MULTI-FREQUENCY SCALE GAUSSIAN REGRESSION FOR NOISY TIME-SERIES DATA

Kian Seng Neo[1], W. E. Leithead[1,2]

[1]*Hamilton Institute, National University of Ireland, Maynooth, Co. Kildare, Ireland*
[2]*Dept of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, U.K.*
e-mail: kian.neo@.strath.ac.uk, w.leithead@eee.strath.ac.uk

Abstract: Regression using Gaussian process models is applied to time-series data analysis. To extract from the data separate components with different frequency scales, the Gaussian regression methodology is extended through the use of multiple Gaussian process models. Fast and memory-efficient methods, as required by Gaussian regression to cater for large time-series data sets, are discussed. These methods are based on the generalised Schur algorithm and a procedure to determine the Schur decomposition of matrices, the key step to realising them, is presented. In addition, a procedure to appropriately initialise the Gaussian process model training is presented. The utility of the procedures is illustrated by application of a multiple Gaussian process model to extract separate components with different frequency scales from a 5000-point time-series data set with gaps.

Keywords: Gaussian regression, multi-length scale, time-series analysis, missing data, generalised Schur algorithm.

1. INTRODUCTION

Following the early work by MacKay (1998) and Williams (1999), there has been increasing interest in the application of Gaussian process prior models to data analysis (Gibbs and MacKay, 2000; Yoshioka, and Ishii, 2001), including data filtering and fitting, statistical modelling and system identification. Gaussian regression based on models with two stochastic processes is discussed in Leithead et al (2005b). During training of the prior model the two Gaussian processes are conditioned on data subject to the condition that they remain independent. Separate components in the data with different characteristics or, more precisely, their description by probability distributions can then be extracted.

In this paper, the application of Gaussian regression to time-series data analysis is considered. A brief overview of Gaussian regression is given in Section 2. The methodology based on Gaussian process models with two stochastic processes is extended in Section 3 to models with M stochastic processes. When applied to time-series data, separate components with different frequency scales can then be extracted. However, for a data set of size N, many matrix manipulations requiring $O(N^3)$ operations and $O(N^2)$ memory-storage, such as matrix inversion and log-determinant, are necessary during training and prediction. Since the matrices, encountered in the application of Gaussian regression to the analysis of large time-series data sets, are Toeplitz-like, fast and memory-efficient methods for matrix manipulation are possible. Methods, based on the Schur algorithm rather than the Modified Levinson-Durbin's algorithm (Zhang and Leithead, 2004), are more general and so are preferred; for example, when analysing time-series data with gaps. Fast and memory-efficient methods based on the Schur algorithm, with the focus on the key step of determining the Schur decomposition, are discussed in Section 4. A procedure to determine the Schur decomposition is proposed. In addition, training of the Gaussian process prior models is non-convex.

Hence, it may be inefficient and can converge on an incorrect model. A procedure to ensure appropriate initialisation, when applying Gaussian regression to time-series data, is presented in Section 5. Finally, the utility of the above procedures is illustrated in Section 6 by application of a multiple Gaussian process model to extract independent components from a time-series data set with gaps.

2. GAUSSIAN PROCESS PRIOR MODELS

A brief explanation of the standard Gaussian regression methodology and its application to data analysis is reviewed in this section. Consider a smooth scalar function f(.) dependent on the explanatory variable, $\mathbf{z} \in \mathrm{D} \subseteq \Re^p$. Suppose N measurements, $\{(\mathbf{z}_i, \mathrm{y}_i)\}_{i=1}^N$, of the value of the function with additive Gaussian white measurement noise, i.e. y_i=f($\mathbf{z}_i$)+n_i, are available and denote them by M. It is of interest here to use this data to learn the mapping f($\mathbf{z}$) or, more precisely, to determine a probabilistic description of f($\mathbf{z}$) on the domain, D, containing the data. Note that this is a regression formulation and it is assumed the input $\mathbf{z}$ is noise free. The probabilistic description of the function, f($\mathbf{z}$), adopted is the stochastic process, $\mathrm{f}_\mathbf{z}$, with the E[$\mathrm{f}_\mathbf{z}$], as $\mathbf{z}$ varies, interpreted to be a fit to f($\mathbf{z}$). By necessity, to define the stochastic process, $\mathrm{f}_\mathbf{z}$, the probability distributions of $\mathrm{f}_\mathbf{z}$ for every choice of value of $\mathbf{z} \in \mathrm{D}$ are required together with the joint probability distributions of $\mathrm{f}_{\mathbf{z}_i}$ for every choice of finite sample, {$\mathbf{z}_1$,…,$\mathbf{z}_k$}, from D, for all k>1. Given the joint probability distribution for $\mathrm{f}_{\mathbf{z}_i}$, i=1..N, and the joint probability distribution for n_i, i=1..N, the joint probability distribution for y_i, i=1..N, is readily obtained since the measurement noise, n_i, and the f($\mathbf{z}_i$) (and so the $\mathrm{f}_{\mathbf{z}_i}$) are statistically independent. M is a single event belonging to the joint probability distribution for y_i, i=1..N.

