Description of the Gaussian process model used in GEM-SA

Marc Kennedy
University of Sheffield

29 November, 2004

Abstract

The GEM-SA (Gaussian Emulation Machine for Sensitivity Analysis) uses a Gaussian process prior probability distribution to describe beliefs about an unknown code output, as a function of the code inputs. The mathematical details behind this Gaussian process model are outlined below.

1 Standardisation

It is convenient within GEM-SA to work with a standardised set of inputs and outputs. This allows for a wide range of different codes to be treated within a generic framework, and helps to prevent numerical problems. The details that are given below all refer to the Gaussian process model of the *transformed inputs and outputs* $\boldsymbol{x}, \boldsymbol{y}$, rather than the original scale, which we call $\boldsymbol{x}', \boldsymbol{y}'$:

For each *input* component x_i, we set

$$x_i = \frac{x_i' - x_i^{min}}{x_i^{max} - x_i^{min}} \tag{1}$$

where $\boldsymbol{x}_i^{min}$ and $\boldsymbol{x}_i^{max}$ define the minimum and maximum values of all the ith input components seen within the training input data.

For the *output*, we first calculate the sample mean and variance of the output points

$$m_y = \frac{1}{n}\sum_{i=1}^{n} y_i' \tag{2}$$

$$v_y = \frac{1}{n-1}\sum_{i=1}^{n}(y_i' - m_y)^2 \tag{3}$$

$$\tag{4}$$

then we let

$$y = \frac{y' - m_y}{\sqrt{v_y}} \tag{5}$$

2 Prior Gaussian process model for the code output

After identifying a suitable *input space* $\mathscr{X} \subseteq \mathbb{R}^p$, the computer code can be represented by a smooth function $\boldsymbol{f}\colon \mathscr{X} \longmapsto \mathbb{R}$, which yields a one dimensional output vector. For any given $\boldsymbol{x} \in \mathscr{X}$ let $\boldsymbol{h}(\boldsymbol{x}) \in \mathbb{R}^m$ be a vector of arbitrary regression functions.

The following Gaussian process model is adopted:

$$\big[\boldsymbol{f}(\cdot) \mid \beta, \sigma, \boldsymbol{r}\big] \sim N\big(\boldsymbol{m}(\cdot), \sigma^2 c(\cdot,\cdot)\big), \tag{6}$$

where the mean and correlation functions take the form

$$\begin{aligned} \boldsymbol{m}(\cdot) &= \beta^T \boldsymbol{h}(\cdot), \\ c(\boldsymbol{x}_1, \boldsymbol{x}_2) &= \exp\big\{-(\boldsymbol{x}_1 - \boldsymbol{x}_2)^T \boldsymbol{R}(\boldsymbol{x}_1 - \boldsymbol{x}_2)\big\} \qquad \forall\, \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathscr{X}. \end{aligned}$$

Here the parameters are: the vector of regression coefficients $[\beta_1, \cdots, \beta_q] \in \mathbb{R}_q$; the variance σ^2 ; and the positive definite roughness matrix $\boldsymbol{R} = \mathrm{diag}\{r_i\} \in \mathbb{R}_{p,p}$. The diagonal form of $\boldsymbol{R}$ implies a correlation structure between any pair $f(\boldsymbol{x}_1)$ and $f(\boldsymbol{x}_2)$ being insensitive to inputs' interactions.

The prior specification is completed by

$$p(\beta, \sigma^2) \propto \sigma^{-2} \tag{7}$$

and for each $i = 1, \ldots, p$ independently

$$r_i \sim \mathrm{Exp}(0.01) \tag{8}$$

3 Conditioning on data: The posterior distribution

Running the computer code on a pre-selected *design set* $\mathscr{D} = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n\} \subset \mathscr{X}$ yields simulations $\boldsymbol{d} = \big[f_i(\boldsymbol{s}_r)\big] \in \mathbb{R}_n$. In light of the assumptions listed above, the joint distribution of the outcomes in $\boldsymbol{d}$ conditional on nuisance parameters β, σ^2 and $\boldsymbol{R}$ is the Normal distribution

$$[\boldsymbol{d} \mid \beta, \sigma^2, \boldsymbol{R}] \sim N_n\big(\boldsymbol{H}\beta,\ \sigma^2 \boldsymbol{A}\big), \tag{9}$$

where $\boldsymbol{H}^T = \big[\boldsymbol{h}(\boldsymbol{s}_1), \cdots, \boldsymbol{h}(\boldsymbol{s}_n)\big] \in \mathbb{R}_{m,n}$ and $\boldsymbol{A} = [c(\boldsymbol{s}_r, \boldsymbol{s}_l)] \in \mathbb{R}_{n,n}$. The code may also have a nugget term, to account for variability in the code data, possibly due to numerical errors in the solver. In this case the nugget variance $1/\lambda$ is added to each diagonal element of $\boldsymbol{A}$.

