Using Known Boundaries to Improve Bayesian Emulation

Rebekah Fearnhead

Durham University

February 21, 2024

Introduction

Introduction

- Complex Mathematical Models are being used in many scientific areas to help describe complex physical systems. For example:
  - Systems Biology
  - Predator-Prey Models
  - Galaxy Formation
  - Disease Modelling
- One problem with this is that it could take from a week to a month to do a single evaluation.
- This problem can be solved using an emulator. This mimics the computer model function that we want to evaluate, but will do it much faster than the full model for the complex system.

Emulation

Emulation in 2D

- To emulate in two dimensions, the simplest model that can be used is $f(x) = u(x)$, where $u(x)$ is a weakly stationary process.
- Using a Bayes Linear Emulator, only the prior beliefs of the second order structure need to be specified.
- Once runs, D, of the model have been performed, $f(x)$ needs to be updated, using:

$$\begin{aligned} \mathsf{E}_D[f(x)] &= \mathsf{E}[f(x)] + \mathsf{Cov}[f(x), D]\mathsf{Var}[D]^{-1}(D - \mathsf{E}[D]), \\ \mathsf{Var}_D[f(x)] &= \mathsf{Var}[f(x)] - \mathsf{Cov}[f(x), D]\mathsf{Var}[D]^{-1}\mathsf{Cov}[D, f(x)]. \end{aligned}$$

Example of Emulation

For example, using the function,

$$f(x) = -\sin(2\pi x_2) + 0.9\sin(2\pi(1 - x_1)(1 - x_2)),$$

and evaluating runs at 9 design points.

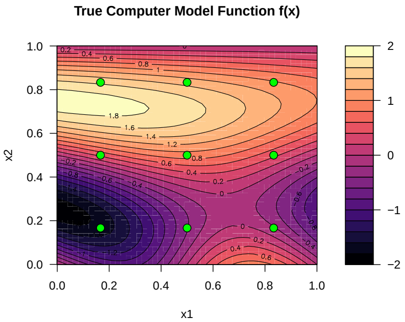

Figure: The True Value of the Example Function

The two dimensional emulator will produce the following results:

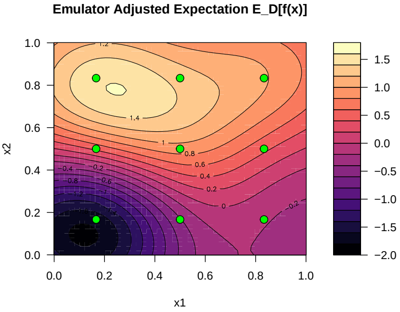

Figure: Expectation of the Emulator

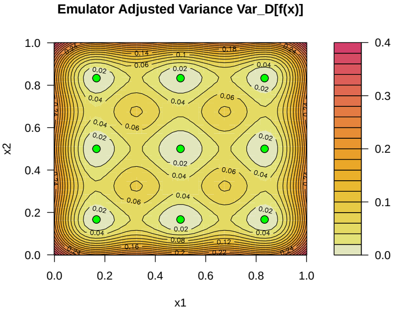

Figure: Variance of the Emulator

Comparing the emulator expectation to the true value:

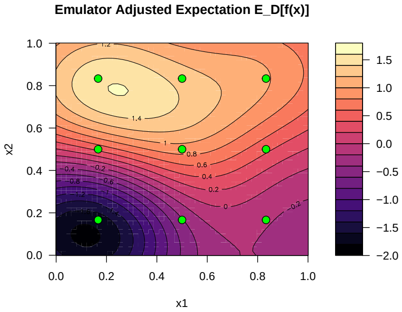

Figure: Expectation of the Emulator

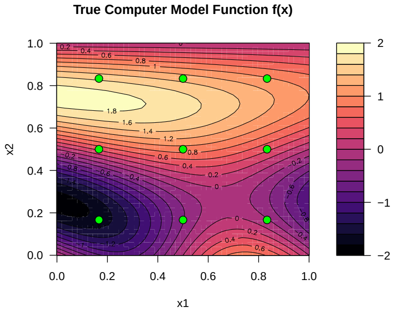

Figure: True Value of the Example Function

Improving Design

- Whilst the emulator in the previous example performs well and models the characteristics of the function well, sometimes it struggles to model some details,
- For example:
  - periodic behaviour
  - behaviour close to boundaries

Latin Hypercube Design

- Latin Hypercube Design can be used to pick the points at which to evaluate
- This is done by splitting each of the axes into n sections and making sure that there is one point in each of the n sections in each dimension.

