Bayesian Dynamic Linear Modelling for Complex Computer Models

Fei Liu, Liang Zhang, Mike West

Abstract

Computer models may have functional outputs. With no loss of generality, we assume that a single computer run is generating a function of time. For complex computer models, Bayarri *et al.* (2002) considers the time as a computer model associated input parameter, and uses the Gaussian Response Surface Approximation method `(GaSP)` with the `Kronecker product` correlation matrix in the augumented space. However, this approach is only applicable when there are only a few time points. In this paper, we consider the Bayesian Dynamic Linear Model West and Harrison (1997) as an alternative approach when there are many time points. Our method also allows the forecasting for the future.

Keywords: Computer model; Bayesian Dynamic Linear Model; Gaussian stochastic process; Bayesian analysis; Forwarding filtering and backward sampling; MCMC.

1 Introduction

The computer models can be represented as deterministic functions of the associated parameters. There are generally two types of parameters: (a) calibration parameters $\boldsymbol{u}$ are only associated with the computer codes. They may be uncertain physical properties. (b) unknown parameters $\boldsymbol{x}$ are associated with both the computer models and the field experiments. They are characteristics associated with the real experiments. For simplicity, we use $\boldsymbol{x}$ to represent $(\boldsymbol{x}, \boldsymbol{u})$. As a result, we can represent the computer model as a function of $\boldsymbol{x}$, $y(\boldsymbol{x})$. On the other hand, exsercising the code is very time consuming for complex computer models. Consequently, the function $y(\boldsymbol{x})$ is only evaluated at selected locations $(\boldsymbol{x}_i, i = 1, \dots, n)$.

In this paper, we focus on the computer models with the functional outputs. We assume that the computer model outputs are functions of time $t, t = 1, \dots, T$. We represent such computer model output as $y(\boldsymbol{x}, t)$. This type of computer models has been studied both in Bayarri *et al.* (2002) and Bayarri *et al.* (2006). The `SAVE` model in Bayarri *et al.* (2002) uses the Gaussian Response Surface Approximation method (`GaSP`) on the augmented space of $(\boldsymbol{x}, t)$ by assuming seperable correlation in the space of $\boldsymbol{x}$ and t. They assume that the computer model outputs are realizations from a Gaussian Stochastic Process defined on the $(\boldsymbol{x}, t)$ space, i.e.,

$$y(\cdot,\cdot) \sim \mathrm{GP}\left(\mu, \frac{1}{\lambda^M}\mathbb{C}\mathrm{orr}\left((\cdot,\cdot),(\cdot,\cdot)\right)\right)$$

where $\mathbb{C}\mathrm{orr}\left(y(\boldsymbol{x},t), y\left(\boldsymbol{x}', t'\right)\right) = \exp\left(-\sum \beta_i \mid \boldsymbol{x}_i - \boldsymbol{x}_i' \mid^{\alpha_i}\right) \exp\left(-\beta_{(t)} \mid t - t' \mid^{\alpha_{(t)}}\right)$. We use $\boldsymbol{y}(\boldsymbol{x})$ to represent the functional output of a single computer run whose inputs is $\boldsymbol{x}$, $(\boldsymbol{y}(\boldsymbol{x}))^t = (y(\boldsymbol{x}, t_j), j = 1, \dots, T))$. The likelihood in `SAVE` is represented as,

$$\begin{pmatrix} \boldsymbol{y}(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{y}(\boldsymbol{x}_n) \end{pmatrix} \sim \mathrm{N}\left(\mu \times \mathbf{1}, \frac{1}{\lambda^M}\Sigma_1 \otimes \Sigma_2\right) \tag{1}$$

where $(\Sigma_1)_{k,l} = \exp\left(-\sum \beta_i \mid \boldsymbol{x}_{ki} - \boldsymbol{x}_{li}' \mid^{\alpha_i}\right)$ and $(\Sigma_2)_{k,l} = \exp\left(-\beta_{(t)} \mid t_k - t_l' \mid^{\alpha_{(t)}}\right)$.

To implement the `SAVE` model, one needs to invert the matrices Σ_1 and Σ_2, where Σ_1 is a n by n matrix, and Σ_2 is T by T. In the context of complex computer models, inverting Σ_1 is feasible because n is generally small. But however, the dimension of Σ_2 may be too huge to invert. Bayarri *et al.* (2006) uses basis expansion method (`SAVE2`), i.e.,

$$y\left(\boldsymbol{x},\cdot\right)=\sum_{i=1}^{I} w_i\left(\boldsymbol{x}\right)\phi_i(\cdot)$$

where $\{\phi_i(\cdot)\}$ is a basis library, and they use a wavelet for their application. Then, they model the coefficients as independent spatial processes, $w_i\left(\cdot\right)\sim \mathrm{GP}\left(\mu_i, \frac{1}{\lambda_i^M}\mathbb{C}\mathrm{orr}_i\left(\cdot,\cdot\right)\right)$.

`SAVE2` can give predictions with confidence bounds for the computer model output at any values of $\boldsymbol{x}$ by spatial interpolation. However, it can only handle the computer models with fixed time grids $t=1,\ldots,T$. Some applications of the computer model may require forecasting for the future, weather forecasting models for instance. In this paper, we will discuss modelling the computer model code by Dynamic Linear Models (`DLM`), as to capture the temporal structures in the data.

The paper is organized as follows. We will first introduce our `DLM` model and make connections with the `SAVE` model in section 2. In section 3, we will give the likelihood and specify the prior distributions for the unknown parameters associated with the `DLM` model. Section 4 discusses the `MCMC` method to get draws from the posterior distributions of the unknown quantities, and also gives spatial interpolation for the computer model at arbitray locations in the $\boldsymbol{x}$ space . The method will be applied on an example data set in section 5.

