Emulator-Based Simulator Calibration for High-Dimensional Data

Jonathan Rougier
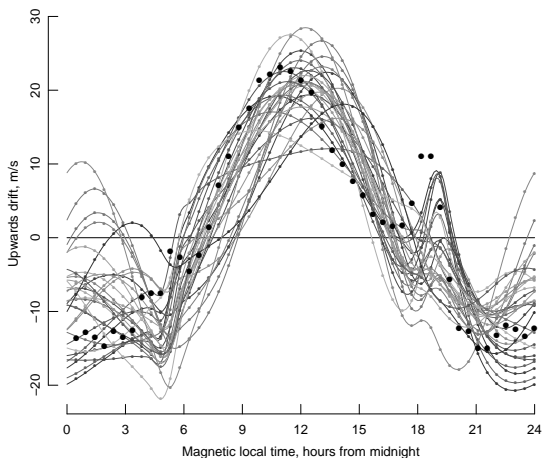
Department of Mathematics
University of Bristol, UK
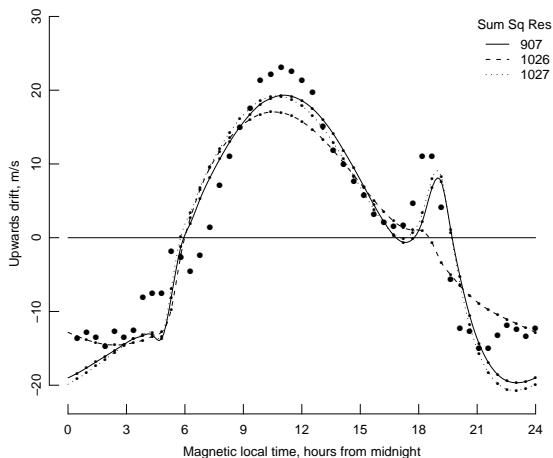
`http://www.maths.bris.ac.uk/~mazjcr/`

Aug 2009, Joint Statistics Meeting, Washington DC

TIEGCM simulator evaluations

Thirty evaluations of the TIEGCM atmospheric simulator (latin hypercube design) along with the actual observations: Jicamarca incoherent scatter radar observatory (JRO)

TIEGCM simulator evaluations

Best three evaluations by minimised sum of squared residuals. The observations seem to lie outside the convex hull of the design.

Calibration

Calibration is using the observations z^{obs} to learn about the 'best' value of the simulator parameters, θ^*. It requires a statistical model linking Z and θ^*, and then proceeds in a Bayesian fashion to compute $\Pr(\theta^* \mid Z = z^{obs})$.

Calibration

Calibration is using the observations z^{obs} to learn about the 'best' value of the simulator parameters, θ^*. It requires a statistical model linking Z and θ^*, and then proceeds in a Bayesian fashion to compute $\Pr(\theta^* \mid Z = z^{\text{obs}})$.

▶ In general the statistical model comprises two sources of uncertainty, where Y is the actual system value and $Z \perp\!\!\!\perp$ everything $\mid Y$:

$$\Pr(Y, \theta^*) = \underbrace{\Pr(Y \mid \theta^*)}_{\text{structural}} \times \underbrace{\Pr(\theta^*)}_{\text{parametric}}$$

Calibration

Calibration is using the observations z^{obs} to learn about the 'best' value of the simulator parameters, θ^*. It requires a statistical model linking Z and θ^*, and then proceeds in a Bayesian fashion to compute $\Pr(\theta^* \mid Z = z^{obs})$.

- In general the statistical model comprises two sources of uncertainty, where Y is the actual system value and $Z \perp\!\!\!\perp$ everything $\mid Y$:

$$\Pr(Y, \theta^*) = \underbrace{\Pr(Y \mid \theta^*)}_{\text{structural}} \times \underbrace{\Pr(\theta^*)}_{\text{parametric}}$$

