0j} \sim \text{N}\left(h(\mathbf{x}_{0j}^*)^T \hat{\beta}, \hat{\sigma}^2\right).$$

### 4.3. Results

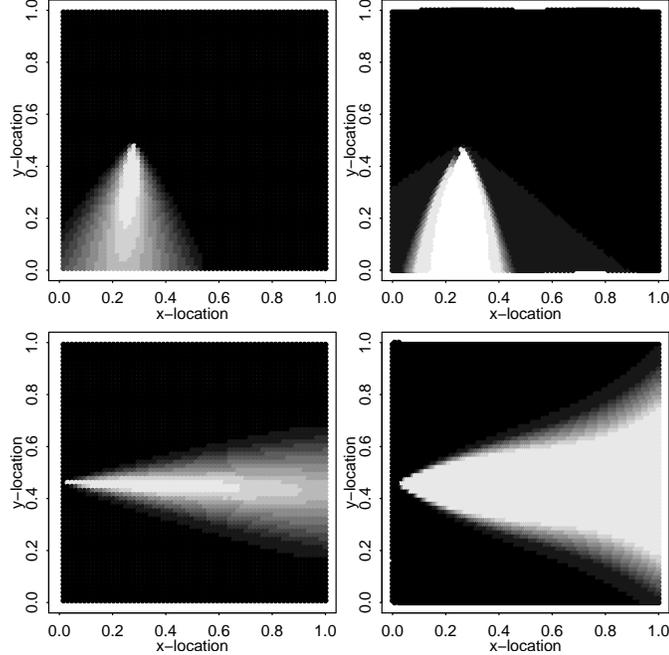
Let  $n_i = 50$ , for  $i = 1, \dots, 237$ , resulting in  $n^* = 11850$ . Only 30% of the sampled  $Y_i^*$ s are non zero, indicating the high level of zero-inflation.

After taking logarithms of wind speed,  $x_3$  and release mass,  $x_4$ , we specify an  $h(\cdot)$  that corresponds to an intercept, all linear effects, all two-way interactions and quadratic terms for the continuous input variables; thus  $m = 56$ . We use non-informative choices for the hyperparameters of the prior of  $\beta$  and  $\sigma^2$ , i.e.  $\mathbf{m} = \mathbf{0}$ ,  $\Omega^{-1} = \mathbf{0}$ ,  $s = 0$  and  $\delta = -2$ .

A sample of size  $B = 10^4$  is generated from  $\beta, \sigma^2 | \mathbf{Y}^*$  using the Gibbs sampling scheme, after a burn-in phase of  $5 \times 10^3$  iterations. We set  $\hat{\beta}$  and  $\hat{\sigma}^2$  to be the sample means of  $\beta$  and  $\sigma^2$ , respectively.

We use the test runs of the computer model to evaluate the emulator. We find the posterior mean of each location of each test run and also the 95% posterior predictive probability interval

Figure 5: Predicted plumes for two runs of the computer model.



by taking the 2.5th and 97.5th quantiles of the posterior predictive distribution as the lower and upper limits, respectively. Explicitly the  $r$ th quantile is given by

$$y_{0j}^{(r)} = \begin{cases} 0 & \text{if } r < \Phi\left(-\frac{h(\mathbf{x}_{0j}^*)^T \hat{\boldsymbol{\beta}}}{\hat{\sigma}}\right), \\ h(\mathbf{x}_{0j}^*)^T \hat{\boldsymbol{\beta}} + \Phi^{-1}(r) \hat{\sigma} & \text{otherwise.} \end{cases}$$

where  $\mathbf{x}_{0j}^* = (x_{10}, x_{20}, x_{30}, x_{40}, x_{50}, r_{0j}, \theta_{0j})$ .