In the Bayesian probability context, the prior belief is placed directly on the probability distributions describing $\mathrm{f}_\mathbf{z}$ which are then conditioned on the information, M, to determine the posterior probability distributions. In particular, in the Gaussian process prior model, it is assumed that the prior probability distributions for the $\mathrm{f}_\mathbf{z}$ are all Gaussian with zero mean (in the absence of any evidence the value of f($\mathbf{z}$) is as likely to be positive as negative). Only a definition of the covariance function $C(\mathbf{z}_i, \mathbf{z}_j) = E[\mathrm{f}_{\mathbf{z}_i}, \mathrm{f}_{\mathbf{z}_j}]$, for all $\mathbf{z}_i$ and $\mathbf{z}_j$, is required to complete the statistical description. The resulting posterior probability distributions are also Gaussian. This model is used to carry out inference as follows.

Clearly, by Bayes' rule, p($\mathrm{f}_\mathbf{z}$|M)=p($\mathrm{f}_\mathbf{z}$,M)/p(M) where p(M) acts as a normalising constant. Hence, with the Gaussian prior assumption,

$$p(\mathrm{f}_\mathbf{z}, M) \propto \exp\left[-\frac{1}{2}\begin{bmatrix}\mathrm{f}_\mathbf{z} & \mathbf{Y}^T\end{bmatrix}\begin{bmatrix}\Lambda_{11} & \mathbf{\Lambda}_{12} \\ \mathbf{\Lambda}_{21} & \mathbf{\Lambda}_{22}\end{bmatrix}^{-1}\begin{bmatrix}\mathrm{f}_\mathbf{z} \\ \mathbf{Y}\end{bmatrix}\right]$$

where $\mathbf{Y}$=[y_1,…y_N]$^\mathrm{T}$, Λ_{11} is E[$\mathrm{f}_\mathbf{z}$, $\mathrm{f}_\mathbf{z}$], the ij^{th} element of the covariance matrix $\mathbf{\Lambda}_{22}$ is E[$\mathrm{y_i}$, $\mathrm{y_j}$] and the i^{th} element of vector $\mathbf{\Lambda}_{21}$ is E[$\mathrm{y_i}$, $\mathrm{f}_\mathbf{z}$]. Both Λ_{11} and $\mathbf{\Lambda}_{21}$ depend on $\mathbf{z}$. Applying the partitioned matrix inversion lemma, it follows that

$$p(\mathrm{f}_\mathbf{z} \mid M) \propto \exp\left[-\frac{1}{2}\left(\mathrm{f}_\mathbf{z} - \hat{\mathrm{f}}_\mathbf{z}\right)\Lambda_\mathbf{z}^{-1}\left(\mathrm{f}_\mathbf{z} - \hat{\mathrm{f}}_\mathbf{z}\right)\right]$$

with $\mathrm{f}_\mathbf{z} = \mathbf{\Lambda}_{12}\mathbf{\Lambda}_{22}^{-1}\mathbf{Y}$, $\Lambda_\mathbf{z} = \Lambda_{11} - \mathbf{\Lambda}_{12}\mathbf{\Lambda}_{22}^{-1}\mathbf{\Lambda}_{21}$. Therefore, the prediction from this model is that the most likely value of f($\mathbf{z}$) is the mean, $\hat{\mathrm{f}}_\mathbf{z}$, with variance Λ_z. Note that $\hat{\mathrm{f}}_\mathbf{z}$ is simply a $\mathbf{z}$-dependent weighted linear combination of the measured data points, $\mathbf{Y}$, using weights $\mathbf{\Lambda}_{12}\mathbf{\Lambda}_{22}^{-1}$. The measurement noise, n_i, i=1,..N, is statistically independent of f($\mathbf{z}_\mathrm{i}$), i=1,..N, and has covariance matrix $\mathbf{B}$. Hence the covariances for the measurements, y_i, are simply

$$\mathrm{E}[\mathrm{y}_{i,}\mathrm{y}_j] = \mathrm{E}[\mathrm{f}_{\mathbf{z}_i}, \mathrm{f}_{\mathbf{z}_j}] + \mathrm{B}_{ij}\ ;\ \mathrm{E}[\mathrm{y}_{i,}\mathrm{f}_\mathbf{z}] = \mathrm{E}[\mathrm{f}_{\mathbf{z}_i}, \mathrm{f}_\mathbf{z}]$$

The prior covariance function is generally dependent on a few hyperparameters, θ. To obtain a model given the data, M, the hyperparameters are adapted to maximise the likelihood, p(M|θ), or equivalently to minimise the negative log likelihood, $L(\theta)$,

$$L(\theta) = \frac{1}{2}\log|C(\theta)| + \frac{1}{2}\mathbf{Y}^T C(\theta)^{-1}\mathbf{Y} \qquad (1)$$

where $C(\theta) = \mathbf{\Lambda}_{22}$, the covariance of the measurements.