- When working with a deterministic simulator $g(\cdot)$ we often adopt the 'best input' approach for structural uncertainty:

$$\Pr(Y \mid \theta^*) = \phi\big(Y; g(\theta^*), \Sigma\big)$$

where Σ is the discrepancy variance. This is an instantiation of the more general model $Y - g(\theta^*) \perp\!\!\!\perp \theta^*$.
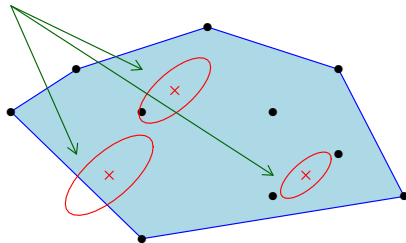
Emulation

When the simulator is expensive to evaluate, we replace $g(\cdot)$ in the inference by an emulator, which predicts $g(\theta)$ at any θ on the basis of an ensemble of simulator evaluations. Crucially, a good emulator can extrapolate somewhat beyond the convex hull of the ensemble.
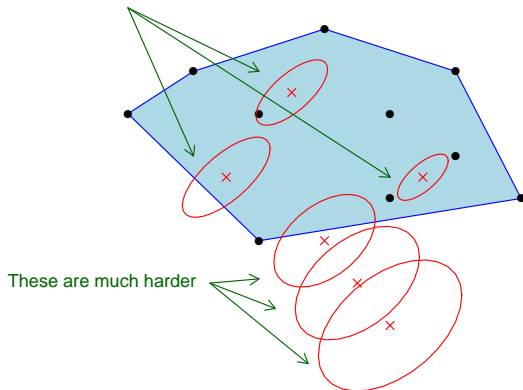
Emulation

When the simulator is expensive to evaluate, we replace $g(\cdot)$ in the inference by an emulator, which predicts $g(\theta)$ at any θ on the basis of an ensemble of simulator evaluations. Crucially, a good emulator can extrapolate somewhat beyond the convex hull of the ensemble.

These predictions are 'easy'

Emulation

When the simulator is expensive to evaluate, we replace $g(\cdot)$ in the inference by an emulator, which predicts $g(\theta)$ at any θ on the basis of an ensemble of simulator evaluations. Crucially, a good emulator can extrapolate somewhat beyond the convex hull of the ensemble.

Multivariate emulators

A multivariate emulator for when all of the simulator outputs are of the same type:

$$g_j(\theta) = \sum_k \beta_k h_k(\theta, s_j) + \varepsilon(\theta, s_j)$$

where $\boldsymbol{\beta}$ comprises uncertain coefficients, **h** is specified regressor functions, and ε is a scalar residual process.

- j indexes the simulator outputs, each of which is associated with a point s_j in the output domain $\mathcal{S}$. This emulator exploits smoothness across $\mathcal{S}$.

Multivariate emulators

A multivariate emulator for when all of the simulator outputs are of the same type:

$$g_j(\theta) = \sum_k \beta_k h_k(\theta, s_j) + \varepsilon(\theta, s_j)$$

where $\boldsymbol{\beta}$ comprises uncertain coefficients, **h** is specified regressor functions, and ε is a scalar residual process.

- j indexes the simulator outputs, each of which is associated with a point s_j in the output domain $\mathcal{S}$. This emulator exploits smoothness across $\mathcal{S}$.

- A very efficient implementation is the Outer Product Emulator (OPE), introduced in Rougier (2008). This has been implemented for the TIE-GCM simulator in Rougier *et al* (2009). [References at the end.]

The Outer Product Emulator (OPE)
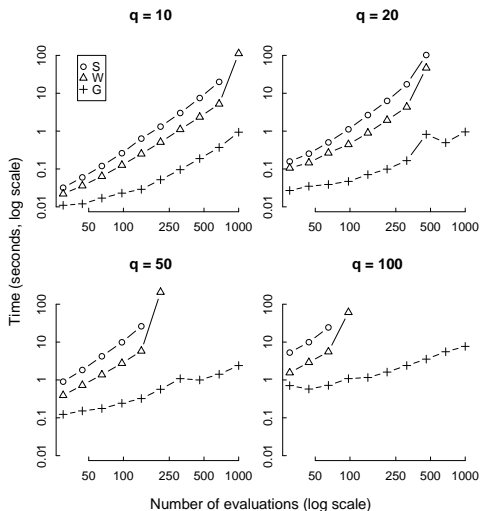
Timings for constructing and using an emulator, where q = number of outputs. 'G' shows the OPE, 'W' is the best current alternative, and 'S' is the naive implementation (from Rougier, 2008).