2 the `DLM` for the computer model outputs

For a single computer model run at $\boldsymbol{x}$, we use the time varying autoregressive model (`TVAR`) (West and Harrison, 1997) to model its temporal structure.

$$y(\boldsymbol{x}, t) = \sum_{j}^{p} \phi_{t,j} y(\boldsymbol{x}, t-j) + \epsilon_t(\boldsymbol{x}) \tag{2}$$

The computer model runs are correlated by assuming a Gaussian stochastic processes for the evolutions $\epsilon_t(\boldsymbol{x})$ in equation (2), i.e.,

$$\epsilon_t(\cdot) \sim \mathrm{GP}(0, v_t \mathbb{C}\mathrm{orr}^{(t)}(\cdot, \cdot)) \tag{3}$$

where, we are assuming that $\mathbb{C}\mathrm{orr}^{(t)}(\cdot,\cdot)) = \mathbb{C}\mathrm{orr}(\cdot,\cdot))$ is the same for all t. And we use seperable power exponential function for the evolution correlation, i.e.,

$$\mathbb{C}\mathrm{orr}(\boldsymbol{x}, \boldsymbol{x}^{'}) = \exp(-\sum_{i} \beta_i \mid \boldsymbol{x}_i - \boldsymbol{x}_i^{'} \mid^{\alpha_i})$$

The model in equation (2) can be connected with the `SAVE` model given in equation (1), in an approximation sense. Consider the likelihood for the `SAVE` model in equation (1). Let $\boldsymbol{y}_t = (y(\boldsymbol{x}_1, t), \ldots, y(\boldsymbol{x}_n, t))^{'}$. We represent the likelihood in equation (1) by the product of conditional likelihoods,

$$\mathrm{L}\left(\boldsymbol{y}_T, \boldsymbol{y}_{T-1}, \ldots, \boldsymbol{y}_1 \mid \Theta\right) = \left(\prod_{i=T}^{p+1} \mathrm{L}\left(\boldsymbol{y}_i \mid \boldsymbol{y}_{i-1}, \ldots, \boldsymbol{y}_1, \Theta\right)\right) \mathrm{L}\left(\boldsymbol{y}_p, \boldsymbol{y}_{p-1}, \ldots, \boldsymbol{y}_1 \mid \Theta\right) \tag{4}$$

Next, at any time t, we approximate the conditional likelihood as,

$$\mathrm{L}\left(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}, \ldots, \boldsymbol{y}_1, \Theta\right) \approx \mathrm{L}\left(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}, \ldots, \boldsymbol{y}_{t-p}, \Theta\right) \tag{5}$$

Let $\rho(k,l) = \exp\left(-\beta_{(t)} \mid k-l \mid^{\alpha_{(t)}}\right)$, $\boldsymbol{\rho}_{t,t-1:t-p} = (\rho(t,t-1), \ldots, \rho(t,t-p))^{'}$, $\left(\tilde{\Sigma}_2\right)_{k,l} = \rho(k,l), k,l = t-1, \ldots, t-p$. The conditional likelihoods in equation (5) are multivariate normals with mean vectors,

$$\begin{aligned}
\mathbb{E}\left(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1},, \ldots, \boldsymbol{y}_{t-p}, \Theta\right) &= \left(\left(\boldsymbol{\rho}_{t,t-1:t-p} \otimes \Sigma_1\right)' \left(\tilde{\Sigma}_2 \otimes \Sigma_1\right)^{-1}\right) \begin{pmatrix} \boldsymbol{y}_{t-1} \\ \vdots \\ \boldsymbol{y}_{t-p} \end{pmatrix} \\
&= \left(\left(\boldsymbol{\rho}_{t,t-1:t-p}\right)' \left(\tilde{\Sigma}_2\right)^{-1}\right) \otimes \mathbf{I}_{n\times n} \begin{pmatrix} \boldsymbol{y}_{t-1} \\ \vdots \\ \boldsymbol{y}_{t-p} \end{pmatrix}
\end{aligned}$$

This implies the auto-regressive term in equation (2),

$$y^M\left(\boldsymbol{x}, t\right) = \left(\boldsymbol{\rho}_{t,t-1:t-p}\right)' \left(\tilde{\Sigma}_2\right)^{-1} \begin{pmatrix} y^M\left(\boldsymbol{x}, t-1\right) \\ \vdots \\ y^M\left(\boldsymbol{x}, t-p\right) \end{pmatrix}$$

We assume that $\mathbb{C}\mathrm{orr}^{(t)}(\cdot,\cdot)) = \mathbb{C}\mathrm{orr}(\cdot,\cdot))$ in equation (3) because the covariance matrices of the conditional likelihoods $\mathrm{L}\left(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}, \ldots, \boldsymbol{y}_{t-p}, \Theta\right)$ is time-independent. To see this, we representat $\mathbb{C}\mathrm{ov}\left(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}, \ldots, \boldsymbol{y}_{t-p}, \Theta\right)$ as,

$$\begin{aligned}
\mathbb{C}\mathrm{ov}\left(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}, \ldots, \boldsymbol{y}_{t-p}, \Theta\right) &= \frac{1}{\lambda^M}\left(\Sigma_1 - \left(\left(\rho_{t,t-1:t-p}\right)' \otimes \Sigma_1\right)' \left(\tilde{\Sigma}_2 \otimes \Sigma_1\right)^{-1} \left(\left(\rho_{t,t-1:t-p}\right)' \otimes \Sigma_1\right)\right) \\
&= \frac{1}{\lambda^M}\left(1 - \left(\rho_{t,t-1:t-p}\right)' \left(\tilde{\Sigma}_2\right)^{-1} \left(\rho_{t,t-1:t-p}\right)\right) \Sigma_1
\end{aligned}$$

Finally, realizing that the functinal outputs of the computer models are usually temporally inhomogenous, we adapt our model to such inhomogenienty by allowing time-varying autoregressive coefficients and time-varying variances of the innovations in equation (2).