When, as here, the data set is a time series, the explanatory variable is simply time, i.e. $\mathbf{z}_i$=t_i. A common choice of the covariance function for time series analysis is $a\exp[-\frac{1}{2}d(\mathrm{t}_i - \mathrm{t}_j)^2]$, where the hyperparameter, d, is related to the length-scale of the data and the hyperparameter, a, to the amplitude of the data such that $a = \mathrm{E}[\mathrm{f}_{\mathrm{t}_i}, \mathrm{f}_{\mathrm{t}_i}]$. Assuming the measurement noise is white, its covariance function is $n\delta_{ij}$, where n is the noise variance such that $n = \mathrm{E}[n_i, n_i]$. It follows that the covariance function for the measured data and so the ij-th element of $C(\theta)$ is

$$a\left\{\exp[-\tfrac{1}{2}d(\mathrm{t}_i - \mathrm{t}_j)^2] + b\delta_{ij}\right\} \qquad (2)$$

where $n = ab$. The hyperparameters for the prior model (2) are θ=(a,d,b).

3. MODELS WITH MULTIPLE GAUSSIAN PROCESSES

The procedure outlined in Section 2 is very effective when used to identify a single function. However, suppose that the measurements are the sum of the values of M functions, each with different characteristics; that is, the measured values are $y_i = \mathrm{f}_1(\mathbf{z}_i) + \ldots + \mathrm{f}_\mathrm{M}(\mathbf{z}_i) + n_i$. The case with M=2 is discussed in detail in (Leithead et al 2005b). A possible probabilistic description of $\mathrm{h}(\mathbf{z}) = \mathrm{f}_1(\mathbf{z}) + \ldots + \mathrm{f}_\mathrm{M}(\mathbf{z})$ is by means of the sum of M independent Gaussian processes, $\mathrm{f}_{1\mathbf{z}}$,..,$\mathrm{f}_{\mathrm{M}\mathbf{z}}$. Let the covariance functions for these independent

Gaussian processes be $C_{\mathrm{f_l}}(\mathbf{z_i},\mathbf{z_j}),\ldots,C_{\mathrm{f_M}}(\mathbf{z_i},\mathbf{z_j})$, respectively, then $\mathrm{h}_{\mathbf{z}}=\mathrm{f}_{\mathrm{l}\mathbf{z}}+\cdots+\mathrm{f}_{\mathrm{M}\mathbf{z}}$, is itself a Gaussian process with covariance function $C_{\mathrm{h}}=C_{\mathrm{f_l}}+\cdots+C_{\mathrm{f_M}}$.

With $\mathbf{F}_{k\mathbf{z}}=[\mathrm{f}_{k\mathbf{z}_1},\cdots,\mathrm{f}_{k\mathbf{z}_K}]^T$, the ij$^{\mathrm{th}}$ element of $\mathbf{\Lambda}_{\tilde{\mathbf{z}}\hat{\mathbf{z}}}^{\mathbf{F}_m\mathbf{F}_n}=\mathrm{E}[\mathbf{F}_{m\tilde{\mathbf{z}}},\mathbf{F}_{n\hat{\mathbf{z}}}]$ is $\mathbf{\Lambda}_{\tilde{\mathbf{z}}_i\hat{\mathbf{z}}_j}^{\mathrm{f}_m\mathrm{f}_n}$. The prior joint probability distribution for $\mathbf{F}_{\mathrm{l}\hat{\mathbf{z}}},\ldots,\mathbf{F}_{\mathrm{M}\hat{\mathbf{z}}}$ and $\mathbf{Y}$ is Gaussian with mean zero and covariance matrix

$$\mathbf{\Lambda}=\mathrm{E}[[\mathbf{F}_{\mathrm{l}\hat{\mathbf{z}}}^T\ \vdots\ \cdots\ \vdots\ \mathbf{F}_{\mathrm{M}\hat{\mathbf{z}}}^T\ \vdots\ \mathbf{Y}^T]^T[\mathbf{F}_{\mathrm{l}\hat{\mathbf{z}}}^T\ \vdots\ \cdots\ \vdots\ \mathbf{F}_{\mathrm{M}\hat{\mathbf{z}}}^T\ \vdots\ \mathbf{Y}^T]]$$

$$=\begin{bmatrix}\mathbf{\Lambda}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}^{\mathbf{F}_\mathrm{l}\mathbf{F}_\mathrm{l}} & 0 & \cdots & 0 & \mathbf{\Lambda}_{\hat{\mathbf{z}}\mathbf{z}}^{\mathbf{F}_\mathrm{l}\mathbf{F}_\mathrm{l}}\\ 0 & \mathbf{\Lambda}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}^{\mathbf{F}_2\mathbf{F}_2} & \cdots & 0 & \mathbf{\Lambda}_{\hat{\mathbf{z}}\mathbf{z}}^{\mathbf{F}_2\mathbf{F}_2}\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & \cdots & \mathbf{\Lambda}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}^{\mathbf{F}_\mathrm{M}\mathbf{F}_\mathrm{M}} & \mathbf{\Lambda}_{\hat{\mathbf{z}}\mathbf{z}}^{\mathbf{F}_\mathrm{M}\mathbf{F}_\mathrm{M}}\\ \mathbf{\Lambda}_{\mathbf{z}\hat{\mathbf{z}}}^{\mathbf{F}_\mathrm{l}\mathbf{F}_\mathrm{l}} & \mathbf{\Lambda}_{\mathbf{z}\hat{\mathbf{z}}}^{\mathbf{F}_2\mathbf{F}_2} & \cdots & \mathbf{\Lambda}_{\mathbf{z}\hat{\mathbf{z}}}^{\mathbf{F}_\mathrm{M}\mathbf{F}_\mathrm{M}} & \mathbf{Q}\end{bmatrix}$$

