Products and Sums of Tree-Structured Gaussian Processes

Christopher K. I. Williams and Stephen N. Felderhof
Division of Informatics
University of Edinburgh
5 Forrest Hill, Edinburgh EH1 2QL
{ckiw,stephenf}@dai.ed.ac.uk
http://anc.ed.ac.uk

Abstract

Recently Hinton (1999) has introduced the Products of Experts (PoE) model. In this paper we consider a PoE model in which each expert is a Gaussian, giving rise to a product model that is also Gaussian. However, if we constrain each expert to be a tree-structured Gaussian process (TSGP) the product of these then has a more complex structure than the individual trees. By way of comparison we also consider the framework within which the resultant process is constructed from the *sum* of tree-structured Gaussian processes. The result of this method is also a Gaussian process. We investigate the approximation of various target stationary processes with these Product of Experts and Sum of Experts models. Our results show that the preferred choice between the two models depends on the type of target process. We also show that for AR(1) and MA(2) target processes, an exact representation of these processes using only two component TSGPs can be found.

Recently Hinton (1999) has introduced the Products of Experts (PoE) model in which several individual probabilistic models for data are combined to provide an overall model of the data. In this paper we consider a PoE model in which each expert is a Gaussian. It is easy to see that in this case the product model will also be Gaussian. However, if each Gaussian has a simple structure, the product can have a richer structure. Below we constrain each expert to be a tree-structured Gaussian process (TSGP), for which efficient inference can be carried out in time linear in the number of leaves. Our aim is then to investigate how products of TSGPs can be used to approximate more general (non tree-structured) Gaussian processes. Such product-of-Gaussians models may be useful in applications such as image analysis, where it is not feasible to deal with the very large matrix calculations that might otherwise be involved.

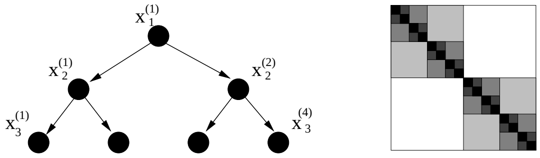
Using Gaussian experts is attractive as it permits a thorough analysis of the product architecture, which can be difficult with other models, e.g. models defined over discrete random variables.

An alternative way of generating more general Gaussian processes from component TSGPs is to consider the process which arises from summing the data generated by each of the individual TSGPs. It is interesting to compare the types of stochastic process which can be effectively modelled by each approach.

The structure of the paper is as follows. In section 1 we discuss the theory of tree-structured Gaussian processes (TSGPs), products of TSGPs and sums of TSGPs. In section 2 we present the results of experiments of matching the products and sums of TSGPs to target stochastic processes. Section 3 provides an analysis of why an exact representation using only two component TSGPs can be found for AR(1) and MA(2) target processes. We conclude with a discussion of the issues raised.

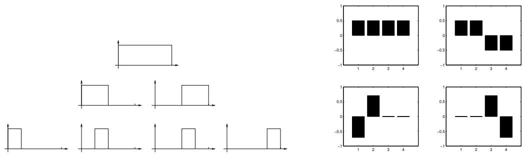
1. Theory

We consider in turn a single tree-structured Gaussian process (TSGP), the product of permutations of TSGPs, the sums of permutations of TSGPs and the fitting of products and sums of TSGPs to target processes.



(a) (b)

Figure 1. (a) A 3-layered Tree-Structured Gaussian Process. (b) A schematic representation of the covariance matrix for the leaves of a 5-layer tree. Areas of constant shading represent equal covariance. Since the covariances are all positive, the darker shades of grey represent higher covariance.



(a) Tree basis (b) Eigenfunctions

Figure 2. The Tree Basis Functions and the Eigenfunctions of a TSGP with 4 leaf nodes..

