Bayesian inference for computer experiments

Computer Experiments Journal Club

Oksana Chkrebtii

Overview

I will provide a brief introduction to inference for computer experiments from a Bayesian hierarchical modeling perspective. We will also briefly touch on model discrepancy as presented in the paper "Bayesian Calibration of Computer Models" by Kennedy and O'Hagan (2001).

The error model (no model discrepancy)

Let $\mathcal{X} \subset \mathbb{R}^q$ be a spatio-temporal domain. We wish to infer unknown model component, $\theta \in \Theta$, from the observations,

$$y(x_i) = m(x_i, \theta) + \varepsilon(x_i), \quad x_i \in \mathcal{X}, \;\; i = 1, \ldots, T,$$

of the deterministic state m contaminated with stochastic noise ε. This appears to be a straightforward nonlinear regression problem.

A problem of nonlinear regression?

A BHM for a nonlinear regression problem looks like this. Note that we need to be able to evaluate the model $m(\mathbf{x}, \theta)$ at arbitrary θ locations.

$$
\begin{aligned}
[y(\mathbf{x}) \mid m(\mathbf{x}, \theta), \theta] &\sim f\left(y(\mathbf{x}) \mid m(\mathbf{x}, \theta)\right) \\
[m(\mathbf{x}, \theta) \mid \theta] &\sim \delta(m(\mathbf{x}, \theta)) \\
[\theta] &\sim \pi(\theta).
\end{aligned}
$$

But: A single computer model run in many modern applications can take days, so direct penalized likelihood optimization or exact posterior sampling are no longer reasonable options.

A different formulation

The now standard approach to this problem consists of augmenting the data vector $y(\mathbf{x})$ by a vector of model runs $m(\mathbf{x}, \tilde{\boldsymbol{\theta}})$ where the model is evaluated at a fixed number of parameter regimes $\tilde{\boldsymbol{\theta}} = (\theta_1, \ldots, \theta_M)^\top$

$$
\begin{aligned}
[y(\mathbf{x}) \mid m(\mathbf{x}, \tilde{\boldsymbol{\theta}}), m(\mathbf{x}, \theta), \theta] &\sim f\left(y(\mathbf{x}) \mid m(\mathbf{x}, \theta), \theta\right) \\
\left[m(\mathbf{x}, \tilde{\boldsymbol{\theta}}) \mid m(\mathbf{x}, \theta), \theta\right] &\sim \delta(m(\mathbf{x}, \tilde{\boldsymbol{\theta}})) \\
[m(\mathbf{x}, \theta)] &\sim \mathcal{GP}(\mu, k) \\
[\theta] &\sim \pi(\theta).
\end{aligned}
$$

That is, we make predictions on the computer model across the parameter space (this is called emulation) and interrogate this surrogate model to evaluate the likelihood of y. The resulting posterior is exact.

Gaussian process regression

Definition: A Gaussian process is a collection of random variables, any finite number of which have a joint multivariate Gaussian distribution.

A Gaussian process $m(\theta)$ is completely specified by its mean function $\mu(\theta)$ and covariance function $k(\theta, \theta')$:

$$\begin{aligned} \mu(\theta) &= \mathsf{E}\left(m(\theta)\right), \\ k(\theta, \theta') &= \mathsf{E}\left[\{m(\theta) - \mu(\theta)\}\{m(\theta') - \mu(\theta')\}\right]. \end{aligned}$$

We will write,

$$m \sim \mathcal{GP}\left(\mu, k\right)$$

Gaussian process regression

We have an unknown latent function $m : [0, T] \to \mathbb{R}$ which we are trying to recover with prior,

$$m \sim \mathcal{GP}(\mu, k)$$

where the prior mean $\mu : [0, T] \to \mathbb{R}$ can be any function, for example, it can be zero for all θ. The prior covariance $k : [0, T] \times [0, T] \to \mathbb{R}$ is a bi-linear operator describing how nearby points are related to one another. Here is an example:

$$k(\theta, \theta') = \exp\left\{-\frac{1}{2}|\theta - \theta'|^2\right\}$$

This is called a squared exponential covariance, and it implies that covariance between $m(\theta)$ and $m(\theta')$ decreases with the exponential of their squared distance.

Gaussian process regression

The following joint distribution,

$$\begin{pmatrix} m(\mathbf{x}, \theta) \\ m(\mathbf{x}, \tilde{\boldsymbol{\theta}}) \end{pmatrix} \sim \mathcal{N}\left\{ \mathbf{0}, \begin{pmatrix} k(\theta, \theta) & k(\theta, \tilde{\boldsymbol{\theta}}) \\ k(\tilde{\boldsymbol{\theta}}, \theta) & k(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\theta}}) + \sigma^2 I \end{pmatrix} \right\},$$

can be used to obtain the likelihood using well-known Gaussian identities. The posterior is:

$$m(\mathbf{x}, \theta) \mid m(\mathbf{x}, \tilde{\boldsymbol{\theta}}) \sim \mathcal{N}\left(\mu(\tilde{\boldsymbol{\theta}}), \Lambda(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\theta}}) \right)$$

where

$$\mu(\tilde{\boldsymbol{\theta}}) = k(\tilde{\boldsymbol{\theta}}, \theta) \left\{ k(\theta, \theta) + \sigma^2 I \right\}^{-1} m(\mathbf{x}, \tilde{\boldsymbol{\theta}})$$

$$\Lambda(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\theta}}) = k(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\theta}}) - k(\tilde{\boldsymbol{\theta}}, \theta) \left\{ k(\theta, \theta) + \sigma^2 I \right\}^{-1} k(\theta, \tilde{\boldsymbol{\theta}})$$

For computer experiments we choose $\sigma^2 = 0$.