where $\mathbf{Q}=\mathbf{B}+\mathbf{\Lambda}_{\mathbf{zz}}^{\mathbf{F}_\mathrm{l}\mathbf{F}_\mathrm{l}}+\cdots+\mathbf{\Lambda}_{\mathbf{zz}}^{\mathbf{F}_\mathrm{M}\mathbf{F}_\mathrm{M}}$ and $\mathbf{B}$ is the noise covariance matrix. Applying the partitioned matrix lemma, the posterior joint probability distribution for $\mathbf{F}_{\mathrm{l}\hat{\mathbf{z}}},\ldots,\mathbf{F}_{\mathrm{M}\hat{\mathbf{z}}}$ conditioned on the data, $\mathbf{Y}$ remains Gaussian with mean, $\overline{\mathbf{M}}$, and covariance matrix, $\overline{\mathbf{\Lambda}}$, where

$$\overline{\mathbf{M}}=\begin{bmatrix}\mathbf{\Lambda}_{\mathbf{z}\hat{\mathbf{z}}}^{\mathbf{F}_\mathrm{l}\mathbf{F}_\mathrm{l}} & \cdots & \mathbf{\Lambda}_{\mathbf{z}\hat{\mathbf{z}}}^{\mathbf{F}_\mathrm{M}\mathbf{F}_\mathrm{M}}\end{bmatrix}^{\mathbf{T}}\mathbf{Q}^{-1}\mathbf{Y}$$

$$\overline{\mathbf{\Lambda}}=\mathrm{diag}\begin{Bmatrix}\mathbf{\Lambda}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}^{\mathbf{F}_\mathrm{l}\mathbf{F}_\mathrm{l}}\\ \vdots\\ \mathbf{\Lambda}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}^{\mathbf{F}_\mathrm{M}\mathbf{F}_\mathrm{M}}\end{Bmatrix}-[\mathbf{\Lambda}_{\hat{\mathbf{z}}\mathbf{z}}^{\mathbf{F}_\mathrm{l}\mathbf{F}_\mathrm{l}}\ \cdots\ \mathbf{\Lambda}_{\hat{\mathbf{z}}\mathbf{z}}^{\mathbf{F}_\mathrm{M}\mathbf{F}_\mathrm{M}}]\mathbf{Q}^{-1}\begin{bmatrix}\mathbf{\Lambda}_{\mathbf{z}\hat{\mathbf{z}}}^{\mathbf{F}_\mathrm{l}\mathbf{F}_\mathrm{l}}\\ \vdots\\ \mathbf{\Lambda}_{\mathbf{z}\hat{\mathbf{z}}}^{\mathbf{F}_\mathrm{M}\mathbf{F}_\mathrm{M}}\end{bmatrix}$$

The mean and covariance matrix for $\mathbf{H}_{\hat{\mathbf{z}}}=\mathbf{F}_{\mathrm{l}\hat{\mathbf{z}}}+\cdots+\mathbf{F}_{\mathrm{M}\hat{\mathbf{z}}}$ are, respectively, $\mathbf{\Lambda}_{\hat{\mathbf{z}}\mathbf{z}}^{\mathbf{H}}\mathbf{Q}^{-1}\mathbf{Y}$ and $\mathbf{\Lambda}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}^{\mathbf{H}}-\mathbf{\Lambda}_{\hat{\mathbf{z}}\mathbf{z}}^{\mathbf{H}}\mathbf{Q}^{-1}\mathbf{\Lambda}_{\mathbf{z}\hat{\mathbf{z}}}^{\mathbf{H}}$, where $\mathbf{\Lambda}_{\mathbf{z}\hat{\mathbf{z}}}^{\mathbf{H}}=\mathbf{\Lambda}_{\mathbf{z}\hat{\mathbf{z}}}^{\mathbf{F}_\mathrm{l}\mathbf{F}_\mathrm{l}}+\ldots+\mathbf{\Lambda}_{\mathbf{z}\hat{\mathbf{z}}}^{\mathbf{F}_\mathrm{M}\mathbf{F}_\mathrm{M}}$.

However, the requirement here is to determine the posterior probability distribution for $\mathbf{F}_{\mathrm{l}\mathbf{z}},\ldots,\mathbf{F}_{\mathrm{M}\mathbf{z}}$, conditioned on the data, subject to the condition that they remain independent. It is met (Leithead et al 2005b) through a transformation of the $\mathbf{F}_{\mathrm{l}\mathbf{z}},\ldots,\mathbf{F}_{\mathrm{M}\mathbf{z}}$ such that the mean and covariance of the posterior joint probability distribution become, respectively,