Letting $\boldsymbol{t}^T(\cdot) = \big[c(\cdot, \boldsymbol{s}_1), \cdots, c(\cdot, \boldsymbol{s}_n)\big] \in \mathbb{R}^n$, standard Normal theory and some matrix calculus manipulations thus imply the following form for the posterior conditional distribution of the computer code:

$$\big[\boldsymbol{f}(\cdot) \mid \beta, \sigma^2, \boldsymbol{R}, \boldsymbol{d}\big] \sim N_q\big(\boldsymbol{m}^\star(\cdot), \sigma^2 c^\star(\cdot,\cdot)\big), \tag{10}$$

where

$$\begin{aligned} \boldsymbol{m}^\star(\boldsymbol{x}) &= \beta^T \big[\boldsymbol{h}(\boldsymbol{x}) - \boldsymbol{H}^T \boldsymbol{A}^{-1} \boldsymbol{t}(\boldsymbol{x})\big] + \boldsymbol{d}^T \boldsymbol{A}^{-1} \boldsymbol{t}(\boldsymbol{x}), \\ c^\star(\boldsymbol{x}_1, \boldsymbol{x}_2) &= c(\boldsymbol{x}_1, \boldsymbol{x}_2) - \boldsymbol{t}^T(\boldsymbol{x}_1) \boldsymbol{A}^{-1} \boldsymbol{t}(\boldsymbol{x}_2). \end{aligned}$$

The posterior distribution of $\boldsymbol{f}(\cdot)$ conditional on the roughness matrix $\boldsymbol{R}$ alone is found by integrating (10) with respect to the posterior distribution of β and σ^2. This produces the

following marginal Student's t posterior process for the computer code $\boldsymbol{f}(\cdot)$, still conditional upon the roughness matrix $\boldsymbol{R}$:

$$[\boldsymbol{f}(\cdot) \mid \boldsymbol{R}, \boldsymbol{d}] \sim t_{n-q}\big(\boldsymbol{m}^{\star\star}(\cdot), \hat{\sigma}^2 c^{\star\star}(\cdot,\cdot)\big). \tag{11}$$

where

$$\begin{aligned} m^{\star\star}(\boldsymbol{x}) &= \hat{\boldsymbol{\beta}}^T \boldsymbol{h}(\boldsymbol{x}) + (\boldsymbol{d} - \boldsymbol{H}\hat{\boldsymbol{\beta}})^T \boldsymbol{A}^{-1}\boldsymbol{t}(\boldsymbol{x}), & (12)\\ c^{\star\star}(\boldsymbol{x}, \boldsymbol{x}') &= c^{\star}(\boldsymbol{x}, \boldsymbol{x}') + \big[\boldsymbol{h}(\boldsymbol{x}) - \boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{t}(\boldsymbol{x})\big]^T \big(\boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{H}\big)^{-1} \cdot & (13)\\ &\qquad \cdot \big[\boldsymbol{h}(\boldsymbol{x}') - \boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{t}(\boldsymbol{x}')\big] \\ & & (14) \end{aligned}$$

and where $\hat{\beta} = \big(\boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{H}\big)^{-1}\boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{d}$ is the GLS estimator of β and $\hat{\sigma}^2 = \boldsymbol{d}^T\{\boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{H}\big(\boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{H}\big)^{-1}\boldsymbol{H}^T\boldsymbol{A}^{-1}\}\boldsymbol{d}$.

The inferences required for uncertainty and sensitivity analysis are found directly from these posterior quantities for $f(\cdot)$. For more details of Gaussian process models to analyse computer models, see for example O'Hagan (2004), Sacks et al. (1989), Kennedy and O'Hagan (2001).

References

M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 63(3):425–464, 2001.

A. O'Hagan. Bayesian analysis of computer code outputs: a tutorial. *Reliability Engineering and System Safety*, submitted, 2004.

J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statist. Sci.*, 4(4):409–435, 1989. With comments and a rejoinder by the authors.