1.1 Tree-Structured Gaussian Processes

Consider m nodes with indices $1, \ldots, m$. A Gaussian process over these nodes is defined by the mean vector (which without loss of generality we take to be zero) and the covariance matrix. Tree-structured Gaussian processes are a subset of Gaussian process in which the distribution over the m leaf nodes is constructed by summing independent random variables organised in a tree-like manner, as illustrated in Figure 1(a). Below we consider one-dimensional TSGPs, although it is easy to extend this to higher dimensions; for example, in two dimensions a quad-tree branching structure can be used.

To sample from a TSGP we start at the root node and work downwards, using the equation

$$x_i^{(j)} = a x_{i-1}^{Pa(j,i)} + b_i w_i^{(j)}. \tag{1}$$

Here $x_i^{(j)}$ denotes the jth node in the ith layer, $Pa(j,i)$ is a function which returns the index of the parent of the node $x_i^{(j)}$, a and $\{b_i\}$ are constants, and the w's are a set of independent $N(0,1)$ Gaussian random variables. The root node takes the value $x_1^{(1)} = b_1 w_1^{(1)}$. The process that we are interested in is $\mathbf{x}_L$, the set of x variables at level L. By repeatedly applying equation 1 we find that

$$x_L^{(j)} = \sum_{i=1}^{L} a^{L-i} b_i w_i^{Pa^{L-i}(j,L)}, \tag{2}$$

where $Pa^k(j,L)$ is a function that returns the value of the kth ancestor of the node $x_L^{(j)}$. The variance of a leaf node is easily calculated as

$$var(x_L^{(j)}) = \sum_{i=1}^{L} a^{2(L-i)} b_i^2 \stackrel{def}{=} \sum_{i=1}^{L} \sigma_i^2, \tag{3}$$

using the reparameterization $\sigma_i = a^{L-i} b_i$. The covariance of any two leaf nodes turns out to be the variance of their common parent. Note that allowing extra freedom in the parameterization of a (e.g. allowing a to depend on the level i) does not increase the flexibility of the TSGP as it is σ_i which defines the covariance structure. The form of the covariance matrix for the tree-structured process is schematically represented in Figure 1b. As there is a σ_i parameter for every layer there are $\log_2 m + 1$ parameters for a binary tree. Since σ_i^2 is added at the ith node, the every entry in the TSGP covariance matrix is constrained to be non-negative.

Tree-structured Gaussian processes have been studied by a number of authors including the group led by Prof. Willsky at MIT [2, 6]. In [6] a more constrained type of hierarchical stochastic process is considered in which the b_js decay exponentially: $b_j = 2^{-\delta j/2}$, (δ is a constant), leaving only two free parameters.

It is also useful to consider relaxing the form of equation 1. For example different values of a might be used for the left and right branches at each non-leaf node; this kind of construction is discussed in section 3. It is also possible to generalize equation 1 so that $x_i^{(j)}$ is vector valued, and a and the b's then become matrices (see e.g. [2]).

1.1.1 Basis function representation

An equivalent way of constructing this process is to consider set of basis functions as shown in figure 2(a). These are scaled and translated "box-car" functions. In this representation we write the 1-d process $x_L(t)$ as

$$x_L(t) = \sum_{k=1}^{2m-1} \beta_k \psi_k(t)$$

where ψ_i are the basis functions illustrated and β_i are zero-mean gaussian random variables with $\beta_k \sim N(0, \sigma^2_{l(k)})$, where $l(k)$ is the function that returns the layer to which basis function k corresponds. Here we regard the discrete process as a subsampling of the continuous process at regular points along the line. We are not aware of previous work taking this view of the TSGP.

Note that the basis functions are not orthogonal. However, if we define the covariance function $K(t,t') = \sum_k \sigma^2_{l(k)} \psi_k(t)\psi_k(t')$, we can calculate the eigenfunctions of this kernel $\phi_i(t)$ as linear combinations of the basis functions, i.e. $\phi_i(t) = \sum_j \gamma^i_j \psi_j(t)$, giving rise to a matrix eigenproblem to determine the γs. For the 4-node example of Figure 2(a) the four eigenfunctions with non-zero eigenvalues are plotted in Figure 2(b).[1]

The multiscale nature of the basis function decomposition is reminiscent of wavelets; see e.g. [8] for further discussion/analysis of this kind of construction.

