Efficient Multioutput Gaussian Processes through Variational Inducing Kernels

Mauricio A. Álvarez
School of Computer Science
University of Manchester
Manchester, UK, M13 9PL
alvarezm@cs.man.ac.uk

David Luengo
Depto. Teoría de Señal y Comunicaciones
Universidad Carlos III de Madrid
28911 Leganés, Spain
luengod@ieee.org

Michalis K. Titsias, Neil D. Lawrence
School of Computer Science
University of Manchester
Manchester, UK, M13 9PL
{mtitsias,neill}@cs.man.ac.uk

Abstract

Interest in multioutput kernel methods is increasing, whether under the guise of multitask learning, multisensor networks or structured output data. From the Gaussian process perspective a multioutput Mercer kernel is a covariance function over correlated output functions. One way of constructing such kernels is based on convolution processes (CP). A key problem for this approach is efficient inference. Álvarez and Lawrence recently presented a sparse approximation for CPs that enabled efficient inference. In this paper, we extend this work in two directions: we introduce the concept of variational inducing functions to handle potential non-smooth functions involved in the kernel CP construction and we consider an alternative approach to approximate inference based on variational methods, extending the work by Titsias (2009) to the multiple output case. We demonstrate our approaches on prediction of school marks, compiler performance and financial time series.

1 Introduction

In this paper we are interested in developing priors over multiple functions in a Gaussian processes (GP) framework. While such priors can be trivially specified by considering the functions to be independent, our focus is on priors which specify correlations between the functions. Most attempts to apply such priors (Teh *et al.*, 2005; Rogers *et al.*, 2008; Bonilla *et al.*, 2008) have focused on what is known in the geostatistics community as "linear model of coregionalization" (LMC) (Goovaerts, 1997). In these models the different outputs are assumed to be linear combinations of a set of one or more "latent functions". GP priors are placed, independently, over each of the latent functions inducing a correlated covariance function over the D outputs $\{f_d(\mathbf{x})\}_{d=1}^{D}$.

Appearing in Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9. Copyright 2010 by the authors.

We wish to go beyond the LMC framework, in particular, our focus is on *convolution processes* (CPs). Using CPs for multi-output GPs was proposed by Higdon (2002) and introduced to the machine learning audience by Boyle and Frean (2005). Convolution processes allow the integration of prior information from physical models, such as ordinary differential equations, into the covariance function. Álvarez *et al.* (2009a), inspired by Lawrence *et al.* (2007), have demonstrated how first and second order differential equations, as well as partial differential equations, can be accommodated in a covariance function. They interpret the set of latent functions as a set of *latent forces*, and they term the resulting models "latent force models" (LFM). The covariance functions for these models are derived through convolution processes. In the CP framework, output functions are generated by convolving R independent latent processes $\{u_r\}_{r=1}^{R}$ with smoothing kernel functions $G_{d,r}(\mathbf{x})$, for each output d and latent force r,

$$f_d(\mathbf{x}) = \sum_{r=1}^{R} \int_{\mathcal{Z}} G_{d,r}(\mathbf{x}-\mathbf{z})\, u_r(\mathbf{z})\, \mathrm{d}\mathbf{z}. \qquad (1)$$

The LMC can be seen as a particular case of the CP, in which the kernel functions $G_{d,r}(\mathbf{x})$ correspond to a scaled Dirac δ-function $G_{d,r}(\mathbf{x}-\mathbf{z}) = a_{d,r}\delta(\mathbf{x}-\mathbf{z})$.

A practical problem associated with the CP framework is that in these models inference has computational complexity $O(N^3D^3)$ and storage requirements $O(N^2D^2)$. Recently Álvarez and Lawrence (2009) introduced an efficient approximation for inference in this multi-output GP model. Their idea was to exploit a conditional independence assumption over the output functions $\{f_d(\mathbf{x})\}_{d=1}^{D}$: if the latent functions are fully observed then the output functions *are* conditionally independent of one another (as can be seen in (1)). Furthermore, if the latent processes are sufficiently smooth, the conditional independence assumption will hold approximately even for a finite number of observations of the latent functions $\left\{\{u_r(\mathbf{z}_k)\}_{k=1}^K\right\}_{r=1}^R$, where the variables $\{\mathbf{z}_k\}_{k=1}^K$ are usually referred to as the inducing inputs. These assumptions led to approximations that were very similar in spirit to the PITC and FITC approximations of Snelson and Ghahramani (2006); Quiñonero Candela and Rasmussen (2005).