3 Likelihood and the Prior Distributions

3.1 The Multivariate DLM representation

We can represent the likelihood in the matrix form, i.e.,

$$\begin{pmatrix} y(\boldsymbol{x}_1, t) \\ y(\boldsymbol{x}_2, t) \\ \dots \\ y(\boldsymbol{x}_n, t) \end{pmatrix} = \begin{pmatrix} y(\boldsymbol{x}_1, t-1) & y(\boldsymbol{x}_1, t-2) & \dots & y(\boldsymbol{x}_1, t-p) \\ y(\boldsymbol{x}_2, t-1) & y(\boldsymbol{x}_2, t-2) & \dots & y(\boldsymbol{x}_2, t-p) \\ \vdots & \vdots & \ddots & \vdots \\ y(\boldsymbol{x}_n, t-1) & y(\boldsymbol{x}_n, t-2) & \dots & y(\boldsymbol{x}_n, t-p) \end{pmatrix} \begin{pmatrix} \phi_{t,1} \\ \phi_{t,2} \\ \vdots \\ \phi_{t,p} \end{pmatrix} + \begin{pmatrix} \epsilon_t(\boldsymbol{x}_1) \\ \epsilon_t(\boldsymbol{x}_2) \\ \vdots \\ \epsilon_t(\boldsymbol{x}_n) \end{pmatrix} \tag{6}$$

And we model the TVAR coefficients $\Phi_t = \begin{pmatrix} \phi_{t,1} \\ \phi_{t,2} \\ \vdots \\ \phi_{t,p} \end{pmatrix}$ as,

$$\Phi_t = \Phi_{t-1} + w_t$$

where $w_t \sim \mathrm{N}(0, W_t)$. Let G_t be the identity matrix of size p, $\boldsymbol{V}_t = v_t \Sigma_1$, and

$$F_t^{'} = \begin{pmatrix} y(\boldsymbol{x}_1, t-1) & y(\boldsymbol{x}_1, t-2) & \dots & y(\boldsymbol{x}_1, t-p) \\ y(\boldsymbol{x}_2, t-1) & y(\boldsymbol{x}_2, t-2) & \dots & y(\boldsymbol{x}_2, t-p) \\ \vdots & \vdots & \ddots & \vdots \\ y(\boldsymbol{x}_n, t-1) & y(\boldsymbol{x}_n, t-2) & \dots & y(\boldsymbol{x}_n, t-p) \end{pmatrix}$$

We can represent the likelihood in the way of Multivariate DLM (West and Harrison, 1997),

$$\{F_t, G_t, \boldsymbol{V}_t, W_t\}_{t=1}^{T}$$

3.2 The Prior distributions

Let D_t be the data up to time t. We sequentially specify the prior distribtions for W_t and V_t by two discounting factors δ_1, δ_2.

$$v_t^{-1} \mid \mathrm{D}_{t-1} \sim \mathrm{G}(\delta_1 n_{t-1}/2, \delta_1 d_{t-1}/2)$$

For W_t, we assume,

$$W_t \mid \mathrm{D}_{t-1} = (1-\delta_2)C_{t-1}/\delta_2, C_{t-1} = \mathrm{Cov}(\Phi_{t-1} \mid \mathrm{D}_{t-1})$$

where $C_{t-1} = \mathbb{C}\mathrm{ov}\left(\Phi_{t-1} \mid \mathrm{D}_{t-1}\right)$ and will be specified recursively in section A. The values for (n_0, d_0, C_0) will be prespecified.

Finally, for the spatial parameters $\boldsymbol{\alpha} = \{\alpha_i\}$ and $\boldsymbol{\beta} = \{\beta_i\}$, we use the Jeffereys' rule prior $\pi\left(\boldsymbol{\alpha}, \boldsymbol{\beta}\right)$ discussed in Berger *et al.* (2001) and Paulo (2005).

$$\pi\left(\boldsymbol{\alpha}, \boldsymbol{\beta}\right) \propto \mid I\left(\boldsymbol{\alpha}, \boldsymbol{\beta}\right) \mid^{1/2} \propto \sqrt{\mid \mathrm{tr}(\Sigma_1^{-1}\dot{\Sigma}_1)^2 \mid}$$

where $I\left(\boldsymbol{\alpha}, \boldsymbol{\beta}\right)$ is the Fisher information matrix, and $\dot{\Sigma}_1 = \frac{\partial \Sigma_1}{\partial(\boldsymbol{\alpha}, \boldsymbol{\beta})}$.

4 MCMC method for the Multivariate `DLM`

We use the `Monte Carlo Markov Chain` method (`MCMC`) to draw samples from the posterior distributions, $\pi\left(\{v_1, \ldots v_T\}; \{\Phi_1, \ldots, \Phi_T\}; \{\boldsymbol{\alpha}, \boldsymbol{\beta}\} \mid \mathrm{D}_T\right)$. We first give the algorithm as follows. At the i'th iteration,

1. Sample $\left(\{\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}\} \mid D_T, \{v_1^{(i-1)}, \ldots, v_T^{(i-1)}\}, \{\Phi_1^{(i-1)}, \ldots, \Phi_T^{(i-1)}\}\right)$ by the `Metroplis-Hastings` algorithm.

2. Sample $\left(\{v_1^{(i)}, \ldots, v_T^{(i)}\}, \{\Phi_1^{(i)}, \ldots, \Phi_T^{(i)}\} \mid D_T, \{\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}\}\right)$ as,

   2.1 Sample $\left(\{v_1^{(i)}, \ldots v_T^{(i)}\} \mid D_T, \{\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}\}\right)$. This will be discussed in section 4.1.

