Mathematical framework for a separable Gaussian Process emulator

Roman Olson and Won Chang

November 26, 2013

1 Introduction

The purpose of this document is to present statistical background and equations for the separable emulator recently developed in our research group and implemented in R programming language. The discussion is tailored to a specific example of Greenland Ice Sheet (GIS) mass anomaly output from ice sheet model SICOPOLIS.

2 Emulator Equations

2.1 Emulator Equations

Let $y_{i,j} \in \mathbb{R}$ be physical model output of at parameter setting $\boldsymbol{\theta_i}$ and time t_j. In our case this is SICOPOLIS model output of GIS ice mass anomaly (Gt). The time settings form an n-dimensional vector $\mathbf{t} = (t_1, ..., t_n)^T$. Each parameter setting is a m-dimensional vector: $\boldsymbol{\theta_i} = (\theta_{1,i}, ..., \theta_{m,i})$. In out case, m=5. The parameter settings $\boldsymbol{\theta_i}$ form a $p \times m$ parameter matrix Θ. Then $\mathbf{y_j} = (y_{1,j}, ..., y_{p,j})^T$ is a p-dimensional vector of model outputs for all p parameters for time t_j. Consecutively, the stacked $pn \times 1$ column matrix of all model output for times from 1 through n is $\mathbf{Y} = (\mathbf{y_1}^T, ..., \mathbf{y_n}^T)^T$. Associated with $\mathbf{Y}$ is the $pn \times (m+1)$ design matrix $\mathbf{D}$. Its columns are the parameter values and time values of the ensemble. It is calculated as:

$$\mathbf{D} = \left(\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{n\times 1} \otimes \Theta \qquad \mathbf{t} \otimes \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{p\times 1} \right) \tag{1}$$

We model the SICOPOLIS output as a Gaussian process such that:

$$\mathbf{Y} \sim N(\boldsymbol{\mu_\beta}, \Sigma(\boldsymbol{\xi_y})), \tag{2}$$

where $\boldsymbol{\mu_\beta}$ is a mean function that is linear in time, and $\boldsymbol{\xi_y}$ is a vector of covariance matrix parameters. The mean for parameter setting $\boldsymbol{\theta_i}$ and time j is $\mu_{i,j}$. Specifically, $\boldsymbol{\mu_\beta} = \mathbf{X}\boldsymbol{\beta}$ where $\boldsymbol{\beta}$ is a column matrix of regression coefficients β and $\mathbf{X}$ is a matrix of covariates. It includes the column of ones (always the first column), and can also have columns of the design matrix $\mathbf{D}$. In our specific case, we are are using the mean function that is linear in time. Hence, $\boldsymbol{\beta}$ has dimension of 2×1, and $\mathbf{X}$ is $pn \times 2$. It is calculated as:

$$\mathbf{X} = \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_n \end{bmatrix}_{n\times 2} \otimes \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{p\times 1} \tag{3}$$

Under the assumption of separability, the covariance matrix Σ can be represented as a Kroenecker product of a separate covariance matrix in the time Σ_t and in parameters Σ_θ.

$$\Sigma = \Sigma_t \otimes \Sigma_\theta. \tag{4}$$

The time covariance matrix Σ_t $(n \times n)$ has AR(1) covariance. To reduce identifiability issues, we assume that the AR(1) process has innovation standard deviation of 1. Specifically, its (j, k) element is (Shumway and Stoffer, 2006):

$$\varsigma_{t,jk} = \frac{\rho^{|t_j - t_k|}}{1-\rho^2}. \tag{5}$$

where ρ is the lag-1 autocorrelation parameter.

The parameter covariance $\Sigma_\theta = [\varsigma_{\theta,ij}]$ $(p \times p)$ is assumed to be squared exponential. Its (i, j) element is:

$$\varsigma_{\theta,ij} = \kappa \exp\Big(-\sum_{k=1}^{m} \frac{|\theta_{k,i} - \theta_{k,j}|^2}{\phi_k^2}\Big) + \zeta 1(i = j). \tag{6}$$

Here κ is partial sill, ζ is nugget, and ϕ_k is range parameter for k^{th} model input parameter. The range parameters form a vector $\boldsymbol{\phi} = \phi_1, ..., \phi_m$.