The coverage of the intervals is 0.998 with a mean width of  $2.07 \times 10^{-4}$ . The MSE between the predictive mean and the true dosage is  $3.84 \times 10^{-3}$ . The value of MSE is of the same order as the MSE obtained by Woods and Lewis (2009) for the same computer model but using a Gaussian process as the emulator as described in Section 4.1.

Figure 5 shows the true plumes for runs 44 and 448 (featured in Figure 4) and the corresponding plume where we have used the posterior predictive mean as the point prediction.

#### 4.4. Sensitivity Analysis

In this section we conduct a sensitivity analysis to identify the most important input variables. This sensitivity analysis is undertaken under a fixed wind direction and release location. To incorporate an unknown wind direction or release location we would just transform the locations,  $r$  and  $\theta$ , using the inverse of the transformation given by (17) and (18).

The mean of the posterior predictive distribution of the dosage at the location given by  $r$  and  $\theta$ , given the input variables  $x_1, \dots, x_5$  is

$$\mu(x_1, \dots, x_5; r, \theta) = \hat{\eta} \left(1 - \Phi\left(-\frac{\hat{\eta}}{\hat{\sigma}}\right)\right) + \hat{\sigma} \phi\left(-\frac{\hat{\eta}}{\hat{\sigma}}\right), \quad (19)$$

where  $\hat{\eta} = h(\mathbf{x}^*)^T \hat{\boldsymbol{\beta}}$  and  $\mathbf{x}^* = (x_1, \dots, x_5, r, \theta)$ .

The posterior mean of the main effect of the  $l$ th input variable,  $x_l$ , at the location given by  $r$  and  $\theta$ , is

$$z_l(x_l; r, \theta) = R_l(x_l; r, \theta) - R(r, \theta),$$

where

$$R_l(x_l; r, \theta) = \int_{\mathcal{X}_{-l}} \mu(x_1, \dots, x_5; r, \theta) dG_{-l|l}(\mathbf{x}_{-l}|x_l), \quad (20)$$

$$R(r, \theta) = \int_{\mathcal{X}} \mu(x_1, \dots, x_5; r, \theta) dG(\mathbf{x}), \quad (21)$$

for  $l = 1, \dots, 5$ , where  $G$  denotes the joint distribution of the input variables  $x_1, \dots, x_5$ . The integrals involved in (20) and (21) will be analytically intractable for a typical  $G$  so we use Monte Carlo methods to approximate these quantities. To approximate  $R_l(x_l; r, \theta)$ , we generate  $\mathbf{x}_{-l}^{(i)}$  from  $G_{-l|l}$  and evaluate  $\mu(x_1, \dots, x_5; r, \theta)^{(i)}$  using (19), for  $i = 1, \dots, B$ . We then approximate  $R_l(x_l; r, \theta)$  as the mean of the  $\mu(x_1, \dots, x_5; r, \theta)^{(i)}$ s. It is straightforward to extend this methodology to approximate  $R(r, \theta)$ .

For  $G$ , we let  $x_l \stackrel{\text{iid}}{\sim} U[0, 1]$ , for  $l = 1, \dots, 4$  and let  $x_5$  be discretely uniform on the set  $\{\alpha_D, \lambda_D, \mu_C, \mu_D, \mu_{2C}\}$ .

We investigate the sensitivity of the dosage to changes in the input variables at six different locations, also shown in Figure 6:

1.  $(r, \theta) = (0.10, 0)$ ,
2.  $(r, \theta) = (0.25, 0)$ ,
3.  $(r, \theta) = (0.50, 0)$ ,
4.  $(r, \theta) = (0.10, \pi/8)$ ,
5.  $(r, \theta) = (0.25, \pi/8)$ ,
6.  $(r, \theta) = (0.50, \pi/8)$ .

The first three locations lie on the up-downwind axis in the downwind direction at three different distances from the release. The second three locations are further away from the up-downwind axis.