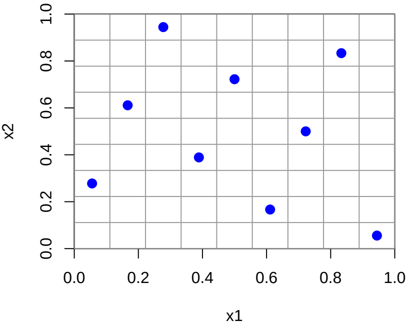

Figure: Example of a Latin Hypercube Design with 9 points

The emulator using a Latin Hypercube Design will produce the following results:

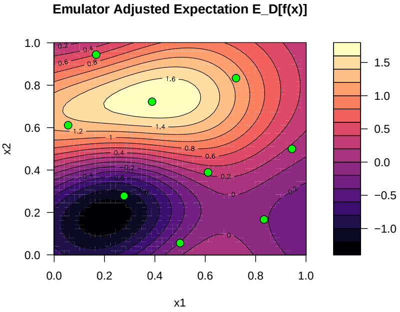

Figure: Expectation of the Emulator

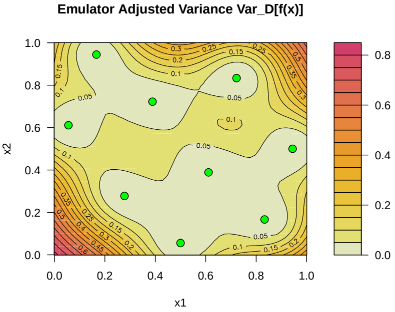

Figure: Variance of the Emulator

Emulator expectation with points in a LHD compared to a grid design:

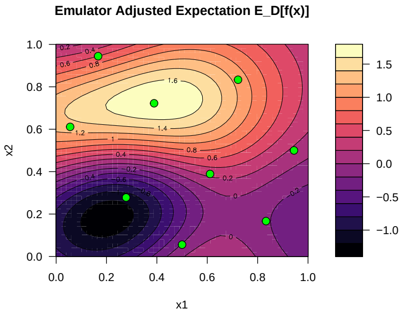

Figure: Expectation of the Emulator using a LHD

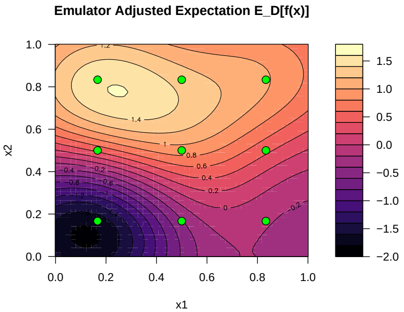

Figure: Expectation of the Emulator using a grid design

Adding a Known Boundary

Adding a Known Boundary

- One way to improve the performance of an emulator, without having to evaluate more runs, is to add a known boundary to the model.
- This performs well when setting one of the variables in the model to a certain value, simplifies the equations of the model being used.
- Setting $x_1 = 0$ in the example function simplifies it to:

$$f(x) = -1.9\sin(2\pi x_2)$$

- Given a known boundary, $\mathcal{K}$, where x^K is the orthogonal projection of the point x, onto the boundary at distance a, and $r_1(a)$ is the correlation structure, the new emulator update equations are:

$$\begin{aligned}
\mathsf{E}_K[f(x)] &= \mathsf{E}[f(x)] + r_1(a)(f(x^K) - \mathsf{E}[f(x^k)]), \\
\mathsf{Cov}_K[f(x), f(x')] &= \sigma^2(r_1(a-a') - r_1(a)r_1(a'))r_{-1}(x^K - x'^K), \\
\mathsf{Var}_K[f(x)] &= \sigma^2(1 - r_1(a)^2).
\end{aligned}$$