   2.2 Sample $\left(\{\Phi_1^{(i)}, \ldots, \Phi_T^{(i)}\} \mid D_T, \{v_1^{(i)}, \ldots v_T^{(i)}\}, \{\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}\}\right)$ as in section 4.2.

4.1 Sampling the variances

We give the algorithm to update the variances $\left(\{v_1^{(i)}, \ldots v_T^{(i)}\} \mid D_T, \{\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}\}\right)$.

1. Do the `Forward filtering` assuming $\{v_1, \ldots, v_T\}$ unknown, as discussed in the appendix B.

2. Sample $\left(\left(v_T^{-1}\right)^{(i)} \mid \mathrm{D}_T, \{\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}\}\right) \sim \mathrm{G}\left(n_T/2, d_T/2\right)$.

3. Sample $v_t, t = T-1, \ldots, 1$ recursively as,

$$v_t^{-1} = \delta_1 v_{t+1}^{-1} + \mathrm{G}\left((1-\delta_1)n_t/2, d_t/2\right)$$

4.2 Sampling the `TVAR` coefficients

Below is the algorithm to make draws from $\pi\left(\{\Phi_1, \ldots, \Phi_T\} \mid D_T, \{v_1, \ldots v_T\}, \{\boldsymbol{\alpha}, \boldsymbol{\beta}\}\right)$.

1. Do the `Forward filtering` conditional on $\{v_1, \ldots, v_T\}$. This will be discussed in the appendix A.

2. Sample $\left(\Phi_T \mid D_T, \{v_1, \ldots, v_T\}\right) \sim \mathrm{MVN}\left(m_T, C_T\right)$.

3. Sample $\Phi_t, t = T-1, \ldots, 1$ recursively from,

$$(\Phi_t \mid D_T, \Phi_{t+1}, \{v_1, \ldots, v_T\}) \sim \text{MVN}\left((1-\delta_2)m_t + \delta_2 \Phi_{t+1}, (1-\delta_2)C_t\right)$$

4.3 Spatial interpolation

We predict the output of a computer model at a new input value by spatial interpolation. Suppose $\boldsymbol{x}$ is the new (unexsercised) input value. Let $e_t(\boldsymbol{x}_i) = y_t(\boldsymbol{x}_i) - \sum_j y_{t-j}(\boldsymbol{x}_i)\phi_{t,j}$ and $\rho^x(\boldsymbol{x}, \boldsymbol{x}_{1:n}) = (\mathbb{C}\text{orr}(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, \mathbb{C}\text{orr}(\boldsymbol{x}, \boldsymbol{x}_n))'$, we have,

$$(y_t(\boldsymbol{x}) \mid \{y_{t-1}(\boldsymbol{x}), \ldots y_{t-p}(\boldsymbol{x})\}, \text{Data}, \{v_1, \ldots, v_T\}, \{\boldsymbol{\alpha}, \boldsymbol{\beta}\}) \sim \text{N}\left(\mu_t(\boldsymbol{x}), \sigma_t^2(\boldsymbol{x})\right)$$

where,

$$\mu_t(\boldsymbol{x}) = \sum_j y_{t-j}(\boldsymbol{x})\phi_{t,j} + v_t^{-1}\rho^x(\boldsymbol{x}, \boldsymbol{x}_{1:n})\Sigma_1^{-1} \begin{pmatrix} e_t(\boldsymbol{x}_1) \\ e_t(\boldsymbol{x}_2) \\ \vdots \\ e_t(\boldsymbol{x}_n) \end{pmatrix}$$

and,

$$\sigma_t^2(\boldsymbol{x}) = v_t\left(1 - \rho^x(\boldsymbol{x}, \boldsymbol{x}_{1:n})\,\Sigma_1^{-1}\rho^x(\boldsymbol{x}, \boldsymbol{x}_{1:n})\right)$$

As all the computer model emulators do, the `DLM` modelling approach gives back the computer model output, when we are trying to make predictions for the exsercised computer input values. In other words, if $x \in \{x_1, \ldots, x_n\}$, we have $\mu_t(\boldsymbol{x}) = y_t(\boldsymbol{x})$ and $\sigma_t^2(\boldsymbol{x}) = 0$.

5 An example

5.1 The data

Figure 1 gives an example of the functional outputs of computer models. Each time series is associated with an $\boldsymbol{x}$ value located to the left of the series. The $\boldsymbol{x}$ values are considered as the computer model inputs. The data with $\boldsymbol{x} = 0.5$ (in red) is obtained from some real physical experiment. This data is observed at T = 3000 time points. We use $\boldsymbol{y}_t(0.5) = (y_t(0.5), t = 1, \ldots, T)$ to represent it. Given $\boldsymbol{y}_t(0.5)$ and its $\texttt{TVAR}_{20}$ fit $\{\phi_{t,j}, v_t\}$, we simulate the data for $\boldsymbol{x} = 0.25, \ldots, 0.75$ by fixing $\alpha = 2, \beta = 1.6$. The details are discussed in Appendix C.