1.2 Products of Tree-Structured Gaussian Processes

The process $\mathbf{x}_L$ at the leaf nodes of a single Gaussian tree is a non-stationary process, i.e. the covariance between $x^{(i)}_L$ and $x^{(j)}_L$ does not depend only on $|i-j|$. Here we will consider only cyclic permutations which involve moving the whole tree-structure relative to the nodes, wrapping the structure around at the edges. We will not consider more general permutations which would involve changing the ordering of the nodes with respect to one another.

Let $\mathbf{P}_i$ denote a cyclic permutation matrix, (a matrix of zeros with a single "1" in each row and column), such that $\mathbf{P}_i\mathbf{y}$ is the ith cyclic permutation of vector $\mathbf{y}$. (The 0th permutation is the identity). If $\mathbf{C}_0$ denotes the unpermuted covariance matrix, the covariance of $\mathbf{P}_i\mathbf{y}$ is $\mathbf{C}_i = \mathbf{P}_i\mathbf{C}_0\mathbf{P}^T_i$.

Let the product of n zero-mean Gaussians have covariance matrix $\mathbf{C}_{model}$. We then have $\mathbf{C}^{-1}_{model} = \sum_{i=0}^{n-1} \mathbf{C}^{-1}_i$. Note that if $\mathbf{x}_L$ has length m, then the product of all m cyclic permutations of a TSGP will give rise to a stationary Gaussian process with a Toeplitz covariance matrix.

1.3 Sums of Tree-Structured Gaussian Processes

In this construction, the process that the model generates, $\mathbf{x}$, is formed by summing permuted TSGPs, $\mathbf{x}^{(i)}_L$. In other words, $\mathbf{x} = \sum_i \mathbf{x}^{(i)}_L$. The overall covariance matrix is the sum of the covariance matrices of all the component processes[2], $\mathbf{C}_{model} = \sum_i \mathbf{C}_i$. A special case of this technique is the cyclostationary process, described in [7] (section 11-4). The cyclostationary construction with a discrete process of period m would produce an overall covariance matrix which is the average of m shifted copies of the $\mathbf{C}_0$.

Notice that for the products of TSGPs (PTSGPs) we multiply the probability distributions of the component processes, $P(\mathbf{x}_L) = \prod_i P(\mathbf{x}^{(i)}_L)$. The overall process is a conjunctive representation of its experts. A particular data pattern $\mathbf{x}_L$ only has a high probability if it has high probability under each of the permuted TSGPs. On the other hand, the Sum of Tree-Structured Gaussian Processes (STSGP) is generated by adding together the component processes. In this case the covariance of the model is the sum of the permuted covariance matrices whereas for the PTSGP, the *inverse* covariance of the model is the sum of the permutations of the *inverse* covariance matrix of the TSGP.

1.4 Target Processes

As we have indicated above, a problem with a single TSGP is its non-stationarity. Our aim will be to find out how well products of TSGPs can model a stationary target process. Our target processes are (a) first order autoregressive processes, AR(1), (b) second order moving average processes, MA(2), (c) second order autoregressive processes, AR(2). We construct each of these processes on a circle. If there are m discrete points in the process, then the mth point wraps around to be adjacent to the first point. For AR(1), the value of the process at the ith point $X_i = \alpha X_{\mathrm{mod(i-1,m)}} + Z_i$ where Z_i is a zero mean Gaussian random variable with variance v and $i \in \{0,\ldots,m-1\}$. In this case the inverse covariance matrix has a tridiagonal circulant structure. The diagonal elements have value $(1+\alpha^2)/v$ and the off-diagonal elements are $-\alpha/v$. Similarly for the AR(2) process, $X_i = \beta X_{\mathrm{mod(i-1,m)}} + \alpha X_{\mathrm{mod(i-2,m)}} + Z_i$. Now the inverse covariance matrix has a pentadiagonal circulant structure and reading from the central diagonal outwards the lines have values $(1+\alpha^2+\beta^2)/v$, $(\alpha\beta-\beta)/v$, $-\alpha/v$. A general MA(l) process on a discrete lattice is a linear filter applied to $\{Z_i\}$, of the form $X_i = \sum_{j=0}^{l-1} \beta_j Z_{\mathrm{mod(i-j,m)}}$ and has covariance $\gamma(k) = \sum_{i=0}^{l-k} \beta_i\beta_{i+k}$ for $k = 0,\ldots,l$ and zero for $k > l$ [1].