Before adding any design points, D, the expectation and variance given $\mathcal{K}$ are:

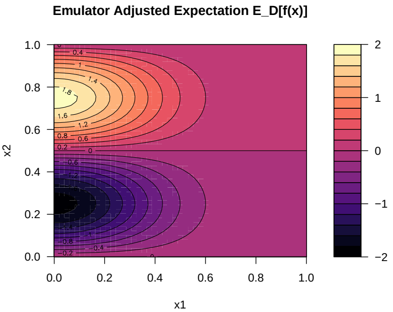

Figure: Expectation of the Emulator due to the known boundary

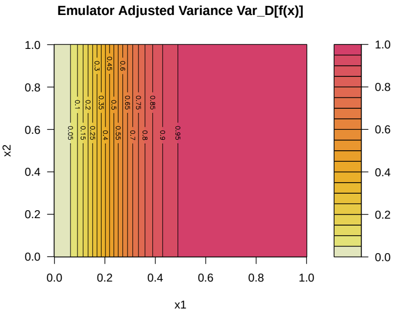

Figure: Variance of the Emulator due to the known boundary

Comparing the emulator expectation due to the boundary to the true value:

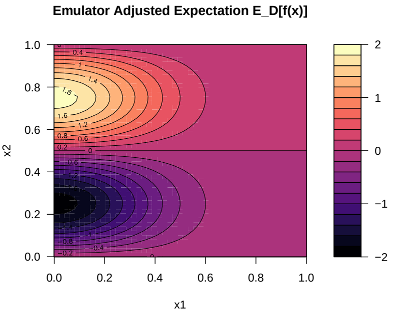

Figure: Expectation of the Emulator due to the known boundary

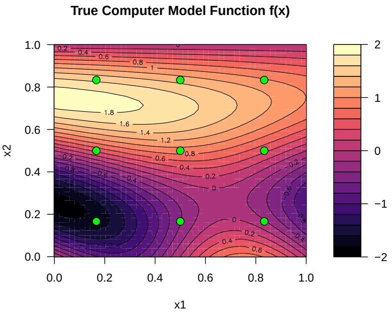

Figure: True Value of the Example Function

Lotka-Volterra Model

Lokta-Volterra Model

- The Lotka-Volterra Predator-Prey model uses deterministic differential equations to describe the populations of prey (g_1) and predators (g_2) over time, t:

$$\begin{aligned} \frac{dg_1}{dt} &= x_1 g_1 - x_2 g_1 g_2, \\ \frac{dg_2}{dt} &= x_2 g_1 g_2 - x_3 g_2, \end{aligned}$$

- The inputs x_1, x_2 and x_3 represent the reproduction speed of the prey, the predator-prey interaction, and the death rate of the predators respectively.

Using suggested input values $x_1 = 1$, $x_2 = 0.00044$, and $x_3 = 1.8$, and starting populations of 2000 prey and 800 predators produces the following behaviour for the populations over time:

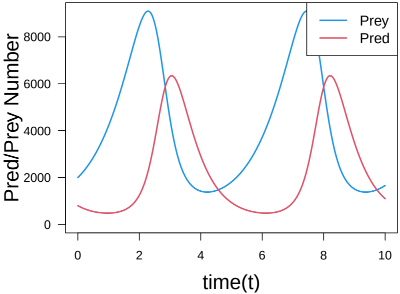

Figure: Predator and Prey Population over time.

Future Research

Future Research

- Applying the known boundary technique to the Lotka-Volterra model as setting any one of the input variables to 0 would simplify the equations, making them much faster to evaluate.
- Adding more variables to the Lotka-Volterra model, for example, additional animals to interact with the predator and prey, or adding a natural death rate for the prey.
- Adding a second known boundary to the emulator, either perpendicular or parallel to the first known boundary.

Thank you For Listening

Any Questions?

References

[1] Vernon, I., Jackson, S. E., and Cumming, J. A. Known boundary emulation of complex computer models. *SIAM/ASA Journal on Uncertainty Quantification*, 7(3):838–876, 2019.