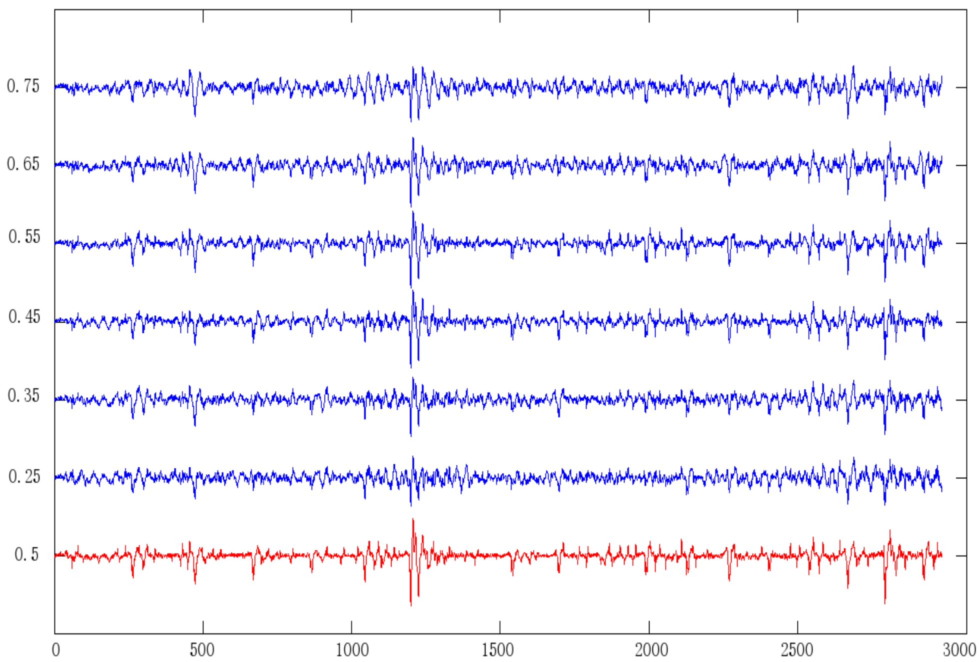

Figure 1: The simulated computer model data at various input values

5.2 MCMC Results

In section 4, we can perfectly sample $\{v_1^{(i)}, \ldots, v_T^{(i)}\}, \{\Phi_1^{(i)}, \ldots, \Phi_T^{(i)}\}$ conditional on $\{\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}\}$. This implies that, we do not need to update $\{v_1^{(i)}, \ldots, v_T^{(i)}\}, \{\Phi_1^{(i)}, \ldots, \Phi_T^{(i)}\}$ in every iteration. In particular, we update $\{v_1^{(i)}, \ldots, v_T^{(i)}\}, \{\Phi_1^{(i)}, \ldots, \Phi_T^{(i)}\}$ after every 200 iterations of sampling $\{\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}\}$ by the `Metroplis-Hastings` algorithm. We fix $\{\alpha^{(i)}\}$ at 2 for the example data set. For the other unknowns, starting the `MCMC` from "true" parameter values, we obtained $N = 2000$ samples, among which the first 1000 are treated as burnin samples and will be discarded in all the posterior inferences. Figure 2 gives the trace plot, prior distribution (up to a normalizing constant), posterior distribution, autocorrelation function for β. For the purpose of making comparison between the prior and the posterior distribution for β, we highlight with red line the prior distribution in the interval $(1, 2)$, within which the posterior draws are concentrated.

Suppose $\left\{\phi_{t,j}^{(i)}\right\}$ is the i'th `MCMC` draw for the `TVAR` coefficients $\{\phi_{t,j}\}$, where $i = 1, \ldots,$ N, $t = 1, \ldots, T$, and $j = 1, \ldots, 20$. We calculate the posterior mean for $\phi_{t,j}$, $\hat{\phi}_{t,j}$ by,

$$\hat{\phi}_{t,j} = \frac{1}{N} \sum_i \phi_{t,j}^{(i)}$$

And the point-wise posterior means of the `TVAR` coefficients are shown in the left panel of figure 3. The right panel shows $\{\hat{v}_t, t = 1, \ldots, T\}$, the point-wise posterior means of $\{v_t\}$.

5.3 Spatial interpolation

One direct application of the multivariate `DLM`, as we discussed in section 4.3, is to get the prediction for the computer model at input other than the design points. In figure 4, we give our prediction for the dynamic computer model outputs at input value $\boldsymbol{x} = 0.5$. We also make comparison between the true outputs and our prediction at the time intervals