In this paper we build on the work of Álvarez and Lawrence and extend it in two ways. First, we notice that if the locations of the inducing points are close relative to the length scale of the latent function, the PITC approximation will be accurate enough. However, if the length scale becomes small the approximation requires very many inducing points. In the worst case, the latent process could be white noise (as suggested by Higdon (2002) and implemented by Boyle and Frean (2005)). In this case the approximation will fail completely. To deal with such type of latent functions, we develop the concept of an *inducing function*, a generalization of the traditional concept of *inducing variable* commonly employed in several sparse GP methods. As we shall see, an inducing function is an artificial construction generated from a convolution operation between a smoothing kernel or *inducing kernel* and the latent functions u_r. The artificial nature of the inducing function is based on the fact that its construction is immersed in a variational-like inference procedure that does not modify the marginal likelihood of the true model. This leads us to the second extension of the paper: a problem with the FITC and PITC approximations can be their tendency to overfit when inducing inputs are optimized. A solution to this problem was given in a recent work by Titsias (2009) who provided a sparse GP approximation that has an associated variational bound. In this paper we show how the ideas of Titsias can be extended to the multiple output case. Our variational approximation is developed through the inducing functions and the quality of the approximation can be controlled through the inducing kernels and the number and location of the inducing inputs. Our approximation allows us to consider latent force models with a large number of states, D, and data points N. The use of inducing kernels also allows us to extend the inducing variable approximation of the latent force model framework to systems of *stochastic differential equations* (SDEs). We apply the approximation to different real world datasets, including a multivariate financial time series example.

A similar idea to the inducing function one introduced in this paper, was simultaneously proposed by Lázaro-Gredilla and Figueiras-Vidal (2010). Lázaro-Gredilla and Figueiras-Vidal (2010) introduced the concept of inducing feature to improve performance over the pseudo-inputs approach of Snelson and Ghahramani (2006) in sparse GP models. Our use of inducing functions and inducing kernels is motivated by the necessity to deal with non-smooth latent functions in the CP model of multiple outputs.

2 Multioutput GPs (MOGPs)

Let $\mathbf{y}_d \in \mathbb{R}^N$, where $d = 1, \ldots, D$, be the observed data associated with the output function $y_d(\mathbf{x})$. For simplicity, we assume that all the observations associated with different outputs are evaluated at the same inputs $\mathbf{X}$ (although this assumption is easily relaxed). We will often use the stacked vector $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_D)$ to collectively denote the data of all the outputs. Each observed vector $\mathbf{y}_d$ is assumed to be obtained by adding independent Gaussian noise to a vector of function values $\mathbf{f}_d$ so that the likelihood is $p(\mathbf{y}_d|\mathbf{f}_d) = \mathcal{N}(\mathbf{y}_d|\mathbf{f}_d, \sigma_d^2 I)$, where $\mathbf{f}_d$ is defined via (1). More precisely, the assumption in (1) is that a function value $f_d(\mathbf{x})$ (the noise-free version of $y_d(\mathbf{x})$) is generated from a common pool of R independent latent functions $\{u_r(\mathbf{x})\}_{r=1}^R$, each having a covariance function (Mercer kernel) given by $k_r(\mathbf{x}, \mathbf{x}')$. Notice that the outputs share the same latent functions, but they also have their own set of parameters $(\{\boldsymbol{\alpha}_{dr}\}_{r=1}^R, \sigma_d^2)$ where $\boldsymbol{\alpha}_{dr}$ are the parameters of the smoothing kernel $G_{d,r}(\cdot)$. Because convolution is a linear operation, the covariance between any pair of function values $f_d(\mathbf{x})$ and $f_{d'}(\mathbf{x}')$ is given by $k_{f_d,f_{d'}}(\mathbf{x}, \mathbf{x}') = \mathrm{Cov}[f_d(\mathbf{x}), f_{d'}(\mathbf{x}')] = \sum_{r=1}^R \int_{\mathcal{Z}} G_{d,r}(\mathbf{x} - \mathbf{z}) \int_{\mathcal{Z}} G_{d',r}(\mathbf{x}' - \mathbf{z}') k_r(\mathbf{z}, \mathbf{z}') \mathrm{d}\mathbf{z}\mathrm{d}\mathbf{z}'$. This covariance function is used to define a fully-coupled GP prior $p(\mathbf{f}_1, \ldots, \mathbf{f}_D)$ over all the function values associated with the different outputs. The joint probability distribution of the multioutput GP model can be written as $p(\{\mathbf{y}_d, \mathbf{f}_d\}_{d=1}^D) = \prod_{d=1}^D p(\mathbf{y}_d|\mathbf{f}_d) p(\mathbf{f}_1, \ldots, \mathbf{f}_D)$. The GP prior $p(\mathbf{f}_1, \ldots, \mathbf{f}_D)$ has a zero mean vector and a $(ND) \times (ND)$ covariance matrix $\mathbf{K}_{\mathbf{f},\mathbf{f}}$, where $\mathbf{f} = (\mathbf{f}_1, \ldots, \mathbf{f}_D)$, which consists of $N \times N$ blocks of the form $\mathbf{K}_{\mathbf{f}_d,\mathbf{f}_{d'}}$. Elements of each block are given by $k_{f_d,f_{d'}}(\mathbf{x}, \mathbf{x}')$ for all possible values of $\mathbf{x}$. Each such block is a cross-covariance (or covariance) matrix of pairs of outputs.