$$\begin{bmatrix}\mathbf{\Lambda}_1\mathbf{Q}_1^{-1}\mathbf{Y}\\ \mathbf{B}\mathbf{Q}_1^{-1}\mathbf{\Lambda}_2\mathbf{Q}_2^{-1}\mathbf{Y}\\ \\ \mathbf{B}\mathbf{Q}_{\mathrm{M}-1}^{-1}\mathbf{\Lambda}_\mathrm{M}\mathbf{Q}_\mathrm{M}^{-1}\mathbf{Y}\end{bmatrix}\quad\text{and}\quad diag\begin{Bmatrix}\mathbf{\Lambda}_1\mathbf{Q}_1^{-1}\mathbf{B}\\ \mathbf{B}\mathbf{Q}_1^{-1}\mathbf{\Lambda}_2\mathbf{Q}_2^{-1}\mathbf{B}\\ \vdots\\ \mathbf{B}\mathbf{Q}_{\mathrm{M}-1}^{-1}\mathbf{\Lambda}_\mathrm{M}\mathbf{Q}_{\mathrm{M}-1}^{-1}\mathbf{B}\end{Bmatrix}$$

where $\mathbf{\Lambda}_k=\mathbf{\Lambda}_{\mathbf{zz}}^{\mathbf{F}_\mathrm{l}\mathbf{F}_\mathrm{l}}+\cdots+\mathbf{\Lambda}_{\mathbf{zz}}^{\mathbf{F}_k\mathbf{F}_k}$ and $\mathbf{Q}_k=\mathbf{\Lambda}_k+\mathbf{B}$. (Note, the likelihood of the data remains unaffected by the transformation).

When applied to time-series data, these multiple Gaussian process models enable separate components with different frequency scales to be extracted.

4. GENERALISED SCHUR ALGORITHM

For time-series data with a constant sampling interval, as here, the covariance matrix, $C(\theta)$, is Toeplitz (or, when there are gaps in the data, block-Toeplitz), and has low displacement rank. Applying the generalised Schur algorithm (Kailath and Sayed, 1999), many manipulations of these low displacement rank matrices require only $O(N^2)$ operations, rather than $O(N^3)$. These fast memory-efficient methods then enable the use of Gaussian regression with large time-series data.

Consider a positive-definite matrix, $R\in\mathbb{R}^{N\times N}$. The triangular decomposition is denoted by

$$R=LD^{-1}L^T$$

where $D = diag\{d_0,d_1,..d_{N-1}\}$ is a diagonal matrix. The lower triangular matrix L is normalised so that the elements on its main diagonal are the $\{d_i\}$. This LDL-decomposition can be obtained through the Schur reduction algorithm.

The Schur algorithm applies to strongly regular Hermitian Toeplitz-like matrices, R, satisfying

$$\Delta R=R-FRF^*\cong GJG^*,\quad J=J^T,J^2=I \qquad (3)$$

for some full rank generator matrix G, and lower triangular matrix, F, for which the diagonal elements, $\{f_i\}$, satisfy

$$1-f_if_j^*\neq 0 \qquad \text{for all } i,j \qquad (4)$$

The signature matrix, J, is defined to be J-unitary, $J=(I_p\oplus -I_q)$, where p and q are, respectively, the number of strictly positive and strictly negative eigen-values of ΔR. $K=p+q$, the total number of non-zero eigen-values, is the displacement rank of R.

A key requirement is a procedure to determine explicitly the rank-revealing decomposition, $\{G, F$ and $J\}$. In the context of Gaussian process prior models, R is real, symmetric and positive-definite. (Its successive Schur complements are also positive-definite.) J is simply defined to be $(I_{K/2}\oplus -I_{K/2})$, i.e. $p = q$. F is the strictly lower-triangular shift matrix, $F=(Z_{N_1}\oplus Z_{N_2}\oplus Z_{N_3}\oplus\ldots)$, depending on the number of inner Toeplitz-blocks inside R. Matrix Z_N is defined here to be a square lower-triangular shift matrix with ones on the first subdiagonal and zeros elsewhere (i.e. a lower-triangular Jordan block with eigenvalue equal to zero). The generator matrix, G, can be obtained by the following procedure.

Procedure 1:

1) Let $R\in\mathbb{R}^{N\times N}$ be a symmetrical and Hermitean matrix, with low displacement rank, $K<<N$, such that the reduced-row echelon form (RREF) is

$$\Delta R=\left[\begin{array}{c:c}\widetilde{A} & \widetilde{B}^T\\ \hdashline \widetilde{B} & 0\end{array}\right]\equiv\left[B\ \vdots\ 0\right]+\left[\begin{array}{c}B^T\\ \hdashline 0\end{array}\right]-\left[\begin{array}{c:c}A & 0\\ \hdashline 0 & 0\end{array}\right]$$

where $B^T=\left[\widetilde{A}\ \vdots\ \widetilde{B}^T\right]$ and $A\in\mathbb{R}^{K/2\times K/2}$ is symmetric. Any symmetric Hermitean block-Toeplitz matrix can be transformed into a matrix with the above RREF by permuting its rows and columns. Let $\Gamma E=ED$ be the eigen-value decomposition of

$$\Gamma=\left[\begin{array}{c:c}A & I\\ \hdashline B^TB-AA & 0\end{array}\right] \qquad (5)$$

where D is diagonal. The non-zero eigen-values of ΔR are real and positive and identical to the eigen-values of Γ. In addition, the eigen-vectors of ΔR are real and equal to the columns of $BX+[I\,\vdots\,0]^T Y$ where $[X^T\,\vdots\,Y^T]^T = E \in \mathbb{R}^{K\times K}$.