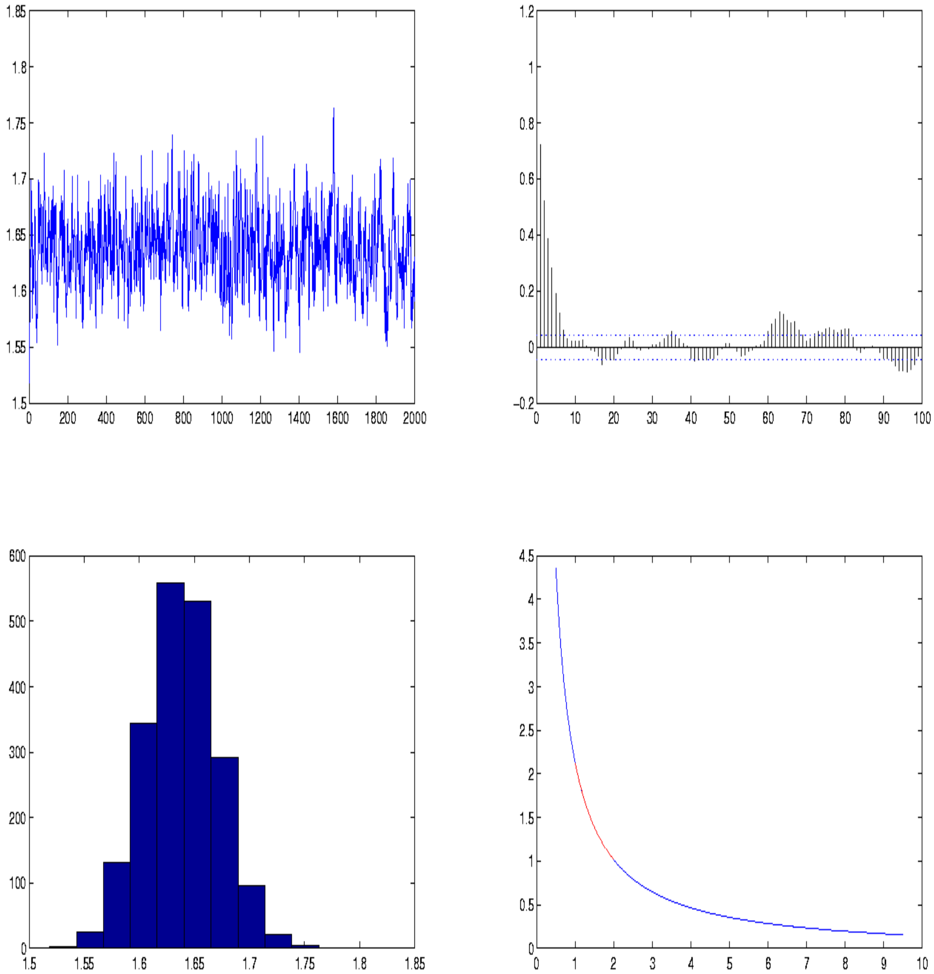

Figure 2: Upper-left: trace plot of the MCMC samples for β; Upper-Right: autocorrelation functions of the MCMC samples for β; Lower-Left: posterior distribution of β; Lower-Right: prior density of β.

$(1100, 1300)$ and $(2700, 2900)$, where the data is exhibiting interesting features.

5.4 Wave and modular decomposition

We can decompose the process $\{y(t)\}$ as

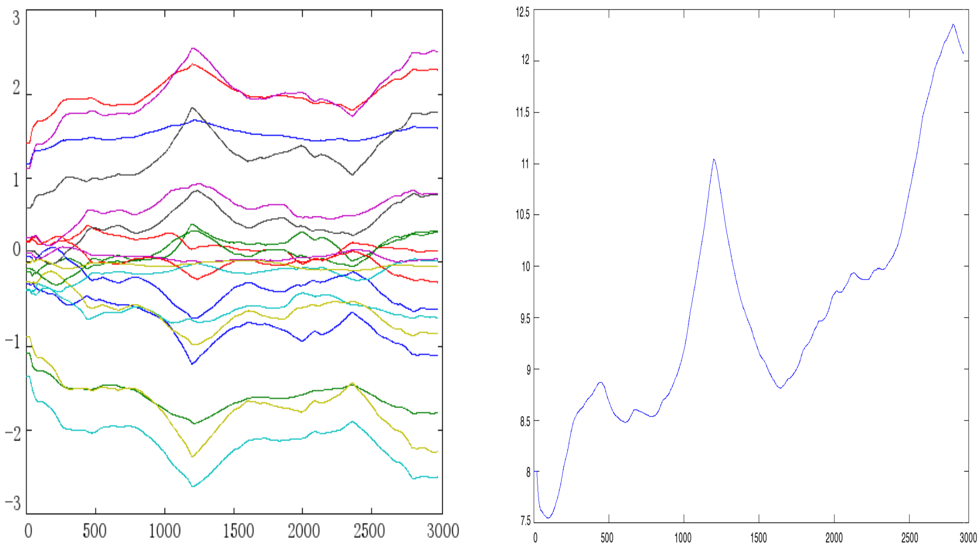

Figure 3: Left: posterior means for the `TVAR` coefficients $\{\phi_{t,j}\}$; Right: posterior means for the time varying variances $\{v_t\}$

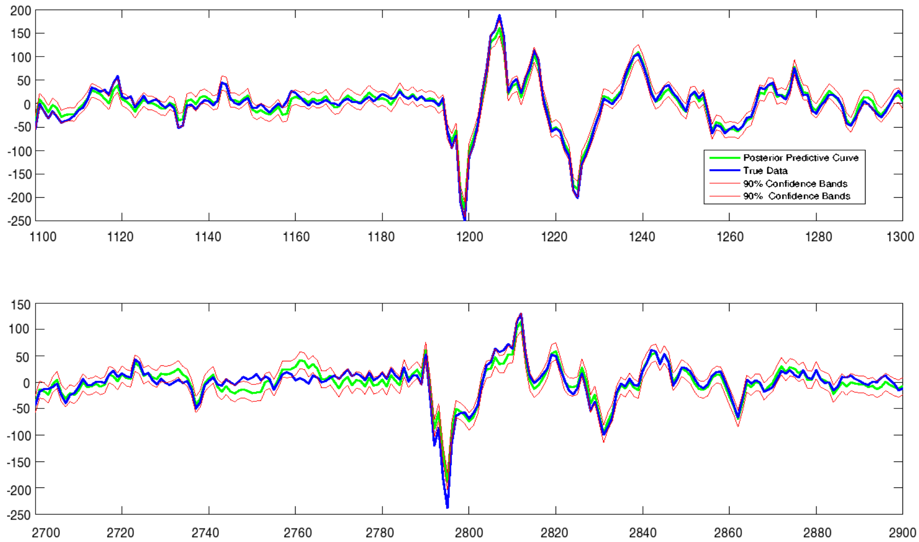

Figure 4: Posterior predictive curve (green), true computer model output (red), and 90% piece-wise predictive intervals for spatial interpolation with input value $\boldsymbol{x} = 0.5$.

$$y_t = \sum_{l=1}^{c} z_{t,l} + \sum_{l=1}^{r} x_{t,l}$$

where the latent processes $\{z_{t,l}\}$ are TVAR's with lag 1 and $x_{t,l}$ are stochastically time-varying damped harmonic components, each of which is associated with the modulars (damping parameters) $\{a_{t,l}\}$ and the wavelengths (periods) $\{\lambda_{t,l}\}$ (West and Harrison, 1997). Such decomposition can help to understand the physics meanings of the computer model outputs. In Figure 5, we show the decompositions for the posterior mean of the process $\{y_t(0.5)\}$. In Figure 6, we show the modulars and the wavelengths of the first 5 components, as a function of t.