Figure 7 shows  $z_l(x_l; r, \theta)$  plotted against  $x_l$ , for the six locations shown in Figure 6 and for  $l = 1, 2, 3, 4$ , i.e. the continuous input variables. Also shown in Figure 7, as grey horizontal lines, are the values of  $z_5(x_5; r, \theta)$  for the five different levels of the material-munition input variable.

The predicted dosage is most sensitive to changes in release mass and wind speed when the location is close to the release location, i.e. when  $r = 0.1$ . As  $r$  increases, i.e. the location becomes further from the release location, the predicted dosage becomes less sensitive to changes in all of the input variables. This indicates that there are interactions between the input variables and  $r$ .

When  $r = 0.1$ , increasing the release mass increases the predicted dosage, whilst increasing the wind speed decreases the predicted dosage. This is in agreement with the simple Gaussian plume model of Clarke (1979).

Figure 6: The six locations where we investigate the dosage sensitivity to the input variables.

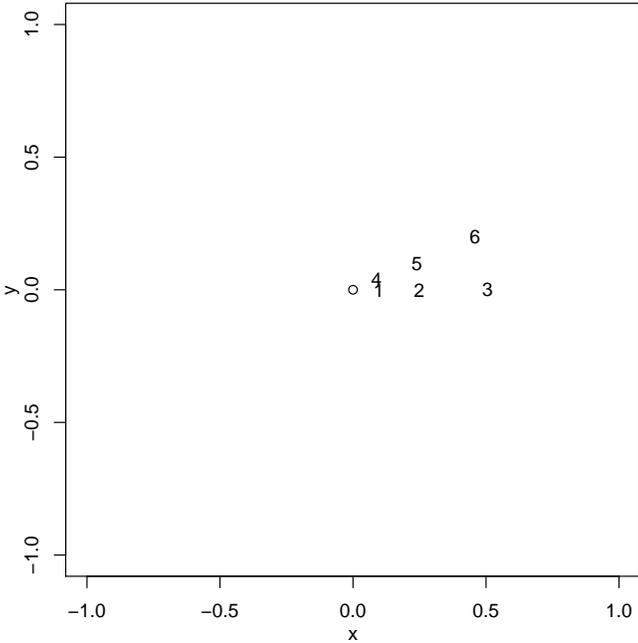
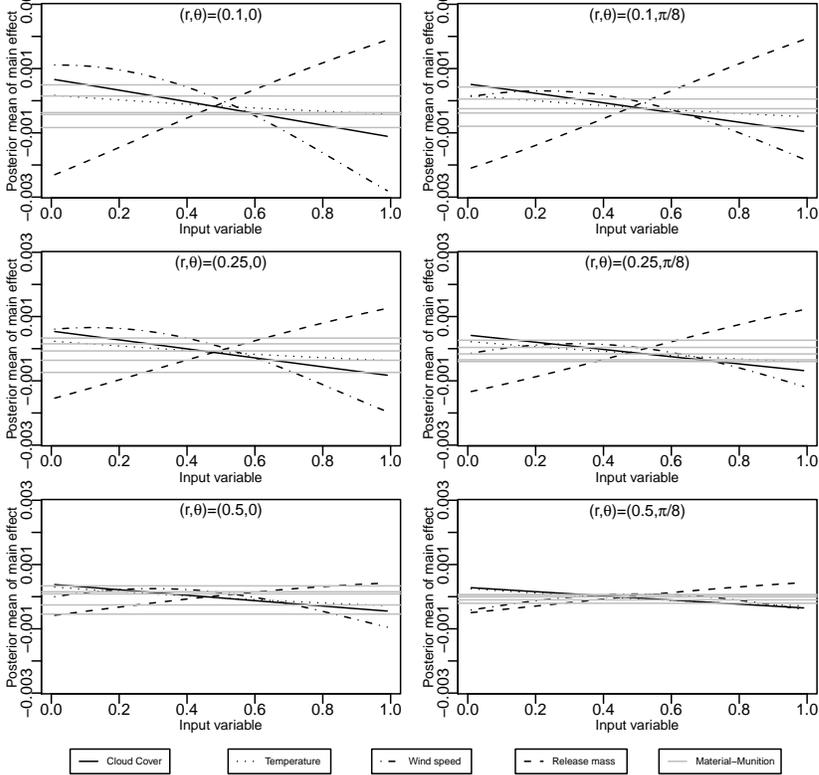


Figure 7: Plot of  $z_l(x_l; \mathcal{X}^{\text{new}}, y^{\text{new}})$  against  $x_l$  at the six locations shown in Figure 6.



