Sparse Multi-output Gaussian Processes

Mauricio Alvarez and Neil Lawrence
School of Computer Science
University of Manchester
{alvarezm, neill}@cs.man.ac.uk

Introduction

We consider the problem of modelling correlated outputs from a single Gaussian process (GP). Modelling multiple output variables is a challenge as we are required to compute cross covariances between the different outputs. These cross covariances allow us to improve our predictions of one output given the others as the correlations between outputs are modelled [1, 2, 3, 4].
One of the most straightforward methods used for this kind of modelling tasks employs a convolution process (CP) in which each output is expressed as the convolution between a smoothing kernel function and a Gaussian white noise [2]. If the white noise is a common input to all convolutions, then the outputs are going to be correlated and will have covariances and cross-covariances that are positive definite. However, the input function doesn't have to be white noise. In principle, it could be any Gaussian process.
Even though the CP framework is an elegant way for constructing dependent output processes, the fact that the full covariance function of the joint Gaussian process must be considered, results in significant computational demand and memory requirements when performing inference as the size of the covariance matrix now scales with the product of the number of outputs and the number of observation points.
In this work we propose a sparse approximation for the full covariance matrix involved in the multiple output convolution process. We exploit the fact that each of the outputs is conditional independent of all others given the input process. This leads to an approximation for the covariance matrix which keeps intact the covariances of each output and approximates the cross-covariances terms with a low rank matrix. It has a similar form to the Partially Independent Training Conditional (PITC) [5] approximation for a single output GP.

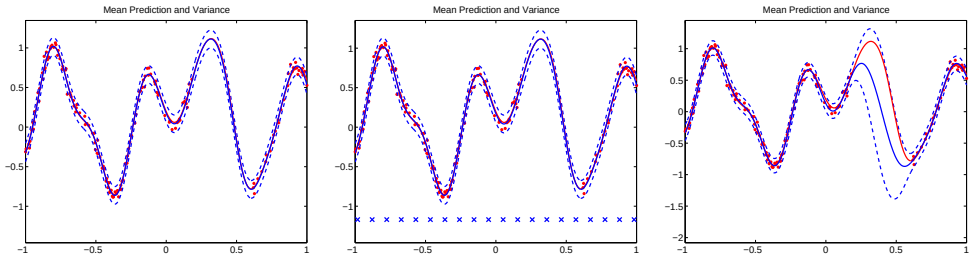
Method

Each output process $y_n(\mathbf{x})$ can be represented as the convolution between a smoothing kernel function $k_{nk}(\mathbf{x})$ and a latent function $u_k(\mathbf{x}')$. In general, $y_n(\mathbf{x}) = \sum_{k=1}^{K} \int_{-\infty}^{\infty} k_{nk}(\mathbf{x}-\mathbf{z}) u_k(\mathbf{z}) \mathrm{d}\mathbf{z} + w_n(\mathbf{x})$, where $w_n(\mathbf{x})$ is a white noise process and we have assumed zero mean for $y_n(\mathbf{x})$. If the latent functions $u_k(\mathbf{z})$ are independent Gaussian processes, the covariance and cross-covariances of the outputs are given by $\operatorname{cov}[y_n(\mathbf{x}), y_m(\mathbf{x}')] = \sum_{k=1}^{K} \int_{-\infty}^{\infty} k_{nk}(\mathbf{x}-\mathbf{z}) \int_{-\infty}^{\infty} k_{mk}(\mathbf{x}'-\mathbf{z}') \operatorname{cov}_{u_k u_k}(\mathbf{z}, \mathbf{z}') \mathrm{d}\mathbf{z}' \mathrm{d}\mathbf{z} + \sigma_n^2 \delta_{nm} \delta_{\mathbf{x},\mathbf{x}'}$, with a similar expression for the cross-covariance $\operatorname{cov}[y_n(\mathbf{x}), u_q(\mathbf{z}'))]$.
The covariance of the joint process, $\mathbf{K_{yy}}$, includes the covariances of each output and the cross-covariances between outputs, namely $\operatorname{cov}[y_n(\mathbf{x}), y_m(\mathbf{x}')]$. For prediction and optimization we require the inverse of this full covariance. The joint density between the output processes can be expressed as $p(\mathbf{y}|\mathbf{u}, \boldsymbol{\phi}) = \prod_{i=1}^{N} p(\mathbf{y}_i|\mathbf{u}, \boldsymbol{\phi})$, where $\boldsymbol{\phi}$ is the set of parameters of all the covariance matrices. Each density $p(\mathbf{y}_i|\mathbf{u}, \boldsymbol{\phi})$ is given as $p(\mathbf{y}_i|\mathbf{u}, \boldsymbol{\phi}) = \mathcal{N}\left(\mathbf{y}_i|\mathbf{K}_{\mathbf{y_i u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{u}, \mathbf{K}_{\mathbf{y_i y_i}} - \mathbf{K}_{\mathbf{y_i u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{uy_i}} + \sigma_i^2\mathbf{I}\right)$ and in this way, the conditional density of the full process is given as $p(\mathbf{y}|\mathbf{u}, \boldsymbol{\phi}) = \mathcal{N}\left(\mathbf{y}|\mathbf{K}_{\mathbf{yu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{u}, \mathbf{D}\right)$, where $\mathbf{D} = \operatorname{blockdiag}\left[\mathbf{K}_{\mathbf{yy}} + \sigma_i^2\mathbf{I} - \mathbf{K}_{\mathbf{yu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{uy}}\right]$. Assuming that the $\mathbf{u}_k$ are independent GP's, the marginalization of the latent functions leads to the evidence $p(\mathbf{y}|\boldsymbol{\phi}) = \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \mathbf{D} + \mathbf{K}_{\mathbf{yu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{uy}}\right)$.
In this manner, the full covariance $\mathbf{K_{yy}}$ has been approximated using $\mathbf{D} + \mathbf{K}_{\mathbf{yu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{uy}}$, which keeps the blocks of the diagonal the same as the original covariance and approximates the off-diagonal blocks with the low rank approximation $\mathbf{K}_{\mathbf{yu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{uy}}$.