1.5 Fitting Products and Sums of TSGPs to Target Processes

Through the construction of either the PTSGP or the STSGP we arrive at the covariance matrix for the model $\mathbf{C}_M$, which depends on the $\{\sigma_i\}$ parameters.

---

[1] The parameters for this tree were $\sigma^2_1 = 1.0, \sigma^2_2 = 0.5, \sigma^2_3 = 0.25$.

[2] $\mathbf{C}_{model} = \left\langle (\mathbf{x}^{(1)}_L + \ldots + \mathbf{x}^{(n)}_L)(\mathbf{x}^{(1)}_L + \ldots + \mathbf{x}^{(n)}_L)^{\mathbf{T}} \right\rangle = \sum_{i=1}^{n} \left\langle \mathbf{x}^{(i)}_L \mathbf{x}^{(i)\mathbf{T}}_L \right\rangle$

(We assume that all experts use the same parameters.) Our task is now to fit this to the target process with covariance matrix $\mathbf{C}_T$. A natural measure of discrepancy is the Kullback-Leibler divergence between the two distributions $\int P_T(\mathbf{x})[\log P_T(\mathbf{x}) - \log P_M(\mathbf{x})]d\mathbf{x}$. For the special case of two zero-mean Gaussians this is given in [5]:

$$D_{KL} = \frac{1}{2}\log\frac{|\mathbf{C}_M|}{|\mathbf{C}_T|} + \frac{1}{2}\mathrm{tr}\mathbf{C}_T(\mathbf{C}_M^{-1} - \mathbf{C}_T^{-1}). \quad (4)$$

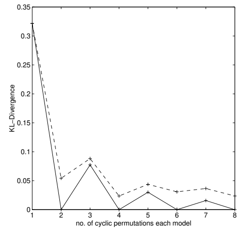
For convenience a simplex search (the `fmins` function in MATLAB) was used to carry out this optimization, although more sophisticated gradient-based methods could also be used. The possibility of local optima in the search space was addressed by initializing the search several times using different randomly-chosen starting points for several target processes. From this we concluded that there do not seem to be significant local optima problems in the search, so in the experiments below only one optimization is carried out per experimental configuration.

2. Experimental Results

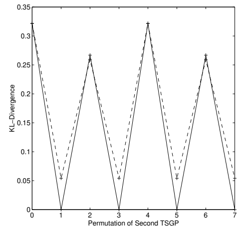
We have conducted experiments to investigate how well the PTSGP and STSGP models can be used to fit the target processes.

The first question to ask is: how many permutations of the single TSGP do we need to model the target processes effectively and which permutations are the best ones to choose? We have tried a large number of possibilities using different ways of picking the permutations to be included in the model, and different target processes with different parameters. Figure 3 is a representative sample of the results. The rest of the results, whilst not identical, follow a qualitatively similar pattern. The figure shows the results of trying to fit PTSGPs and STSGPs to the AR(1) target process with $\alpha = 0.5$ and $m = 8$.