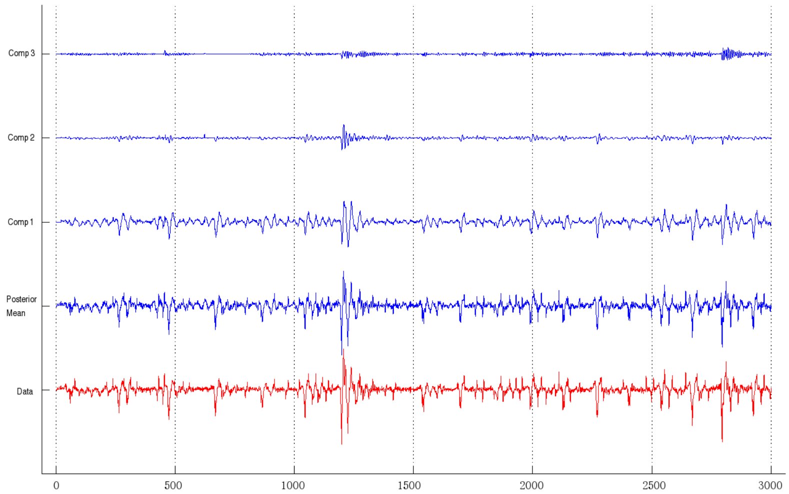

Figure 5: The true computer model output data $\{y_t(0.5)\}$(bottom), posterior mean for $\{y_t(0.5)\}$(second to the bottom), and decomposition of the posterior mean (the rest curves are the first to the third components from bottom to the top).

A Forward filtering with known variances

We briefly review the `forward filtering` algorithm with known variances for multivariate `DLM`. For more details, refer to the Chapter 16 in West and Harrison (1997).

With (m_0, C_0),

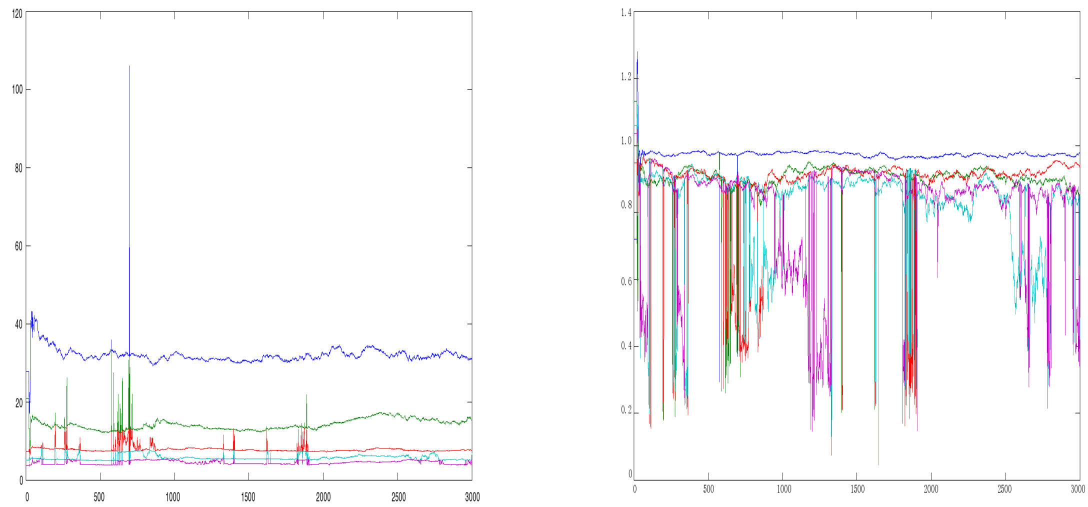

Figure 6: Left: wave decompositions; Right: modular decompositions

(a). Posterior at $t-1$: $(\Phi_{t-1} \mid \mathrm{D}_{t-1}) \sim \mathrm{N}(m_{t-1}, C_{t-1})$

(b). Prior at t: $(\Phi_t \mid \mathrm{D}_{t-1}) \sim \mathrm{N}(a_t, R_t)$, with,

$$a_t = m_{t-1}, R_t = C_{t-1}/\delta_2$$

(c). One-step forecast: $(\boldsymbol{y}_t \mid \mathrm{D}_{t-1}) \sim \mathrm{N}(\boldsymbol{f}_t, Q_t)$, with,

$$\boldsymbol{f}_t = F_t^{'} a_t = F_t^{'} m_{t-1}; Q_t = F_t^{'} C_{t-1} F_t/\delta_2 + v_t$$

(d). Poserior at t: $(\Phi_t \mid \mathrm{D}_t) \sim \mathrm{N}(m_t, C_t)$

with,

$$m_t = a_t + A_t e_t \qquad \text{and} \qquad C_t = R_t - A_t Q_t A_t^{'}$$

where,

$$A_t = R_t F_t Q_t^{-1} \qquad \text{and} \qquad e_t = Y_t - f_t$$

B Forward filtering with unknown variances

We first describe the `forward filtering` algorithm with unknown variances for multivariate `DLM`.