Prediction using the above GP model, as well as the maximization of the marginal likelihood $p(\mathbf{y}) = N(\mathbf{y}|\mathbf{0}, \mathbf{K}_{\mathbf{f},\mathbf{f}} + \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1^2\mathbf{I}, \ldots, \sigma_D^2\mathbf{I})$, requires $O(N^3D^3)$ time and $O(N^2D^2)$ storage which rapidly becomes infeasible even when only a few hundred outputs and data points are considered. Efficient approximations are needed in order to make the above multioutput GP model more practical.

3 PITC-like approximation for MOGPs

Before we propose our variational sparse inference method for multioutput GP regression in Section 4, we review the sparse method proposed by Álvarez and Lawrence (2009). This method is based on a likelihood approximation. More precisely, each output function $y_d(\mathbf{x})$ is independent from the other output functions given the full-length of each latent function $u_r(\mathbf{x})$. This means, that the likelihood of the data factorizes according to $p(\mathbf{y}|u) =$

$\prod_{d=1}^{D} p(\mathbf{y}_d|u) = \prod_{d=1}^{D} p(\mathbf{y}_d|\mathbf{f}_d)$, with $u = \{u_r\}_{r=1}^{R}$ the set of latent functions. The sparse method in Álvarez and Lawrence (2009) makes use of this factorization by assuming that it remains valid even when we are only allowed to exploit the information provided by a finite set of function values, $\mathbf{u}_r$, instead of the full-length function $u_r(\mathbf{x})$ (which involves uncountably many points). Let $\mathbf{u}_r$, for $r = 1, \ldots, R$, be a K-dimensional vector of values from the function $u_r(\mathbf{x})$ which are evaluated at the inputs $\mathbf{Z} = \{\mathbf{z}_k\}_{k=1}^{K}$. The vector $\mathbf{u} = (\mathbf{u}_1, \ldots, \mathbf{u}_R)$ denotes all these variables. The sparse method approximates the exact likelihood function $p(\mathbf{y}|u)$ with the likelihood $p(\mathbf{y}|\mathbf{u}) = \prod_{d=1}^{D} p(\mathbf{y}_d|\mathbf{u}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{y}_d|\boldsymbol{\mu}_{\mathbf{f}_d|\mathbf{u}}, \boldsymbol{\Sigma}_{\mathbf{f}_d|\mathbf{u}} + \sigma_d^2 I)$, where $\boldsymbol{\mu}_{\mathbf{f}_d|\mathbf{u}} = \mathbf{K}_{\mathbf{f}_d,\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}$ and $\boldsymbol{\Sigma}_{\mathbf{f}_d|\mathbf{u}} = \mathbf{K}_{\mathbf{f}_d,\mathbf{f}_d} - \mathbf{K}_{\mathbf{f}_d,\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}_d}$ are the mean and covariance matrices of the conditional GP priors $p(\mathbf{f}_d|\mathbf{u})$. The matrix $\mathbf{K}_{\mathbf{u},\mathbf{u}}$ is a block diagonal covariance matrix where the rth block $\mathbf{K}_{\mathbf{u}_r,\mathbf{u}_r}$ is obtained by evaluating $k_r(\mathbf{z}, \mathbf{z}')$ at the inducing inputs $\mathbf{Z}$. Further, the matrix $\mathbf{K}_{\mathbf{f}_d,\mathbf{u}}$ is defined by the cross-covariance function $\mathrm{Cov}[f_d(\mathbf{x}), u_r(\mathbf{z})] = \int_{\mathcal{Z}} G_{d,r}(\mathbf{x} - \mathbf{z}') k_r(\mathbf{z}', \mathbf{z}) d\mathbf{z}'$. The variables $\mathbf{u}$ follow the GP prior $p(\mathbf{u}) = N(\mathbf{u}|\mathbf{0}, \mathbf{K}_{\mathbf{u},\mathbf{u}})$ and can be integrated out to give the following approximation to the exact marginal likelihood:

$$p(\mathbf{y}|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{D} + \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}} + \boldsymbol{\Sigma}). \quad (2)$$

Here, $\mathbf{D}$ is a block-diagonal matrix, where each block is given by $\mathbf{K}_{\mathbf{f}_d,\mathbf{f}_d} - \mathbf{K}_{\mathbf{f}_d,\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}_d}$ for all d. This approximate marginal likelihood represents exactly each diagonal (output-specific) block $\mathbf{K}_{\mathbf{f}_d,\mathbf{f}_d}$ while each off diagonal (cross-output) block $\mathbf{K}_{\mathbf{f}_d,\mathbf{f}_{d'}}$ is approximated by the Nyström matrix $\mathbf{K}_{\mathbf{f}_d,\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}_{d'}}$.

