Computer model calibration as a method for design, with an application to wind turbine blades

Carl Ehrett

Clemson University Department of Mathematical Sciences

Computer experiments

Researchers increasingly look to computer experiments as a method for investigating phenomena for which it is difficult or impossible to acquire data through direct physical experimentation.

Computer model calibration

Often computer models contain unknown inputs, called calibration inputs, the values of which must be estimated for successful simulation. Examples might include when a model's output depends upon a physical constant the value of which is unknown. The value of a calibration input is often estimated by combining observations of the simulator output with real-world experimental data. Previous explorations of computer model calibration have approached calibration as a matter of bringing a computer model into agreement with physical reality.

Gaussian process emulator

Gaussian processes (GPs) can be thought of as generalizations of multivariate normal random variables. Whereas a multivariate normal random variable is a random vector of finite length, a GP is a random function. Just as a multivariate random variable is characterized by its mean vector and covariance matrix, a GP is fully characterized by its mean function $\mu : D \to R$ and covariance function $V : D \times D \to R$, where D is the domain of the process.

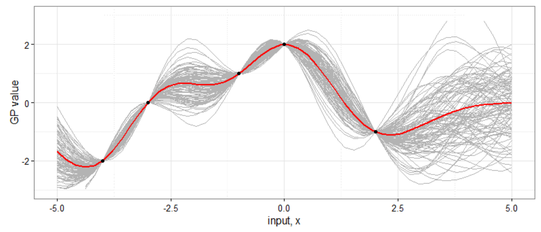

Figure 1:Figure caption

GPs are advantageous for emulating computationally expensive deterministic computer code because

- Use of a GP does not require detailed foreknowledge of the approximate parametric form of the model; we often lack such knowledge
- GPs easily interpolate the observed data; since the simulation code is deterministic and free of observation error, this is desirable
- The variance of GPs provides a natural form of uncertainty quantification (see Figure 1)

Emulator implementation

Our model inputs are a dummy input x_1 (to convert our trivariate output into univariate), temperature x_2, volume fraction x_3 and thickness x_4. For our GP emulator prior we use the covariance function

$$V(\mathbf{x}, \mathbf{x}') = \frac{1}{\lambda} \exp\{-\sum_{i=1}^{4} \beta_i (x_i - x_i')^2\}$$

where $\lambda, \boldsymbol{\beta}$ were estimated using gradient methods to find their MLEs. A prior mean $\mu(\mathbf{x}) = 0$ was also used. Observations $\eta(\mathbf{x}_i)$, $i = 1, \cdots, 504$ were collected from the simulation model using a LHC design. The updated (posterior) GP emulator has mean given by

$$\mu^*(\mathbf{x}) = v(\mathbf{x})^T \Sigma^{-1} \boldsymbol{\eta}$$

where $v(\mathbf{x}) = (V(\mathbf{x}, \mathbf{x}_1), \cdots, V(\mathbf{x}, \mathbf{x}_{504}))^T$, Σ is a matrix with $\Sigma_{i,j} = V(\mathbf{x}_i, \mathbf{x}_j)$, and $\boldsymbol{\eta} = (\eta(\mathbf{x}_1), \cdots, \eta(\mathbf{x}_{504}))^T$. The updated covariance is

$$V^*(\mathbf{x}, \mathbf{x}') = V(\mathbf{x}, \mathbf{x}') - v(\mathbf{x})^T \Sigma^{-1} v(\mathbf{x}')$$

Desired data

We calibrate the model to "desired data" which reflect extremely low tip deflection, rotation, and cost. Because we are antecedently ignorant of how close the model can come to our desired data, we place an improper $1/\sigma^2$ prior on the observation variance of each model output. We considered a range of desired outcomes.

MCMC implementation

- Calibrate volume fraction and thickness to the desired data. Set a uniform prior over each.
- Each iteration of the MCMC, draw new values for volume fraction, thickness, and the observation variances for each of tip deflection, rotation, and cost $(\sigma_d^2, \sigma_r^2, \sigma_c^2)$.
- Where $\mathbf{y}$ is the desired data, $\mathcal{D} = (\mathbf{y}, \boldsymbol{\eta})^T$ and $\Sigma_{\mathcal{D}} = \mathrm{Var}(\mathcal{D})$, the likelihood of $\mathcal{D}$ is

  $$L(\mathcal{D}|x_3, x_4, \lambda, \boldsymbol{\beta}, \Sigma_{\mathcal{D}}) = |\Sigma_{\mathcal{D}}|^{-\frac{1}{2}} \exp\{-\frac{1}{2}\mathcal{D}^T \Sigma_{\mathcal{D}}^{-1} \mathcal{D}\}$$

  hence the full posterior density is

  $$\pi(x_3, x_4, \sigma_d^2, \sigma_r^2, \sigma_c^2) \propto L(\mathcal{D}|x_3, x_4, \lambda, \boldsymbol{\beta}, \Sigma_{\mathcal{D}}) \times \pi(x_3) \times \pi(x_4) \times \pi(\sigma_d^2) \times \pi(\sigma_r^2) \times \pi(\sigma_c^2)$$
  $$= L(\mathcal{D}|x_3, x_4, \lambda, \boldsymbol{\beta}, \Sigma_{\mathcal{D}}) \times \frac{1}{\sigma_d^2 \sigma_r^2 \sigma_c^2}$$

- We eliminate boundary constraints with $z_i = \mathrm{logit}(x_i)$, $\tau_j = \log(\sigma_j^2)$ for $i = 3, 4$, $j = d, r, c$.
- Metropolis-Hastings algorithm is used with normal proposals on $z_i^{(n)}|z_i^{(n-1)}$ and $\tau_j^{(n)}|\tau_j^{(n-1)}$, $\forall i \forall j$.

Central idea

Previous explorations of computer model calibration have approached calibration as a matter of bringing a computer model into agreement with physical reality. **In the present work, we consider computer model calibration as a method for design.** Under this framework, we calibrate a computer model not using physical experimental data, but rather using "desired data" which describes the performance one hopes to achieve in the simulated system.

Additional Information

Maecenas ultricies feugiat velit non mattis. Fusce tempus arcu id ligula varius dictum.

- Curabitur pellentesque dignissim
- Eu facilisis est tempus quis
- Duis porta consequat lorem

References

[1] J. M. Smith and A. B. Jones.
Book Title.
Publisher, 7th edition, 2012.

[2] A. B. Jones and J. M. Smith.
Article Title.
Journal title, 13(52):123–456, March 2013.