2) The values of some of the eigen-values of G can be very similar. For numerical reasons, the computed eigen-values and eigen-vectors can then include complex conjugate pairs, i.e. D and E are complex. Although, the imaginary parts of these computed eigen-values are extremely small, the imaginary parts of the eigen-vectors can be large. Hence, to ensure D and E are real, the following corrections are made,

$$D \to real(D)$$
$$E \to real(E)+imag(E)$$

3) Each column of $BX+[I\,\vdots\,0]^T Y = \Psi \in \mathbb{R}^{N\times K}$ is an eigen-vector of ΔR with eigen-value unchanged; that is, the diagonal elements of D. However, since all the eigen-values are not distinct, the columns of Ψ are not automatically orthonormal as required. To enforce orthogonality the columns of Ψ are updated recursively for $k = 2,\ldots K$ such that, $\forall i \geq k$,

$$\Psi_i \to \Psi_i - \frac{\Psi_{k-1}(\Psi_{k-1}^T \Psi_i)}{\Psi_i^T \Psi_i}$$

where Ψ_i is the i-th column of Ψ. To obtain orthonormality the columns are then rescaled such that, $\forall k$,

$$\Psi_k \to \Psi_k / \sqrt{\Psi_k^T \Psi_k}$$

4) With $\Psi \in \mathbb{R}^{N\times K}$, obtained as above, $\Delta R \cong \Psi D \Psi^T$. D has a decomposition

$$D = X(H\Omega H^T)X^T$$

where H is the unitary permutation matrix separating the positive and negative eigen-values, λ_i from each other. X is a diagonal matrix with its diagonal elements comprised of the square-root of the absolute values of the eigen-values, λ_i; that is

$$X = diag(\sqrt{|d_1|},\cdots,\sqrt{|d_K|})$$

The required decomposition of ΔR is obtained with

$$J = \Omega$$
$$G = \Psi X H$$

that is, $\Delta R \cong GJG^T$, where $G\underline{\Delta}\Omega XH$ is the generator matrix for ΔR.

5. HYPERPARAMETER INITIALISATION

As discussed in Section 2, in Gaussian regression the hyperparameters, on which the covariance function depends, must first be trained; that is, to obtain the Gaussian process prior model given some data M, the hyperparameters are adapted to maximise the likelihood of the data or equivalently to minimise the negative log likelihood, (1). However, in general, minimising the log likelihood is not a simple convex optimisation problem; for example, the log likelihood can have multiple local minima. The local minima can be associated with different aspects of the data. For instance, when the data is a time series consisting of a long length-scale component and a short length-scale component, one minimum may correspond to the long length-scale and another minimum to the short length-scale component. Depending on the choice of initial values for the hyperparameters, the outcome of the optimisation process could be a model of either. To obtain an efficient optimisation of the hyperparameters, that converges quickly, it is essential to choose appropriate initial values. In this section, a procedure for doing so, when Gaussian regression is applied to time-series data, is presented.

For time-series data, suppose the mean of the data is zero. (if not it can always be made so). The initial values of the hyperparameters $\theta=(a,d,b)$ for the covariance function (2) are determined by the following procedure. The discussion is illustrated using the power spectral density in Fig.1.

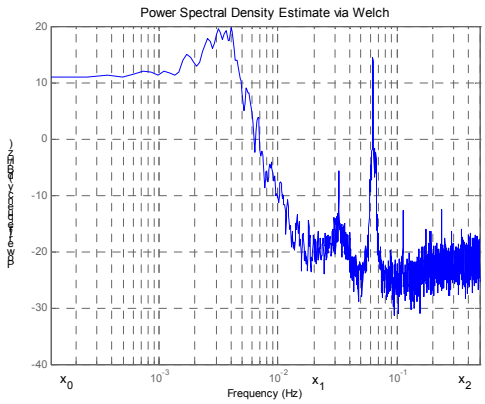

Fig. 1. Power spectrum of time-series data.

Procedure 2:

1) Provided the time series data is of sufficient length, its variance is roughly equal to $a+n$, since the amplitude hyperparameter $a = \mathrm{E}[\mathrm{f}_{t_i}, \mathrm{f}_{t_i}] = \mathbf{Y}^T\mathbf{Y}/N$ and the noise hyperparameter $n = \mathrm{E}[n_i, n_i]$. Let $\mathrm{V_y}$ and $\mathrm{V_n}$, respectively, be the variances of the measured data and the measurement noise. It follows that

$$v = \mathrm{V}_n / \mathrm{V}_y \approx n/(a+n) = b/(1+b)$$

2) The value for $\mathrm{V_y}$ is easily estimated. Since different values of the hyperparameters, especially the length-scale hyperparameter, correspond to models with different length-scale, the value for $\mathrm{V_n}$ depends on the choice of time-series components that is interpreted to be noise. For example, the spectral density in Fig 1 clearly indicates that the corresponding time series data consists of several components with different length-scales. Only the long length-scale component with frequency less than χ_1 might be of interest when all components with higher frequency are interpreted as noise. In this case, $\mathrm{V_n}$ would be estimated as the cumulative sum of the spectrum between χ_1 and χ_2, the Nyquist rate. Hence, a^* and b^*, initial value for a and b, are obtained from

$$a^* = (1-v)\mathrm{V}_y \,,\; b^* = v/(1-v) \quad ; \quad v = \mathrm{V}_n / \mathrm{V}_y$$