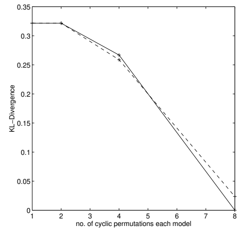
Firstly note, that having just a single TSGP is the worst fitting model and that combining one or more permutations of the TSGP either in the STSGP or PTSGP frameworks always reduces the KL-divergence between target and model (except when the TSGP which is shifted halfway round the cirle so that it has the same covariance structure as the original TSGP, then using just this and the original TSGP in a model gives the model no extra flexibility and KL-divergence doesn't decrease). The use of all of the possible permutations, in either the PTSGP or the STSGP, produces the best fitting model possible. This may also be achieved with a smaller number of permutations. For the AR(1) target, the PTSGP with two TSGPs ($\mathbf{C}_0$ and $\mathbf{C}_1$) fits the target exactly, i.e. the KL-divergence is zero. On the other hand, the STSGP isn't able to fit the target exactly with any number of permutations. However, using just two permutations in the STSGP allows it to do nearly as well as the best it does which is when it uses all the possible permutations. With these considerations in mind, we will subsequently only present results for the cases where (i) only the single TSGP has been used (ii) two TSGPs have been combined (the original TSGP and a TSGP shifted by one) and finally (iii) all the possible TSGP permutations have been combined.



(a)



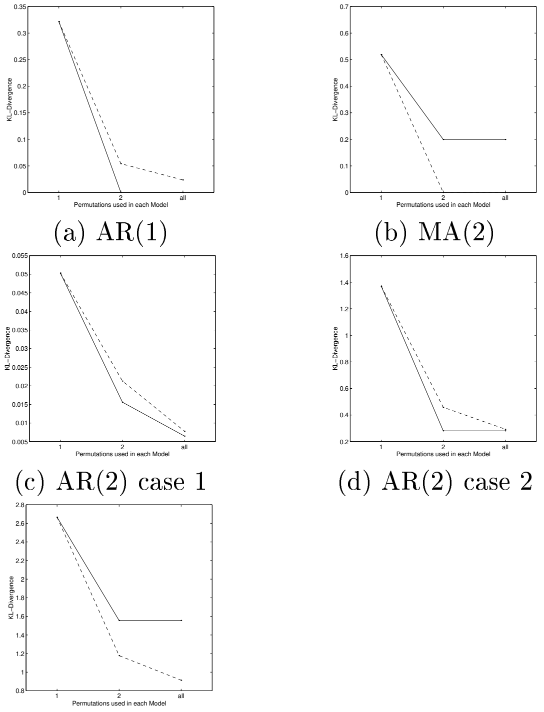
(b)



(c)

Figure 3. Combining the permuted TSGPs. These graphs show the KL-divergence between the target process (in this case AR(1)) and the model process which is either the sum (dotted line) or product (solid line) of TSGPs. Each graph shows a different way that one might use to choose the permutations of the TSGP to include in the sum/product: (a) add extra permutations one at a time, the new permutation added is adjacent to the last one added ie $\{\mathbf{C}_0\}$, $\{\mathbf{C}_0, \mathbf{C}_1\}$, $\{\mathbf{C}_0, \mathbf{C}_1, \mathbf{C}_2\}$, etc; (b) the unpermuted TSGP and one other at each different shifted position, $\{\mathbf{C}_0, \mathbf{C}_0\}$, $\{\mathbf{C}_0, \mathbf{C}_1\}$, $\{\mathbf{C}_0, \mathbf{C}_2\}$, etc; (c) doubling the number of TSGPs keeping them evenly spaced around the circle, $\{\mathbf{C}_0\}$, $\{\mathbf{C}_0, \mathbf{C}_4\}$, $\{\mathbf{C}_0, \mathbf{C}_2, \mathbf{C}_4, \mathbf{C}_6\}$, etc.