We can also determine the most important input variables using Bayesian model selection. The marginal likelihoods, and therefore the posterior model probabilities, of Tobit models are not available analytically. Gawande (1998) proposed a Monte Carlo method for approximation of the marginal likelihood using the marginal likelihood identity method of Chib (1995). However there are approximately 143 million possible models rendering this approach infeasible. Instead we introduce a model update step into the Gibbs sampling scheme in Section 4.2.

Let each possible Tobit model be indexed by  $v \in \mathcal{V}$  where  $V$  is the set of possible models. The model selection Gibbs sampling scheme is:

1. Let the current model parameters be  $\boldsymbol{\beta}_{v^{(j)}}^{(j)}$ ,  $\sigma_{v^{(j)}}^{2(j)}$  and  $v^{(j)}$ .
2. For  $i \in C$ , generate a latent variable  $Z_i^{(j+1)}$  from  $N\left(h_{v^{(j)}}(\mathbf{x}_i^*)^T \boldsymbol{\beta}_{v^{(j)}}^{(j)}, \sigma_{v^{(j)}}^{2(j)}\right)$  truncated to  $(-\infty, 0)$ . Let

$$R_i = \begin{cases} Y_i^* & \text{if } i \notin C, \\ Z_i^{(j+1)} & \text{if } i \in C, \end{cases}$$

and  $\mathbf{R} = (R_1, \dots, R_{n^*})^T$ .

3. Propose a new model  $v'$  and accept this move with probability  $\min(1, a)$  where

$$a = \frac{|\hat{\boldsymbol{\Omega}}_{v'}|^{\frac{1}{2}} |\boldsymbol{\Omega}_{v^{(j)}}|^{\frac{1}{2}} \hat{s}_{v^{(j)}} \pi(v') r_{v^{(j)}}}{|\boldsymbol{\Omega}_{v'}|^{\frac{1}{2}} |\hat{\boldsymbol{\Omega}}_{v^{(j)}}|^{\frac{1}{2}} \hat{s}_{v'} \pi(v^{(j)}) r_{v'}}.$$

If the move is accepted, then  $v^{(j+1)} = v'$ , otherwise  $v^{(j+1)} = v^{(j)}$ .

4. Generate  $\sigma_{v^{(j+1)}}^{2(j+1)}$  from  $\text{IG}\left(\frac{\hat{\delta}_{v^{(j+1)}}}{2}, \frac{\hat{s}_{v^{(j+1)}}}{2}\right)$ .
5. Generate  $\boldsymbol{\beta}_{v^{(j+1)}}^{(j+1)}$  from  $N\left(\hat{\mathbf{v}}_{v^{(j+1)}}, \sigma_{v^{(j+1)}}^{2(j+1)} \hat{\boldsymbol{\Omega}}_{v^{(j+1)}}\right)$ .

We are certain that  $r$  and  $\theta$  and their interaction are terms in  $h(\cdot)$ , so we set  $\pi(v) = 0$  for all models with an  $h_v(\cdot)$  that does not contain these three terms. The prior over the remaining models is uniform. Explicitly

$$\pi(v) \propto \begin{cases} 0 & \text{if } h_v(\cdot) \text{ does not contain } r, \theta \text{ or their interaction,} \\ 1 & \text{otherwise.} \end{cases}$$

For the prior distributions for the model parameters, we set  $\boldsymbol{\Omega}_v^{-1} = \frac{1}{n^*} \mathbf{H}_v^T \mathbf{H}_v$ ,  $s_v = 0$ ,  $\delta_v = 0$  and  $\mathbf{m}_v = \mathbf{0}$ .