Predictions from the emulator mean function
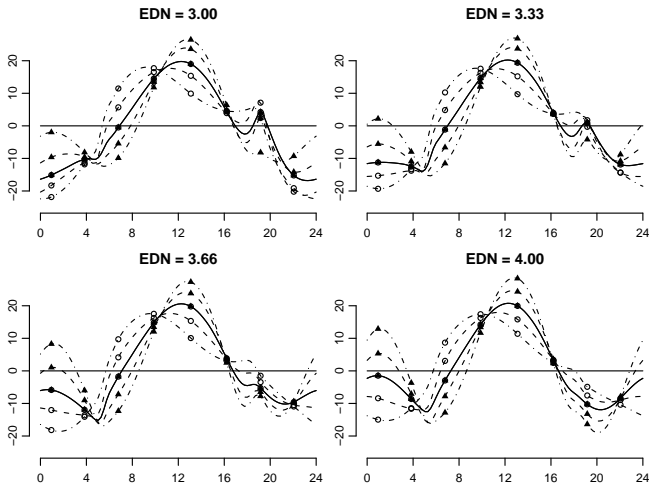
Line styles denote values of AMP: solid = 0, dashed = 18,
dot-dashed = 36. Plotting characters denote values of PHZ:
open circle = 3, filled triangle = 9. Panels show four values of EDN.

Back to calibration!

We adopt the standard approach for structural uncertainty (incorporating an emulator):

$$\Pr(Y \mid \theta^*) = \phi\big(Y; \mu(\theta^*), \Upsilon(\theta^*) + \Sigma\big)$$

where $\mu(\cdot)$ and $\Upsilon(\cdot)$ are the mean- and variance-function from the emulator, and Σ is the discrepancy variance.

Back to calibration!

We adopt the standard approach for structural uncertainty (incorporating an emulator):

$$\Pr(Y \mid \theta^*) = \phi\big(Y; \mu(\theta^*), \Upsilon(\theta^*) + \Sigma\big)$$

where $\mu(\cdot)$ and $\Upsilon(\cdot)$ are the mean- and variance-function from the emulator, and Σ is the discrepancy variance.
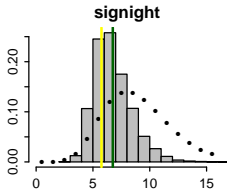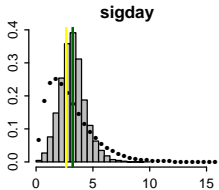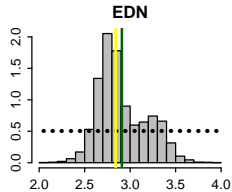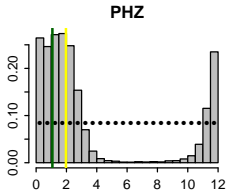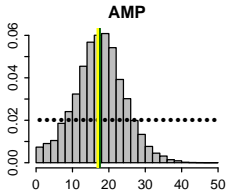
Expert judgements that affect Σ