Figure 4 shows plots of the KL-divergence between the selected target processes and the PTSGPs and STSGPs each with one, two and all permutations as just described, for $m = 64$. Fig. 4(a) shows again that PTSGP can exactly fit an AR(1) process with just two permutations and that the STSGP can't with any number. Fig. 4(b) shows that the STSGP can exactly match an MA(2) process with just two permutations and in this case the PTSGP doesn't



(a) AR(1) (b) MA(2) (c) AR(2) case 1 (d) AR(2) case 2 (e) AR(2) case3

Figure 4. Fitting the models to target processes. The graphs show the KL-divergence between the fitted models (PTSGP solid line, STSGP dotted line) and various different target processes. The three points on each line represent the KL-divergence for the cases where in each model there is: just a single TSGP; the single TSGP and a TSGP which has been shifted by one; and finally where there are all the possible permutations included in the sum/product. In (a) the target process is AR(1), (b) MA(2) with $\beta_0 = \beta_1 = 1$, (c) AR(2) with $\alpha = 0.1$, $\beta = 0.2$, (d) AR(2) with $\alpha = -0.25$, $\beta = 1$, (e) AR(2) with $\alpha = -0.5$, $\beta = 1$. In these experiments, the target and model processes have 64 discrete points corresponding to tree-structures with 7 layers.

have sufficient flexibility to fit the target. Fig. 4(c-e) show the performance of the two approaches when the target process is AR(2). The different cases correspond to different parameter values[3] α and β. We can see that although the different methods perform best in different settings, neither actually manage to reduce the KL-divergence to zero.

[3] Analysis in [3] suggests that we might expect different regimes of behaviour depending on the values of α and β. For the AR(2) process, we solve the equation for the covariance, $\gamma_k = \beta\gamma_{k-1} + \alpha\gamma_{k-2}$ by assuming a solution of the form $\gamma_k = \lambda^k$ which gives a quadratic in λ. Whether $\beta^2 - 4\alpha$ is less than, equal to or greater than zero affects the solution. Note that the specific values that we use for each case are the same as those used in the examples in [3].

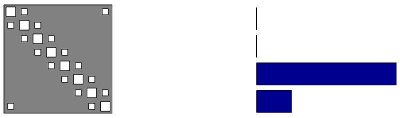
Figure 5(a) shows the structure of the inverse covariance matrix of the product of two TSGPs which have been adjusted so that the product fits the AR(1) target (see figure 6(c)) and similarly figure 5(b) shows the structure of the covariance matrix of the sum of two TSGPs which have been adjusted so that the sum fits the MA(2) target process (figure 5(d)). Notice that in both cases the fitted TSGPs only have covariance in the bottom two layers of the tree (see the right panels of figures 5(a) and (b)). This can be understood by the inspection of figure 6. Figure 6(b) shows the structure of the inverse covariance matrix of a TSGP which only has covariance in the bottom two layers. Using the products construction, we can see that if we add the values in this matrix to those in an identical matrix which has been permuted by one then we will be able to get a inverse covariance matrix in the form of the target AR(1) process. Likewise figure 6(a) shows the structure of the covariance matrix of a TSGP which only has covariance in the bottom two layers. With the summation construction, if we add the values in this matrix to those in an identical matrix but which has been permuted by one then we will be able to get a covariance matrix in the form of the target MA(2) process. This works because we know the AR(1) process is defined by the distance-two interactions in the inverse covariance matrix and similarly the MA(2) process is defined by distance-two interactions in the covariance matrix. The above construction allows all the appropriate distance-two interactions to be generated and the processes are fitted exactly. The individual TSGPs required to achieve the fit have all their variance in the bottom two layers of the tree corresponding to the distance-two interactions. In section 3 below we analyse the nature of the exact fit of the TSGPs to the AR(1) and MA(2) processes.

3. Characterization of AR(1) and MA(2) processes as sums and products of TSGPs

The results in section 2 show that the AR(1) process can be fitted exactly with a product of two TSGPs, and similarly that a MA(2) process can be fitted exactly as a sum of two TSGPs. Below we analyse this in more detail.