3) Let $C(\theta)=aP(d,b)$ in (1). The negative log-likelihood function becomes

$$L(\theta) = \tfrac{1}{2}\log a + \tfrac{1}{2}\log|P(d,b)| + \tfrac{1}{2}a^{-1}\mathbf{Y}^T P(d,b)^{-1}\mathbf{Y}$$

The length-scale hyperparameter, a, can be explicitly eliminated from $L(\theta)$ by substituting the minimising value of a as a function of d and b. The log likelihood function is thus reformulated to be dependent on only two hyper-parameters, d and b, instead of three, viz.,

$$\widetilde{L}(d,b) = \log|P(d,b)| + N\log\left[\mathbf{Y}^{\mathrm{T}} P(d,b)^{-1}\mathbf{Y}\right] \quad (6)$$

with

$$a = \mathbf{Y}^T P(d,b)^{-1}\mathbf{Y}/N \quad (7)$$

The initial value, d^*, for the length-scale hyperparameter, d, is obtained by solving the nonlinear equation $a^* = \mathbf{Y}^T P(d^*,b^*)^{-1}\mathbf{Y}/N$.

The hyperparameter values a^*, d^* and b^*, obtained by *procedure 2* are appropriate initial values for minimising the log likelihood, whether (1) or (6). The latter has the advantage of only being dependent on two hyperparameters and so converges more quickly. There are two cases. In the first, all the hyperparameters are adjusted during the optimisation to converge on a nearby local minimum corresponding to the prior model with the required length-scale characteristic. In the second, the optimisation may fail to locate a suitable local minimum, when all the hyperparameters are adjusted. In the latter situation, d^* needs to be held constant during the optimisation. It may then be necessary to adjust manually the value of d^* and repeat the optimisation to obtain the prior model with the required length-scale characteristic.

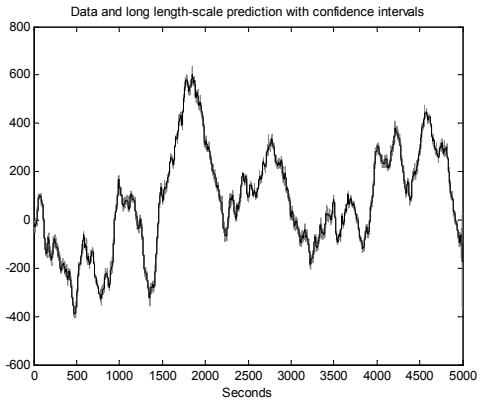

Fig. 2. Data and long length-scale prediction with confidence intervals.

6. APPLICATION TO DATA WITH GAPS

In this section, by exploiting the procedures presented in Sections 4 and 5, the Gaussian regression methodology of Sections 2 and 3 are applied to a data set of 5,000 points sampled at 1Hz (CATS Benchmark, 2004). It contains four gaps, specifically, the intervals, (981s-1000s), (1981s-2000s), (2981s-3000s), (3981s-4000s) and (4981s-5000s). When applying Gaussian regression to data with gaps, depending on the context, a single Gaussian process model with covariance function having more than one term or a multiple Gaussian process model may be required (Leithead *et al* 2005b). The data is shown in Fig. 2 (the grey line) together with its spectral density function in Fig. 3 (the grey line). The data has a component with length-scale longer than the gaps at frequencies less than 0.045Hz, a component with length-scale similar the gaps at frequencies between 0.045Hz and 0.095Hz and a component with length-scale shorter than the gaps above 0.096Hz. Here, to extract each of the above components separately, a multiple Gaussian process model with three processes is employed.

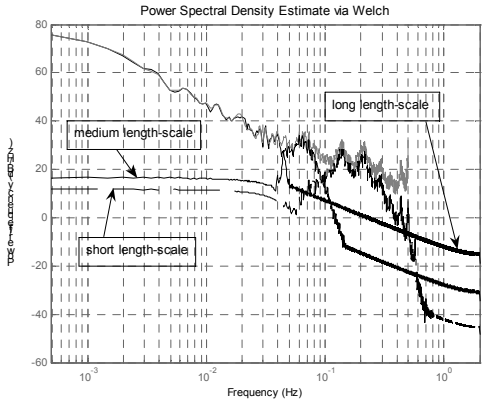

Fig. 3. Power Spectra of the three components.

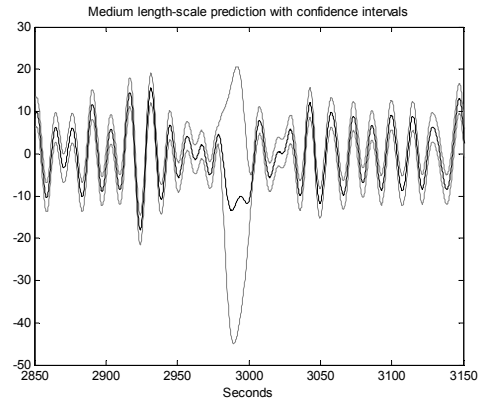

Fig. 4. Medium length-scale component prediction with confidence intervals.