- The simulator performs better during the day;
- Day $= [0600, 1800]$ local magnetic time (`daystart`, `dayend`);
- Standard deviations for structural uncertainty are about $3\,\mathrm{m/s}^2$ (`sigday`) and $9\,\mathrm{m/s}^2$ (`signight`);
- Some systematic effect in the simulator's mis-predictions (`ell`).
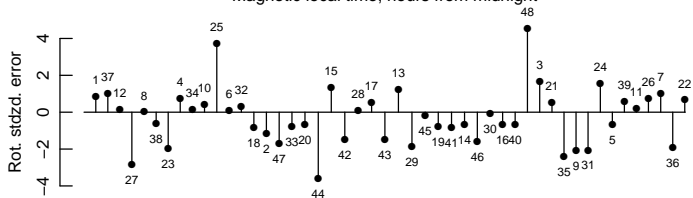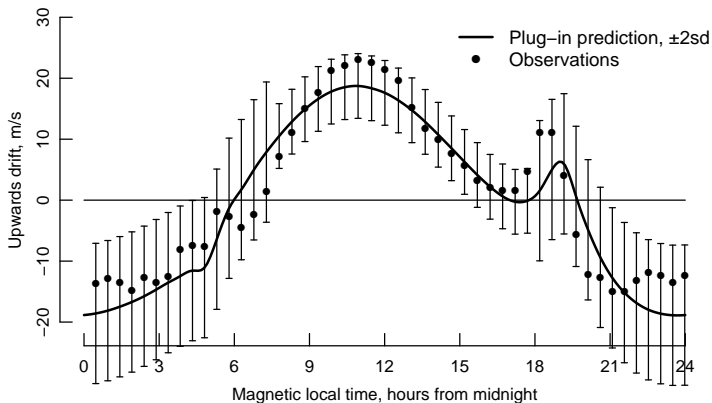
Prior predictive assessment
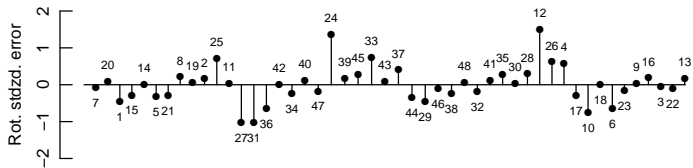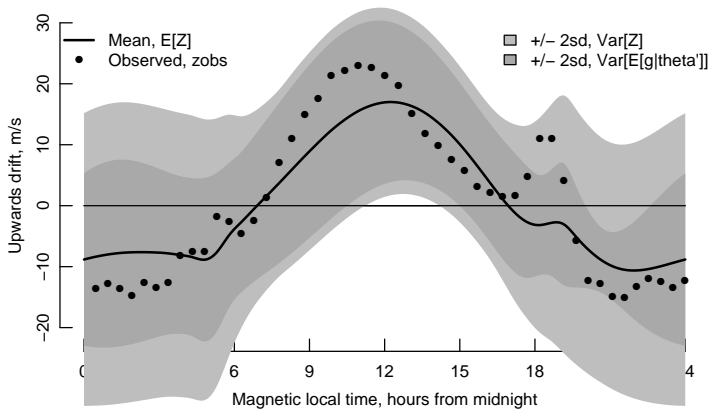
Posterior marginals
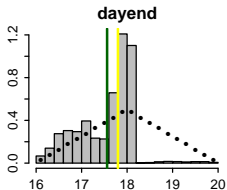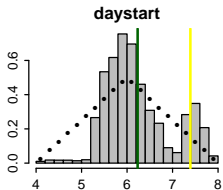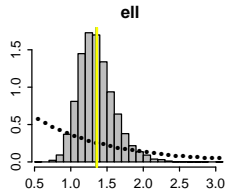
Posterior mean
Posterior mode

Plug-in posterior predictive (mode)

Reflections

The calibration exercise has been somewhat successful, but the plug-in posterior predictive shows room for improvement.

- The main finding is that EDN is probably < 3. But the original design did not use values < 3, and so all most of the inference is based on extrapolating in the emulator.

- The early-evening peak in the emulator is highly localised in time, and is captured mainly by the residual term. As we extrapolate, this tends back to its prior form, with zero mean, and large variance.

- The two solutions are (i, better) to perform additional evaluations with lower EDN values, or (ii, stop-gap) to put a more explicit representation for the early evening excursion into the emulator.

- One speculates that either of these solutions would make a large difference to the values of the Σ parameters, which are currently having to compensate.

References

J.C. Rougier (2008), Efficient Emulators for Multivariate Deterministic Functions, *Journal of Computational and Graphical Statistics*, **17(4)**, 827-843.

J.C. Rougier, S. Guillas, A. Maute, A.D. Richmond (2009), Expert Knowledge and Multivariate Emulation: The Thermosphere-Ionosphere Electrodynamics General Circulation Model (TIE-GCM). *Technometrics*, forthcoming.

S. Guillas, J.C. Rougier, A. Maute, A.D. Richmond, C.D. Linkletter (2009), Bayesian calibration of the Thermosphere-Ionosphere Electrodynamics General Circulation Model (TIE-GCM), submitted to *Geoscientific Model Development*.

Nb: the calibration in these slides has been made independently of that in Guillas *et al* (2009).