With m_0, C_0, s_0, n_0,

(a). Posterior at $t-1$: $(\Phi_{t-1} \mid \mathrm{D}_{t-1}) \sim \mathrm{N}(m_{t-1}, C_{t-1})$

(b). Prior at t: $(\Phi_t \mid \mathrm{D}_{t-1}) \sim \mathrm{N}(a_t, R_t)$, with,

$$a_t = m_{t-1}, R_t = C_{t-1}/\delta_2$$

(c). One-step forecast: $(\boldsymbol{y}_t \mid \mathrm{D}_{t-1}) \sim \mathrm{N}(\boldsymbol{f}_t, Q_t)$, with,

$$\boldsymbol{f}_t = F_t^{'} a_t = F_t^{'} m_{t-1}; Q_t = F_t^{'} C_{t-1} F_t/\delta_2 + s_{t-1}\Sigma_1$$

(d). Posterior at t: $(\Phi_t \mid \mathrm{D}_t) \sim \mathrm{T}_{n_t}(m_t, C_t)$, and, $(V_t^{-1} \mid \mathrm{D}_t) \sim \mathrm{G}(n_t/2, d_t/2)$, with,

$$A_t = R_t F_t Q_t^{-1} = C_{t-1} F_t Q_t^{-1}/\delta_2$$

where $m_t = m_{t-1} + A_t e_t$, $e_t = \boldsymbol{y}_t - F_t^{'} m_{t-1}$, and $C_t = \frac{s_t}{s_{t-1}} \left(\frac{C_{t-1}}{\delta_2} - A_t Q_t A_t^{'} \right)$, and,

$$n_t = \delta_1 n_{t-1} + n; d_t = \delta_1 d_{t-1} + s_{t-1} e_t^t Q_t^{-1} e_t \tag{7}$$

Now, we derive the relationship in equation (7). At time t, the prior for v_t^{-1} is,

$$\left(v_t^{-1} \mid \mathrm{D}_{t-1}\right) \sim \mathrm{G}(\delta_1 n_{t-1}/2, \delta_1 d_{t-1}/2)$$

The likelihood,

$$\left(e_t \mid \mathrm{D}_{t-1}, v_t^{-1}\right) \sim \mathrm{N}(0, Q_t)$$

Therefore, the posterior distribution for v_t^{-1} is,

$$\pi(v_t^{-1} \mid \mathrm{D}_t) \quad \propto \quad \frac{1}{\mid v_t Q_t \mid^{1/2}} \exp(-\frac{s_{t-1}}{v_t} \epsilon_t^t Q_t^{-1} \epsilon_t)(v_t^{-1})^{\delta_1 n_{t-1}/2-1} \exp(-\delta_1 d_{t-1} v_t^{-1}/2)$$

This implies that,

$$(v_t^{-1} \mid \mathrm{D}_t) \sim \mathrm{G}\left(\left(n + \delta_1 n_{t-1}\right)/2, \left(\delta_1 d_{t-1} + s_{t-1} \epsilon_t^t Q_t^{-1} \epsilon_t\right)/2\right)$$

In other words,

$$\begin{aligned} n_t &= \delta_1 n_{t-1} + n; \\ d_t &= \delta_1 d_{t-1} + s_{t-1} e_t^t Q_t^{-1} e_t; \\ s_t &= d_t / n_t \end{aligned}$$

C Data simulation

Suppose we have a functional data $(y_t(\boldsymbol{x}), t = 1, \ldots, T)$. Given $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \{\boldsymbol{\Phi}_t\}, \{v_t\})$, we simulate the data as,

$$\begin{pmatrix} y_t(\boldsymbol{x}_1) \\ y_t(\boldsymbol{x}_2) \\ \cdots \\ y_t(\boldsymbol{x}_n) \end{pmatrix} \mid \{y_{t-1}(\boldsymbol{x}_i)\}, \{\boldsymbol{\Phi}_t\}, y_t(\boldsymbol{x}) \quad \sim \quad \text{MVN} \left(\boldsymbol{\mu}_t = \begin{pmatrix} \mu_t(\boldsymbol{x}_1) \\ \mu_t(\boldsymbol{x}_2) \\ \cdots \\ \mu_t(\boldsymbol{x}_n) \end{pmatrix}, (v_t)\Sigma \right)$$

with,

$$\mu_t(\boldsymbol{x}_i) \quad = \quad \sum_j \phi_{t,j} Y_{t-j}(\boldsymbol{x}_i) + (v_t)^{-1} \rho^{(x)}\left(\boldsymbol{x}, \boldsymbol{x}_{1:n}\right) \left(y_t(\boldsymbol{x}) - \sum_j \phi_{t,j} y_{t-j}(\boldsymbol{x}) \right)$$

and,

$$\Sigma = \Sigma_1 - \rho^{(x)}(\boldsymbol{x}; \boldsymbol{x}_{1:n}) \rho^{(x)}(\boldsymbol{x}; \boldsymbol{x}_{1:n})$$

Σ_1 is $n \times n$ matrix with the (i, j) element $(\Sigma_1)_{i,j} = \mathbb{C}\text{orr}(\boldsymbol{x}_i, \boldsymbol{x}_j)$. And $\rho^{(x)}(\boldsymbol{x}; \boldsymbol{x}_{1:n})$ is an n by 1 vector with the i'th element $\left(\rho^{(x)}(\boldsymbol{x}; \boldsymbol{x}_{1:n})\right)_i = \mathbb{C}\text{orr}(\boldsymbol{x}, \boldsymbol{x}_i)$.

References

Bayarri, M., Berger, J., Garcia-Donato, G., Liu, F., Palomo, J., Paulo, R., Sacks, J., Walsh, D., Cafeo, J., and Parthasarathy, R. (2006). Computer model validation with functional outputs. *Niss tech. report* .

Bayarri, M., Berger, J., Higdon, D., Kennedy, M., Kottas, A., Paulo, R., Sacks, J., Cafeo, J., Cavendish, J., Lin, C., and Tu, J. (2002). A framework for validation of computer models. In D. Pace and S. S. (Eds.), eds., *Proceedings of the Workshop on Foundations for V&V in the 21st Century*. Society for Modeling and Simulation International.

Berger, J., Oliveira, V. D., and Sanso, B. (2001). Objective bayesian analysis of spatially correlated data. *JASA* 1361–1374.

Paulo, R. (2005). Default priors for gaussian processes. *Annals* 556–582.

West, M. and Harrison, P. (1997). *Bayesian Forecasting and Dynamic Models*. Springer, New York, USA.