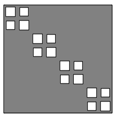
For both the AR(1) and MA(2) cases, Figure 5 shows that the variance is only non-zero at the leaves and for the layer above. This means that each "individual" tree is actually a forest of $m/2$ trees each of height 1. The covariance matrix for the leaves of a

(a) Inverse PTSGP fitted to AR(1) process

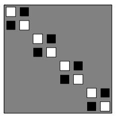


(b) STSGP fitted to MA(2) process

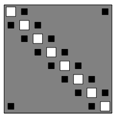
Figure 5. Results for the target processes and models which fit exactly. Subfigure (a) shows how the PTSGP can be used to generate an AR(1) process and subfigure (b) shows how the STSGP can be used to generate an MA(2) process. Only two TSGPs are included in each model: the unpermuted TSGP and the TSGP which has been shifted by one. Within subfigure (a) the plot on the left shows a Hinton diagram of the structure of the inverse PTSGP covariance matrix (where the magnitude of an entry is represented by the area of the box, white represents positive values and black represents negative values), and the plot on the right shows the variance σ_i^2 added at each level of the single TSGP which is used in the model. Subfigure (b) shows the corresponding results for fitting the STSGP to the MA(2) process except the left plot shows the structure of the covariance rather than the inverse covariance matrix.



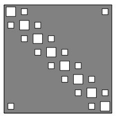
(a) TSGP



(b) Inverse TSGP



(c) Inverse AR(1)



(d) MA(2)

Figure 6. Hinton diagrams showing the structure of the covariance and inverse covariance matrices for a single TSGP, the inverse AR(1) covariance and the MA(2) covariance. The TSGP is shown with variance due to only the last two layers of the tree. For representational convenience, processes with only 8 discrete points are shown.

height-1 tree has the form

$$\begin{pmatrix} \sigma_1^2+\sigma_2^2 & \sigma_2^2 \\ \sigma_2^2 & \sigma_1^2+\sigma_2^2 \end{pmatrix} \quad (5)$$

using the notation of equation 3 with $L = 2$. Thus the covariance matrix of the sum of two such TSGPs (with the second displaced by one leaf node from the first) will have a tridiagonal circulant structure with diagonal elements of $2\sigma_1^2 + 2\sigma_2^2$ and off-diagonal elements of σ_2^2. The general MA(2) process has a tridiagonal circulant covariance structure with $\beta_0^2+\beta_1^2$ on the diagonal and $\beta_0\beta_1$ off. If $\beta_0\beta_1 > 0$ then we find the solution $\sigma_2^2 = \beta_0\beta_1$ and $\sigma_1^2 = (\beta_0-\beta_1)^2/2$. If $\beta_0\beta_1 < 0$ then we need a modification of the parameterization of the tree to allow a negative covariance. If $x_2^{(1)}$ and $x_2^{(2)}$ are the leaves of the height-1 tree, and $x_1^{(1)}$ is the root, then letting

$$x_2^{(1)} = a x_1^{(1)} + b_1 w_1^{(1)}, \quad (6)$$

$$x_2^{(2)} = -a x_1^{(1)} + b_1 w_1^{(2)}, \quad (7)$$

(note the $-$ sign in the second equation) we find that their covariance matrix now has the same form as in equation 5 but with the off-diagonal entry now being $-\sigma_2^2$. With this modification we can obtain a match of the sum of two TSGPs to a general MA(2) process.