We run the above model selection Gibbs sampling scheme for a total of  $2.5 \times 10^4$  iterations after a burn-in phase of  $10^4$  iterations. Table 4.4 shows the marginal inclusion probabilities greater than 0.1. The terms included in Table 4.4 agree with our conclusions from the traditional sensitivity analysis, i.e. that the predicted dosage is most sensitive to changes in wind speed and release mass, and that there exist interactions between the input variables and the distance from the release location,  $r$ .

Table 4: Marginal inclusion probabilities for possible terms in  $h(\cdot)$ .

Term	Marginal inclusion probability
Linear effects	
Wind speed	1.000
Release mass	1.000
$r$	1.000
$\theta$	1.000
Material-munition	1.000
Quadratic effects	
Wind speed	0.959
$r$	1.000
$\theta$	1.000
Interactions	
Wind speed $\times$ Release mass	0.195
$r \times$ Release mass	1.000
$r \times$ Wind speed	0.999
$\theta \times$ Wind speed	1.000
$r \times \theta$	1.000
Wind speed $\times$ Material-munition	1.000
Release mass $\times$ Material-munition	1.000
$r \times$ Material-munition	1.000

## 4.5. Discussion of example 2

In this section we used a generalisation of the LWE to zero-inflated output. The posterior distributions were not available in closed form but a computationally cheap Gibbs sampling algorithm was used to generate a sample from these distributions.

We carried out a sensitivity analysis, in the traditional sense and via Bayesian model selection, and found that the dosage was most sensitive to changes in the release mass and wind speed, and the relationships change as the location moves away from the release location.

Currently, the accuracy of the emulator is suitable for qualitative predictions, and as such is useful for identifying key trends and input variables with substantive impact and also informing broad policy decisions in, for example, military planning and emergency response.

## 5. Conclusions

In this article, we used lightweight emulation in two challenging computer model applications. The key feature of lightweight emulation is to model the features of the computer model through the mean function as opposed to the covariance structure, as in a Gaussian process.

In the first application, the humanitarian relief mission computer model, the lightweight emulator had predictive accuracy only an order of magnitude less than that of the standard Gaussian process, but allowed us to undertake an analytic sensitivity analysis and a Bayesian model comparison to identify the most important input variables.

In the second application, the dispersion computer model, the lightweight emulator did not have entirely satisfactory predictive accuracy but, nevertheless, a descriptive model was produced that allowed us to undertake a useful sensitivity analysis to identify important input variables.

The lightweight emulator has great scope for generalisation. For instance, suppose the output of the computer model was distinctly non-normal, then a class of models which could be used as a lightweight emulator are generalised linear models (GLMs).

## 6. Acknowledgments

This work was funded by a Defence Threat Reduction Agency basic research grant and the Defence Science and Technology Laboratory (Dstl). The authors wish to thank Dr Veronica Bowman and Robin Ashmore from Dstl for providing the simulators and data for the two examples, and related invaluable conversations.

## 7. References

1. Berger, J., De Oliveira, V., and Sanso, B. (2001). Objective Bayesian analysis of spatially correlated data. *Journal of the American Statistical Association*, **96**, 1361-1374.
2. Chib, S. (1992). Bayes inference in the Tobit censored regression model. *Journal of Econometrics*, **51**, 79-99.
3. Chib, S. (1995). Marginal Likelihood from the Gibbs Output. *Journal of the American Statistical Association*, **90**, 1313-1321.
4. Clarke, R.H. (1979). *The First Report of a Working Group on Atmospheric Dispersion:*