The above sparse method has a similar structure to the PITC approximation introduced for single-output regression (Quiñonero Candela and Rasmussen, 2005). Because of this similarity, Álvarez and Lawrence (2009) call their multioutput sparse approximation PITC as well. Two of the properties of this PITC approximation (which may sometimes be seen as limitations) are:

1. It assumes that all latent functions u are smooth.
2. It is based on a modification of the initial full GP model. This implies that the inducing inputs $\mathbf{Z}$ are extra kernel hyparameters in the modified GP model.

Because of point 1, the method is not applicable when the latent functions are white noise processes. An important class of problems where we have to deal with white noise processes arise in linear SDEs where the above sparse method is currently not applicable there. Because of 2, the maximization of the marginal likelihood in eq. (2) with respect to $(\mathbf{Z}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are model hyperparameters, may be prone to overfitting especially when the number of variables in $\mathbf{Z}$ is large. Moreover, fitting a modified sparse GP model implies that the full GP model is not approximated in a systematic and rigorous way since there is no distance or divergence between the two models that is minimized.

In the next section, we address point 1 above by introducing the concept of variational inducing kernels that allow us to efficiently sparsify multioutput GP models having white noise latent functions. Further, these inducing kernels are incorporated into the variational inference method of Titsias (2009) (thus addressing point 2) that treats the inducing inputs $\mathbf{Z}$ as well as other quantities associated with the inducing kernels as variational parameters. The whole variational approach provides us with a very flexible, robust to overfitting, approximation framework that overcomes the limitations of the PITC approximation.

4 Sparse variational approximation

In this section, we introduce the concept of variational inducing kernels (VIKs). VIKs give us a way to define more general inducing variables that have larger approximation capacity than the $\mathbf{u}$ inducing variables used earlier and importantly allow us to deal with white noise latent functions. To motivate the idea, we first explain why the $\mathbf{u}$ variables can work when the latent functions are smooth and fail when these functions become white noises.

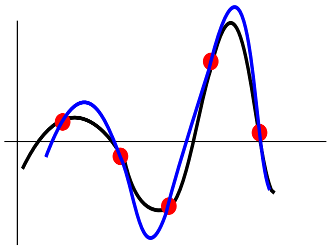
In PITC, we assume each latent function $u_r(\mathbf{x})$ is smooth and we sparsify the GP model through introducing, $\mathbf{u}_r$, inducing variables which are direct observations of the latent function, $u_r(\mathbf{x})$, at particular input points. Because of the latent function's smoothness, the $\mathbf{u}_r$ variables also carry information about other points in the function through the imposed prior over the latent function. So, having observed $\mathbf{u}_r$ we can reduce the uncertainty of the whole function.

With the vector of inducing variables $\mathbf{u}$, if chosen to be sufficiently large relative to the length scales of the latent functions, we can efficiently represent the functions $\{u_r(\mathbf{x})\}_{r=1}^{R}$ and subsequently variables $\mathbf{f}$ which are just convolved versions of the latent functions.[1] When the reconstruction of $\mathbf{f}$ from $\mathbf{u}$ is perfect, the conditional prior $p(\mathbf{f}|\mathbf{u})$ becomes a delta function and the sparse PITC approximation becomes exact. Figure 1(a) shows a cartoon description of a summarization of $u_r(\mathbf{x})$ by $\mathbf{u}_r$.

In contrast, when some of the latent functions are *white noise* processes the sparse approximation will fail. If $u_r(\mathbf{z})$ is white noise[2] it has a covariance function $\delta(\mathbf{z} - \mathbf{z}')$. Such processes naturally arise in the application of *stochastic differential equations* (see section 6) and are the ultimate non-

[1] This idea is like a "soft version" of the Nyquist-Shannon sampling theorem. If the latent functions were bandlimited, we could compute exact results given a high enough number of inducing points. In general they won't be bandlimited, but for smooth functions low frequency components will dominate over high frequencies, which will quickly fade away.

[2] Such a process can be thought as the "time derivative" of the Wiener process.