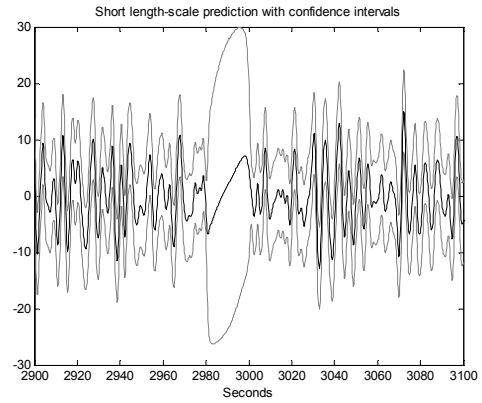

Fig. 5. Short length-scale prediction with confidence intervals.

The values of the hyperparameters, a and d, for the long, medium and short length-scale components are determined sequentially, the first from the data, the

second from the residues of the long length-scale prediction and the third from the residues of the combined long and medium length-scale prediction. The fast and memory efficient generalised Schur algorithm of Section 4 is used together with initialisation algorithm of Section 5. The hyperparameter values for the three components are a=2.422x10^4 & d=0.0038, a=83.6391 & d=0.0473 and a=55.962 & d=1.274, respectively. The noise hyperparameter value is n=35.9130. The long length-scale component prediction and confidence intervals are shown in Fig. 2 (black lines). A typical section, from 2850s to 3150s, of the medium length-scale component prediction and confidence intervals are shown in Fig. 4 and a typical short section, from 2900s to 3100s, of the short length-scale component prediction and confidence intervals in Fig. 5. The spectral density functions for the three components are depicted in Fig. 3. In addition, the differences between the data and the predictions from the complete three Gaussian process model together with the confidence interval (grey lines) are shown in Fig. 6. As would be expected, the confidence interval is much wider during the gaps.

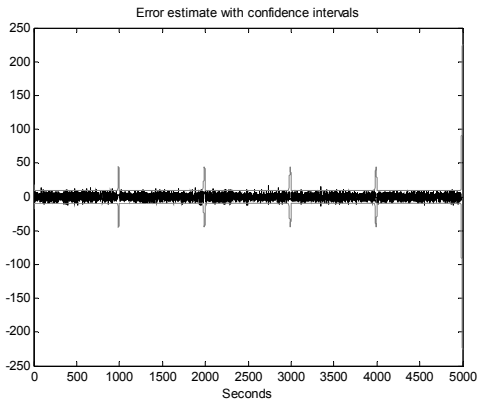

Fig. 6. Complete model: difference between prediction and data with confidence intervals.

7. CONCLUSION

The application of Gaussian regression to time-series data analysis is examined. Through the use of models consisting of multiple independent Gaussian processes, the general methodology is extended such that the individual Gaussian processes are conditioned on the data subject to the condition that they remain independent. When applied to time-series data, separate components with different frequency scales can be extracted.

Fast and memory-efficient methods for the matrix manipulations required by the Gaussian regression methodology are discussed. A procedure to determine the Shur decomposition of Toeplitz-like matrices, a key issue, is presented. A procedure to ensure appropriate initialisation, when training the prior model, is also presented.

A multiple Gaussian process model is applied, using the above procedures, to extract separate components with different frequency scales from a 5,000-point time-series data set with gaps. The effectiveness of the methods is clear.

ACKNOWLEDGEMENTS

This work was supported by Science Foundation Ireland grant 00/PI.1/C067.

REFERENCES

CATS Benchmark (2004). Time Series Prediction Competition, *Int Joint Conf on Neurel Networks*, http://www.cis.hut.fi/~lendasse/competition/competition.html.

Kailath, T. and A. H. Sayed (1999). Fast reliable algorithms for matrices with structure. In: *Society for Industrial and Applied Mathematics (SIAM)*.

Gibbs, M. N and D. J. C. Mackay (2000). Variational Gaussian process classifiers. In: *IEEE Transactions on Neural Networks*, vol. 11, page 1458-1464.

Leithead, W. E., Yunong Zhang and Kian Seng Neo (2005a). Wind turbine rotor acceleration: Identification using Gaussian regression. In: *2nd International Conference on Informatics in Control, Automation and Robotics*. Barcelona, Spain.

Leithead, W. E., Kian Seng Neo and D. J. Leith (2005b). Gaussian regression based on models with two stochastic processes. In: *16th World Congress*. IFAC 2005, Prague.

Mackay, D. J. C. (1998). Introduction to Gaussian processes. In: *Neural Networks and Machine Learning*, F: Computer and Systems Sciences (Bishop, C. M. (Ed)), vol. 168, page 133-165, Springer. Berlin, Heidelberg.

Williams, C. K. I. (1999). Prediction with Gaussian processes: from linear regression to linear prediction and beyond. In: *Learning in Graphical Models* (Jordan, M. I. (Ed)), page 599-621.

Yoshioka, T. and S. Ishii (2001). Fast Gaussian process regression using representative data. In: *Proceedings of International Joint Conference on Neural Networks*, vol. 11, page 132-137.

Yunong Zhang, Leithead, W. E. and Leith, D. J. (2005). Time-series Gaussian process regression based on Toeplitz computation of $O(N^2)$ operations and $O(N)$-level storage. In: *Proceedings of IEEE Conference on Decision and Control and the European Control Conference*. Seville, Spain.