Results

We use a Gaussian kernel as the smoothing kernel function. We first setup a toy problem in which we evaluate the speed of the approximation and the quality of the prediction. The toy problem consists of four outputs, one latent

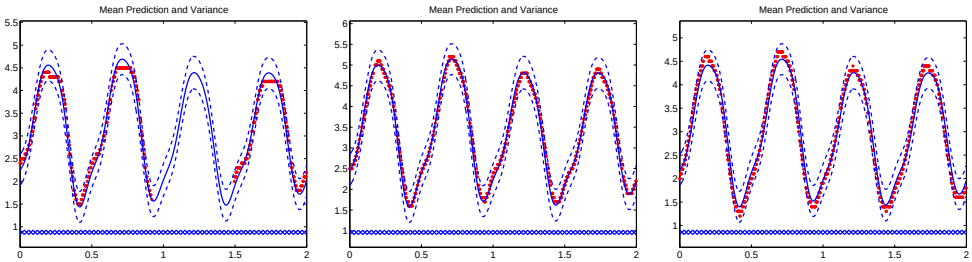
function and 200 observation points for each output. The training data was sampled from our model. We compare predictions from our sparse approximation with those from the full covariance (Figure 1).



(a) Output 4 with the full GP (b) Output 4 with the sparse GP (c) Output 4 with an independent GP

Figure 1: Predictive Mean and variance using the full multi-output GP, the sparse approximation and an independent GP for output 4.

The reduction in computational complexity of the approximation is from $O(N^3d^3)$ (N is number of data points and d is number of output variables) to $O(N^3d)$. This matches the computational complexity for modelling with independent Gaussian processes. However, with independent Gaussian processes the missing ranges from Figure 1 are not accurately captured. Figure 2 shows results over tide height signals obtained from a network of weather sensors located in the south coast of England (for more details of this database see [4]).



(a) Bramblemet Tide Height (b) Cambermet Tide Height (c) Chimet Tide Height

Figure 2: Predictive Mean and variance using the sparse approximation for the tide height signals.

Discussion

We have presented a sparse approximation for multiple output Gaussian processes, capturing the correlated information among outputs and reducing the amount of computational load for prediction and optimization purposes. Linear dynamical systems responses can be expressed as a convolution between the impulse response of the system with some input function. This convolution approach is an equivalent way of representing the behavior of the system through a differential equation. For systems involving high amounts of coupled differential equations [6], the approach presented here is a reasonable way of obtaining approximate solutions.

References

[1] D. Higdon. *Quantitative methods for current environmental issues*, chapter Space and space-time modelling using process convolutions, pages 37–56. Springer Verlag, 2002.

[2] P. Boyle and M. Frean. Dependent Gaussian processes. In Y. Weiss L. K. Saul and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17, pages 217–224. The MIT Press, 2005.

[3] Y.-W. Teh, M. Seeger, and M. Jordan. Semiparametric latent factor models. In *Workshop on AI and Statistics 10*, pages 333–340, 2005.

[4] A. Rogers, M. A. Osborne, S. D. Ramchurn, S. J. Roberts, and N. R. Jennings. Information agents for pervasive sensor networks. In *Proceedings of PerSens 2008*, 2008, (to appear).

[5] J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

[6] Pei Gao, Antti Honkela, Magnus Rattray, and Neil D. Lawrence. Gaussian Process Modelling of Latent Chemical Species: Applications to Inferring Transcription Factor Activities. Accepted for the European Conference on Computational Biology 2008 (ECCB08), 2008.