(a) Latent function is smooth

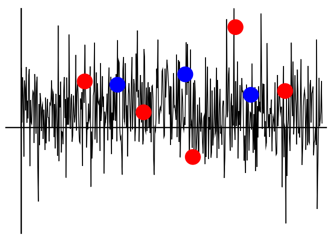

(b) Latent function is noise

Figure 1: With a smooth latent function as in (a), we can use some inducing variables $\mathbf{u}_r$ (red dots) from the complete latent process $u_r(\mathbf{x})$ (in black) to generate smoothed versions (for example the one in blue), with uncertainty described by $p(u_r|\mathbf{u}_r)$. However, with a white noise latent function as in (b), choosing inducing variables $\mathbf{u}_r$ (red dots) from the latent process (in black) does not give us a clue about other points (for example the blue dots).

smooth processes where two values $u_r(\mathbf{z})$ and $u_r(\mathbf{z}')$ are uncorrelated when $\mathbf{z} \neq \mathbf{z}'$. When we apply the sparse approximation a vector of "white-noise" inducing variables $\mathbf{u}_r$ does not carry information about $u_r(\mathbf{z})$ at any input $\mathbf{z}$ that differs from all inducing inputs $\mathbf{Z}$. In other words there is no additional information in the conditional prior $p(u_r(\mathbf{z})|\mathbf{u}_r)$ over the unconditional prior $p(u_r(\mathbf{z}))$. Figure 1(b) shows a pictorial representation. The lack of structure makes it impossible to exploit the correlations in the standard sparse methods like PITC.[3]

Our solution to this problem is the following. We will define a more powerful form of inducing variable, one based not around the latent function at a point, but one given by the convolution of the latent function with a smoothing kernel. More precisely, let us replace each inducing vector $\mathbf{u}_r$ with variables $\boldsymbol{\lambda}_r$ which are evaluated at the inputs $\mathbf{Z}$ and are defined according to

$$\lambda_r(\mathbf{z}) = \int T_r(\mathbf{z}-\mathbf{v})u_r(\mathbf{v})\mathrm{d}\mathbf{v}, \tag{3}$$

where $T_r(\mathbf{x})$ is a smoothing kernel (e.g. Gaussian) which we call the *inducing kernel* (IK). This kernel is not necessarily related to the model's smoothing kernels. These newly defined inducing variables can carry information about $u_r(\mathbf{z})$ not only at a single input location but from the entire input space. We can even allow a separate IK for each inducing point, this is, if the set of inducing points is $\mathbf{Z} = \{\mathbf{z}_k\}_{k=1}^K$, then $\lambda_r(\mathbf{z}_k) = \int T_{r,k}(\mathbf{z}_k - \mathbf{v})u_r(\mathbf{v})\mathrm{d}\mathbf{v}$, with the advantage of associating to each inducing point $\mathbf{z}_k$ its own set of adaptive parameters in $T_{r,k}$. For the PITC approximation, this adds more hyperparameters to the likelihood, perhaps leading to overfitting. However, in the variational approximation we define all these new parameters as variational parameters and therefore they do not cause the model to overfit.

If $u_r(\mathbf{z})$ has a white noise[4] GP prior the covariance function for $\lambda_r(\mathbf{x})$ is

$$\mathrm{Cov}[\lambda_r(\mathbf{x}), \lambda_r(\mathbf{x}')] = \int T_r(\mathbf{x}-\mathbf{z})T_r(\mathbf{x}'-\mathbf{z})\mathrm{d}\mathbf{z} \tag{4}$$

and the cross-covariance between $f_d(\mathbf{x})$ and $\lambda_r(\mathbf{x}')$ is

$$\mathrm{Cov}[f_d(\mathbf{x}), \lambda_r(\mathbf{x}')] = \int G_{d,r}(\mathbf{x}-\mathbf{z})T_r(\mathbf{x}'-\mathbf{z})d\mathbf{z}. \tag{5}$$

Notice that this cross-covariance function, unlike the case of $\mathbf{u}$ inducing variables, maintains a weighted integration over the whole input space. This implies that a single inducing variable $\lambda_r(\mathbf{x})$ can properly propagate information from the full-length process $u_r(\mathbf{x})$ into $\mathbf{f}$.

It is possible to combine the IKs defined above with the PITC approximation of Álvarez and Lawrence (2009), but in this paper our focus will be on applying them within the variational framework of Titsias (2009). We therefore refer to the kernels as variational inducing kernels (VIKs).

Variational inference

We now extend the variational inference method of Titsias (2009) to deal with multiple outputs and incorporate them into the VIK framework.

We compactly write the joint probability model $p(\{\mathbf{y}_d, \mathbf{f}_d\}_{d=1}^D)$ as $p(\mathbf{y}, \mathbf{f}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f})$. The first step of the variational method is to augment this model with inducing variables. For our purpose, suitable inducing variables are defined through VIKs. More precisely, let $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_R)$ be the whole vector of inducing variables where each $\boldsymbol{\lambda}_r$ is a K-dimensional vector of values obtained according to eq. (3). $\boldsymbol{\lambda}_r$'s role is to carry information about the latent function $u_r(\mathbf{z})$. Each $\boldsymbol{\lambda}_r$ is evaluated at the inputs $\mathbf{Z}$ and has its own VIK, $T_r(\mathbf{x})$, that depends on parameters $\boldsymbol{\theta}_{T_r}$. The $\boldsymbol{\lambda}$ variables augment the GP model according to $p(\mathbf{y}, \mathbf{f}, \boldsymbol{\lambda}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\boldsymbol{\lambda})p(\boldsymbol{\lambda})$. Here, $p(\boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{\lambda}|\mathbf{0}, \mathbf{K}_{\boldsymbol{\lambda},\boldsymbol{\lambda}})$ and $\mathbf{K}_{\boldsymbol{\lambda},\boldsymbol{\lambda}}$ is a block diagonal

[3]Returning to our sampling theorem analogy, the white noise process has infinite bandwidth. It is therefore impossible to represent it by observations at a few fixed inducing points.