- a Model for Short and Medium Range Dispersion of Radionuclides Released to the Atmosphere.* London: Her Majesty's Stationary Office.
5. Conti, S. and O'Hagan, A. (2010). Bayesian Emulation of Complex Multi-Output and Dynamic Computer Models. *Journal of Statistical Planning and Inference*, **140**, 640-651.
  - Dawid, A.P. (1981). Some matrix-variate distribution theory: Notational considerations and a Bayesian application. *Biometrika*, **68**, 265-274.
  6. Fang, K., Li, R. and Sudjianto, A. (2006). *Design and Modeling for Computer Experiments.* Chapman and Hall.
  7. Gawande, K. (1998). Comparing Theories of Endogenous Protection: Bayesian Comparison of Tobit Models using Gibbs Sampling Output. *The Review of Economics and Statistics*, **80**, 128-140.
  8. Higdon, D., Gattiker, J., Williams, B. and Rightley, M. (2008). Computer Model Calibration Using High-Dimensional Output. *Journal of the American Statistical Association*, **103**, 570-583.
  9. Kennedy, M.C. and O'Hagan, A. (2001). Bayesian calibration of computer models (with discussion). *Journal of the Royal Statistical Society, B*, **63**, 425-464.
  10. Kotz, S. and Nadarajah, S. (2004). *Multivariate t Distributions and Their Applications.* Cambridge.
  11. Madigan, D. and York, J. (1995). Bayesian graphical models for discrete data. *International Statistics Review*, **63**, 215-232.
  12. McCullagh, P. and Nelder, J. (1989). *Generalised Linear Models* (2nd ed.). Chapman and Hall.
  13. McKay, M.D., Conover, W.J. and Beckman, R.J. (1979). A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, **21**, 239-245.
  14. Morris, M.D. and Mitchell, T.J. (1995). Exploratory Designs for Computer Experiments. *Journal of Statistical Planning and Inference*, **43**, 381-402.
  15. Oakley J.E. and O'Hagan, A. (2004). Probabilistic Sensitivity Analysis of Complex Models: A Bayesian Approach. *Journal of the Royal Statistical Society, B*, **66**, 751-769.
  16. O'Hagan, A. and Forster, J.J. (2004). *Kendall's Advanced Theory of Statistics, Volume 2B: Bayesian Inference.* (2nd edition). Arnold.
  17. Qian, P.Z., Wu, H. and Wu, C.F.J. (2008). Gaussian Process Models for Computer Experiments with Qualitative and Quantitative Factors. *Technometrics*, **50**, 383-396.
  18. Qian, P.Z. and Wu, C.F.J. (2009). Sliced space-filling designs. *Biometrika*, **96**, 945-956.
  19. Rougier, J. (2007). Lightweight emulators for Multivariate Deterministic Functions. *MUCM Technical Report 07/02*, University of Durham, UK.

20. Rougier, J. (2008). Efficient emulators for multivariate deterministic functions. *Journal of Computational and Graphical Statistics*, **17**, 827-843.
21. Sacks, J., Welch, W.J., Mitchell, T.J. and Wynn, H.P. (1989). Design and analysis of computer experiments. *Statistical Science*, **4**, 409-435.
22. Saltelli, A., Chan, K. and Scott, M. (2000). *Sensitivity Analysis*. Wiley.
23. Santner, T.J., Williams, B.J. and Notz, W.I. (2003). *The Design and Analysis of Computer Experiments*. Springer.
24. Tobin, J. (1958). Estimation of relationships for limited dependent variables. *Econometrica*, **26**, 24-36.
25. Woods, D.C. and Lewis, S.M. (2009). Surrogate-modelling for computer experiments with multi-dimensional responses. Technical Report, Southampton Statistical Sciences Research Institute, University of Southampton, United Kingdom.
26. Zellner, A. (1983). Applications of Bayesian Analysis in Econometrics. *Journal of the Royal Statistical Society, D*, **32**, 23-34.