Specifically, the total covariance matrix $(np \times np)$ is constructed as:

$$\Sigma = \begin{bmatrix} \varsigma_{t,11}\Sigma_\theta & \cdots & \varsigma_{t,1n}\Sigma_\theta \\ \vdots & \ddots & \vdots \\ \varsigma_{t,n1}\Sigma_\theta & \cdots & \varsigma_{t,nn}\Sigma_\theta \end{bmatrix} \tag{7}$$

Hence, the covariance parameters are $\boldsymbol{\xi_y} = (\rho, \kappa, \boldsymbol{\phi}, \zeta)^T$. The emulator parameters are $\boldsymbol{\psi}$ $=(\boldsymbol{\beta}^T, \boldsymbol{\xi_y}^T)^T$. The number of emulator parameters will be different depending on the number of model parameters used in the ensemble, and the number of covariates. In the SICOPOLIS case, this is a total of 10 parameters.

2.2 Estimating Emulator Parameters

The log-likelihood for the model output $\mathbf{Y}$ given the emulator parameters $\boldsymbol{\psi}$ can be written as (Rasmussen and Williams, 2006):

$$\ln L(\mathbf{Y}|\boldsymbol{\psi}) = -\frac{1}{2}(\mathbf{Y} - \boldsymbol{\mu_\beta})^T \Sigma^{-1} (\mathbf{Y} - \boldsymbol{\mu_\beta}) - \frac{1}{2} \ln |\Sigma| - \frac{np}{2} \ln 2\pi. \tag{8}$$

The emulator parameters $\boldsymbol{\psi}$ can be found my maximizing this likelihood over a reasonable parameter range using one of standard optimization routines. The regression parameters $\boldsymbol{\beta}$ can either be fixed, or optimized along with other emulator parameters.

2.3 Prediction

We are interested in predicting model output for all times for a given parameter vector $\boldsymbol{\theta^*}$. We denote this output, an n-dimensional vector, by $\mathbf{y}^* = (y_{\boldsymbol{\theta^*},1}, ..., y_{\boldsymbol{\theta^*},n})^T$. Associated with the prediction points is an $n \times 1$ prediction design matrix $\mathbf{D}^*$ which is constructed in a similar manner to $\mathbf{D}$. Likewise, matrix $\mathbf{X}^*$ consists of covariates evaluated at prediction points. It is constructed similarly to $\mathbf{X}$. To give an example, in the SICOPOLIS case:

$$\mathbf{X}^* = \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_n \end{bmatrix}_{n\times 2} \tag{9}$$

The prediction is a multivariate normal distribution (Rasmussen and Williams, 2006):

$$\mathbf{y}^* \sim N(\boldsymbol{\mu}^*_{\boldsymbol{\beta}}, \Sigma^*) \tag{10}$$

Here:

$$\boldsymbol{\mu}^*_{\boldsymbol{\beta}} = \mathbf{X}^*\boldsymbol{\beta} + (\Sigma_t \otimes \Sigma_{\theta^*\theta})\Sigma^{-1}(\mathbf{Y} - \boldsymbol{\mu}_{\boldsymbol{\beta}}), \tag{11}$$

where $\Sigma_{\theta^*\theta}$ is a $1 \times p$ cross-covariance matrix between the prediction parameter setting, and all the ensemble parameter settings, calculated using the same covariance function as for Σ_θ.

The predictive covariance (an $n \times n$ matrix) is given by:

$$\Sigma^* = (\kappa + \zeta)\Sigma_t - \Sigma_t \otimes \Sigma_{\theta^*\theta}\Sigma_\theta^{-1}\Sigma^T_{\theta^*\theta} \tag{12}$$

2.4 Computational Technique

Computational techniques can be used to simplify computations of

1. Likelihood, equation 8. Construct a $p \times n$ matrix $\mathbf{C}$ where $\mathbf{Y} - \boldsymbol{\mu}_{\boldsymbol{\beta}} = \text{vec}(\mathbf{C})$. The vec operation stacks the columns of a matrix into a column vector, from left to right. Then:

$$(\mathbf{Y} - \boldsymbol{\mu}_{\boldsymbol{\beta}})^T\Sigma^{-1}(\mathbf{Y} - \boldsymbol{\mu}_{\boldsymbol{\beta}}) = \text{sum}\left[\mathbf{C} * (\Sigma_\theta^{-1}\mathbf{C}\Sigma_t^{-1})\right] \tag{13}$$

2. Equation 11:

$$\boldsymbol{\mu}^*_{\boldsymbol{\beta}} = \mathbf{X}^*\boldsymbol{\beta} + (\mathbf{I}_{n\times n} \otimes \Sigma_{\theta^*\theta}\Sigma_\theta^{-1})(\mathbf{Y} - \boldsymbol{\mu}_{\boldsymbol{\beta}}). \tag{14}$$