[4]It is straightforward to generalize the method for rough latent functions that are not white noise or to combine smooth latent functions with white noise.

matrix where each block $\mathbf{K}_{\boldsymbol{\lambda}_r,\boldsymbol{\lambda}_r}$ is obtained by evaluating the covariance function in eq. (4) at the inputs $\mathbf{Z}$. Additionally, $p(\mathbf{f}|\boldsymbol{\lambda}) = \mathcal{N}(\mathbf{f}|\mathbf{K}_{\mathbf{f},\boldsymbol{\lambda}}\mathbf{K}_{\boldsymbol{\lambda},\boldsymbol{\lambda}}^{-1}\boldsymbol{\lambda}, \mathbf{K}_{\mathbf{f},\mathbf{f}} - \mathbf{K}_{\mathbf{f},\boldsymbol{\lambda}}\mathbf{K}_{\boldsymbol{\lambda},\boldsymbol{\lambda}}^{-1}\mathbf{K}_{\boldsymbol{\lambda},\mathbf{f}})$ where the cross-covariance $\mathbf{K}_{\mathbf{f},\boldsymbol{\lambda}}$ is computed through eq. (5). Because of the consistency condition $\int p(\mathbf{f}|\boldsymbol{\lambda})p(\boldsymbol{\lambda})\mathrm{d}\boldsymbol{\lambda} = p(\mathbf{f})$, performing exact inference in the above augmented model is equivalent to performing exact inference in the initial GP model. Crucially, this holds for any values of the *augmentation* parameters $(\mathbf{Z}, \{\boldsymbol{\theta}_{T_r}\}_{r=1}^{R})$. This is the key property that allows us to turn these augmentation parameters into variational parameters by applying approximate sparse inference.

Our method now proceeds along the lines of Titsias (2009). We introduce the variational distribution $q(\mathbf{f},\boldsymbol{\lambda}) = p(\mathbf{f}|\boldsymbol{\lambda})\phi(\boldsymbol{\lambda})$, where $p(\mathbf{f}|\boldsymbol{\lambda})$ is the conditional GP prior defined earlier and $\phi(\boldsymbol{\lambda})$ is an arbitrary variational distribution. By minimizing the KL divergence between $q(\mathbf{f},\boldsymbol{\lambda})$ and the true posterior $p(\mathbf{f},\boldsymbol{\lambda}|\mathbf{y})$, we can compute the following Jensen's lower bound on the true log marginal likelihood (a detailed derivation of the bound is available in Álvarez *et al.* (2009b)):

$$F_V = \log \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \mathbf{K}_{\mathbf{f},\boldsymbol{\lambda}}\mathbf{K}_{\boldsymbol{\lambda},\boldsymbol{\lambda}}^{-1}\mathbf{K}_{\boldsymbol{\lambda},\mathbf{f}} + \boldsymbol{\Sigma}\right) - \frac{1}{2}\operatorname{tr}\left(\boldsymbol{\Sigma}^{-1}\widetilde{\mathbf{K}}\right),$$

where $\boldsymbol{\Sigma}$ is the covariance function associated with the additive noise process and $\widetilde{\mathbf{K}} = \mathbf{K}_{\mathbf{f},\mathbf{f}} - \mathbf{K}_{\mathbf{f},\boldsymbol{\lambda}}\mathbf{K}_{\boldsymbol{\lambda},\boldsymbol{\lambda}}^{-1}\mathbf{K}_{\boldsymbol{\lambda},\mathbf{f}}$. Note that this bound consists of two parts. The first part is the log of a GP prior with the only difference that now the covariance matrix has a particular low rank form. This form allows the inversion of the covariance matrix to take place in $O(NDK^2)$ time rather than $O(N^3D^3)$. The second part can be seen as a penalization term that regularizes the estimation of the parameters. Notice also that only the diagonal of the exact covariance matrix $\mathbf{K}_{\mathbf{f},\mathbf{f}}$ needs to be computed. Overall, the computation of the bound can be done efficiently in $O(NDK^2)$ time.