For the AR(1) process, we need to consider the inverse covariance matrices. The inverse of (5) has the form

$$\frac{1}{\Delta}\begin{pmatrix} \sigma_1^2+\sigma_2^2 & -\sigma_2^2 \\ -\sigma_2^2 & \sigma_1^2+\sigma_2^2 \end{pmatrix} \quad (8)$$

where $\Delta = \sigma_1^2(\sigma_1^2+2\sigma_2^2)$. Thus the overall inverse covariance of a product of two such TSGPs (with the second displaced by one leaf node from the first) will have tridiagonal circulant structure with diagonal elements of $2(\sigma_1^2+\sigma_2^2)/\Delta$ and off-diagonal elements of $-\sigma_2^2/\Delta$. For a general AR(1) process (with noise variance $Z_i \sim N(0, v)$ for all i), the diagonal entries in the inverse covariance are $(1+\alpha^2)/v$ and the non-zero off-diagonal entries are $-\alpha/v$. By equating these terms we obtain

$$\frac{1+\alpha^2}{v} = \frac{2}{\Delta}(\sigma_1^2+\sigma_2^2), \quad (9)$$

$$\frac{\alpha}{v} = \frac{\sigma_2^2}{\Delta}. \quad (10)$$

By dividing (9) by (10) we obtain $\sigma_1^2 = \sigma_2^2\frac{(1-\alpha)^2}{2\alpha}$. Substituting this back into (10) we obtain

$$\sigma_2^2 = \frac{4\alpha v}{(1-\alpha)^2(1+\alpha)^2}. \quad (11)$$

Thus for a given values of v and $\alpha > 0$ we can derive the parameters for the equivalent PTSGP. When $\alpha < 0$ the same modification is needed as for the MA(2) process with $\beta_0\beta_1 < 0$.

We are currently working on extending this analysis for higher-order AR and MA processes. For example we conjecture that a product of four TSGPs will be equivalent to a AR(3) process, and that a sum of four TSGPs will be equivalent to a MA(4) process.

4. Discussion

Above we have fitted products and sums of TSGP models to a number of target processes by minimizing the KL divergence criterion. For the AR(1) process an exact representation of the target process was achieved using PTSGPs, and for the MA(2) process using STSGPs. In both cases we only needed two TSGPs in the model to achieve the exact match.

Stationary processes on the circle have sinusoidal eigenfunctions so in general a stationary 1-d process on m nodes needs $m/2 + 1$ Fourier coefficients to describe it. (The sine and cosine components have the same variance, but there is no $\sin 0t$ component to match the $\cos 0t = 1$ basis function). With the current tied parameterization, a sum or product of TSGPs has only $\log_2 m + 1$ free parameters. Thus there are some target processes which cannot be exactly matched using either approach. The answer to this may be to add some extra flexibility to the models' parameter structure. Examples of extra flexibility include (i) increasing the number of parameters in a given tree (see, e.g. section 3), (ii) using vector rather than scalar valued processes in the individual TSGPs and (iii) removing the tying of parameters across TSGPs; if only a small number of permutations were used this would not lead to an explosion in the number of parameters to be fitted.

Acknowledgements

This work is supported by EPSRC grant GR/L78161 *Probabilistic Models for Sequences*. SNF gratefully acknowledges additional support from BAE Systems. We thank Geoff Hinton for helpful conversations.

References

[1] C. Chatfield. *The Analysis of Time Series: An Introduction*. Chapman and Hall, London, 4th edition, 1989.

[2] K.C. Chou, A.S. Willsky, and A. Benveniste. Multiscale Recursive Estimation Data Fusion, and Regularization. *IEEE Trans. on Automatic Control*, 39(3):464–477, March 1994.

[3] Peter J. Diggle. *Time Series: A Biostatistical Introduction*. Oxford University Press, Oxford, 1990.

[4] G. E. Hinton. Products of Experts. In *Proceedings of the Ninth International Conference on Artificial Neural Networks*, volume 1, pages 1–6, 1999.

[5] S. Kullback. *Information Theory and Statistics*. Dover, New York, 1959.

[6] M.R. Luettgen, W.C. Karl, and A.S. Willsky. Multiscale Representations of Markov Random Fields. *IEEE Trans. Signal Processing*, 41(12):3377–3395, 1993.

[7] Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, Singapore, 1991.

[8] G. W. Wornell. A Karhunen-Loève-like Expansion for $1/f$ Processes via Wavelets. *IEEE Trans. Information Theory*, 36(4):859–861, 1990.