Gaussian process regression

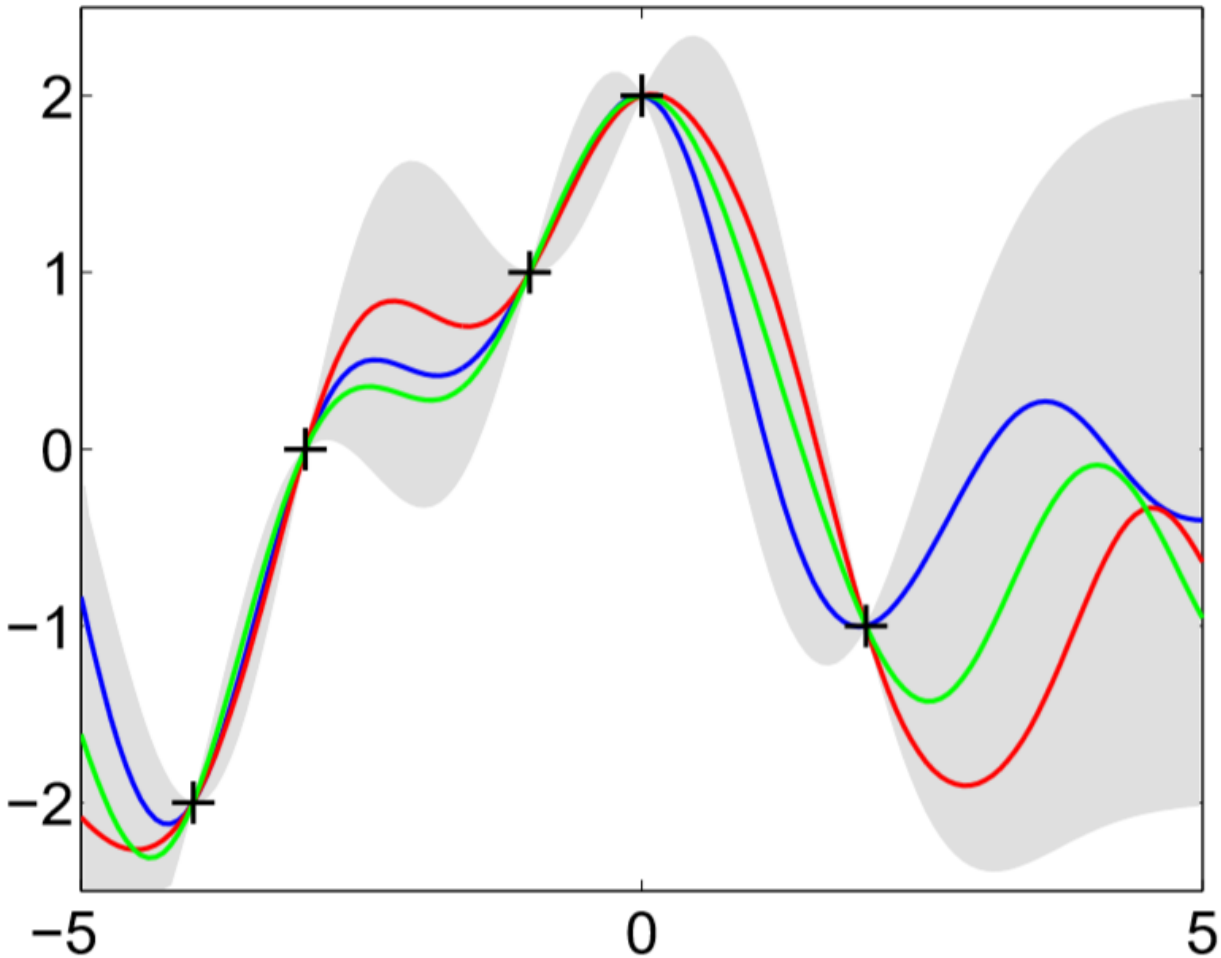

Figure from "Gaussian processes for machine learning" by Rasmussen and Williams

Discrepancy

In reality, the computer model is not perfect, and has some discrepancy that varies with location, and the error model may look like this:

$$y(x_i) = \rho m(x_i, \theta) + d(x_i) + \varepsilon(x_i), \quad x_i \in \mathcal{X}, \;\; i = 1, \ldots, T,$$

where ρ is an unknown constant and $d(\cdot)$ is a model discrepancy function which does not depend on the computer model m (Kennedy and O'Hagan, 2001). We can model ρ and $d(\cdot)$ via an additional layer in our Bayesian hierarchy. Note that since d is a function, a Gaussian process prior may be appropriate.