The bound can be maximized with respect to all parameters of the covariance function; both model parameters and variational parameters. The variational parameters are the inducing inputs $\mathbf{Z}$ and the parameters $\boldsymbol{\theta}_{T_r}$ of each VIK which are rigorously selected so that the KL divergence is minimized. In fact each VIK is also a variational quantity and one could try different forms of VIKs in order to choose the one that gives the best lower bound.

The form of the bound is very similar to the projected process approximation, also known as DTC (Csató and Opper, 2001; Seeger *et al.*, 2003; Rasmussen and Williams, 2006). However, the bound has an additional trace term that penalizes the movement of inducing inputs away from the data. This term converts the DTC approximation to a lower bound and prevents overfitting. In what follows, we refer to this approximation as DTCVAR, where the VAR suffix refers to the variational framework.

5 Experiments

We present results of applying the method proposed for two real-world datasets that will be described in short. We compare the results obtained using PITC, the intrinsic coregionalization model (ICM)[5] employed in Bonilla *et al.* (2008) and the method using variational inducing kernels. For PITC we estimate the parameters through the maximization of the approximated marginal likelihood of equation (2) using the scaled-conjugate gradient method. We use one latent function and both the covariance function of the latent process, $k_r(\mathbf{x},\mathbf{x}')$, and the kernel smoothing function, $G_{d,r}(\mathbf{x})$, follow a Gaussian form, this is $k(\mathbf{x},\mathbf{x}') = \mathcal{N}(\mathbf{x}-\mathbf{x}'|\mathbf{0},\mathbf{C})$, where $\mathbf{C}$ is a diagonal matrix. For the DTCVAR approximations, we maximize the variational bound F_V. Optimization is also performed using scaled conjugate gradient. We use one white noise latent function and a corresponding inducing function. The inducing kernels and the model kernels follow the same Gaussian form. Using this form for the covariance or kernel, all convolution integrals are solved analytically.

5.1 Exam score prediction

In this experiment the goal is to predict the exam score obtained by a particular student belonging to a particular school. The data comes from the Inner London Education Authority (ILEA).[6] It consists of examination records from 139 secondary schools in years 1985, 1986 and 1987. It is a random 50% sample with 15362 students. The input space consists of features related to each student and features related to each school. From the multiple output point of view, each school represents one output and the exam score of each student a particular instantiation of that output.

We follow the same preprocessing steps employed in Bonilla *et al.* (2008). The only features used are the student-dependent ones (year in which each student took the exam, gender, VR band and ethnic group), which are categorical variables. Each of them is transformed to a binary representation. For example, the possible values that the variable year of the exam can take are 1985, 1986 or 1987 and are represented as 100, 010 or 001. The transformation is also applied to the variables gender (two binary variables), VR band (four binary variables) and ethnic group (eleven binary variables), ending up with an input space with dimension 20. The categorical nature of data restricts the input space to 202 unique input feature vectors. However, two students represented by the same input vector $\mathbf{x}$ and belonging both to the same school d, can obtain different exam scores. To reduce this noise in the

[5]The ICM is a particular case of the LMC with one latent function (Goovaerts, 1997).

[6]Data is available at `http://www.cmm.bristol.ac.uk/learning-training/multilevel-m-support/datasets.shtml`

data, we follow Bonilla *et al.* (2008) in taking the mean of the observations that, within a school, share the same input vector and use a simple heteroskedastic noise model in which the variance for each of these means is divided by the number of observations used to compute it. The performance measure employed is the percentage of unexplained variance defined as the sum-squared error on the test set as a percentage of the total data variance.[7] The performance measure is computed for ten repetitions with 75% of the data in the training set and 25% of the data in the test set.

Figure 5.1 shows results using PITC, DTCVAR with one smoothing kernel and DTCVAR with as many inducing kernels as inducing points (DTCVARS in the figure). For 50 inducing points all three alternatives lead to approximately the same results. PITC keeps a relatively constant performance for all values of inducing points, while the DTCVAR approximations increase their performance as the number of inducing points increases. This is consistent with the expected behaviour of the DTCVAR methods, since the trace term penalizes the model for a reduced number of inducing points. Notice that all the approximations outperform independent GPs and the best result of the ICM presented in Bonilla *et al.* (2008).

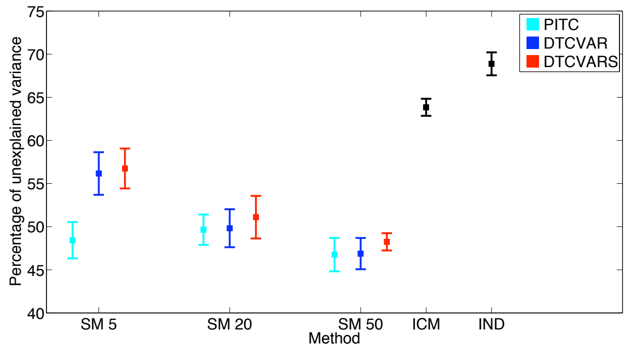


Figure 2: Exam score prediction results for the school dataset. Results include the mean of the percentage of unexplained variance of ten repetitions of the experiment, together with one standard deviation. In the bottom, SM X stands for sparse method with X inducing points, DTCVAR refers to the DTC variational approximation with one smoothing kernel and DTCVARS to the same approximation using as many inducing kernels as inducing points. Results using the ICM model and independent GPs (appearing as IND in the figure) have also been included.

5.2 Compiler prediction performance

In this dataset the outputs correspond to the speed-up of 11 C programs after some transformation sequence has been applied to them. The speed-up is defined as the execution time of the original program divided by the execution time of the transformed program. The input space consists of 13-dimensional binary feature vectors, where the presence of a one in these vectors indicates that the program has received that particular transformation. The dataset contains 88214 observations for each output and the same number of input vectors. All the outputs share the same input space. Due to technical requirements, it is important that the prediction of the speed-up for the particular program is made using few observations in the training set. We compare our results to the ones presented in Bonilla *et al.* (2008) and use $N = 16$, 32, 64 and 128 for the training set. The remaining $88214 - N$ observations are used for testing, employing as performance measure the mean absolute error. The experiment is repeated ten times and standard deviations are also reported. We only include results for the average performance over the 11 outputs.

Figure 3 shows the results of applying independent GPs (IND in the figure), the intrinsic coregionalization model (ICM in the figure), PITC, DTCVAR with one inducing kernel (DTCVAR in the figure) and DTCVAR with as many inducing kernels as inducing points (DTCVARS in the figure). Since the training sets are small enough, we also include results of applying the GP generated using the full covariance matrix of the convolution construction (see FULL GP in the figure). We repeated the experiment for different values of K, but show results only for $K = N/2$. Results for ICM and IND were obtained from Bonilla *et al.* (2008). In general, the DTCVAR variants outperform the

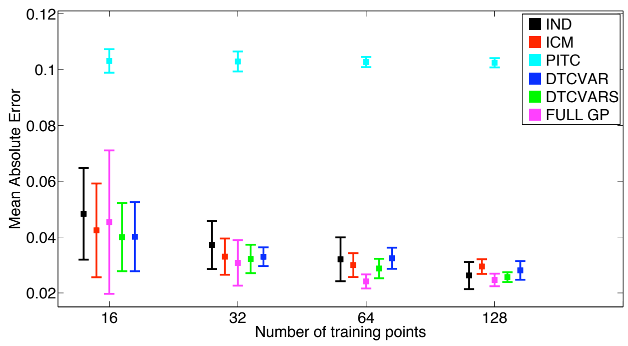


Figure 3: Mean absolute error and standard deviation over ten repetitions of the compiler experiment as a function of the training points. IND stands for independent GPs, ICM stands for intrinsic coregionalization model, DTCVAR refers to the DTCVAR approximation using one inducing kernel, DTCVARS refers to the DTCVAR approximation using as many inducing kernels as inducing points and FULL GP stands for the GP for the multiple outputs without any approximation.

ICM method, and the independent GPs for $N = 16,\ 32$ and 64. In this case, using as many inducing kernels as inducing points improves on average the performance. All methods, including the independent GPs are better than PITC. The size of the test set encourages the application of the sparse methods: for $N = 128$, making the prediction of the whole dataset using the full GP takes in average 22 minutes while the prediction with DTCVAR takes 0.65 minutes. Using more inducing kernels improves the performance, but also

[7]In Bonilla *et al.* (2008), results are reported in terms of explained variance.

makes the evaluation of the test set more expensive. For DTCVARS, the evaluation takes in average 6.8 minutes. Time results are average results over the ten repetitions.

6 Stochastic Latent Force Models

The starting point of stochastic differential equations is a stochastic version of the equation of motion, which is called Langevin's equation:

$$\frac{\mathrm{d}f(t)}{\mathrm{d}t} = -Cf(t) + Su(t), \qquad (6)$$

where $f(t)$ is the velocity of the particle, $-Cf(t)$ is a systematic friction term, $u(t)$ is a random fluctuation external force, i.e. white noise, and S indicates the sensitivity of the ouput to the random fluctuations. In the mathematical probability literature, the above is written more rigorously as $\mathrm{d}f(t) = -Cf(t)\mathrm{d}t + S\mathrm{d}W(t)$ where $W(t)$ is the Wiener process (standard Brownian motion). Since $u(t)$ is a GP and the equation is linear, $f(t)$ must be also a GP which turns out to be the Ornstein-Uhlenbeck (OU) process.

Here, we are interested in extending the Langevin equation to model multivariate time series. The way that the model in (6) is extended is by adding more output signals and more external forces. The forces can be either smooth (systematic or drift-type) forces or white noise forces. Thus,

$$\frac{\mathrm{d}f_d(t)}{\mathrm{d}t} = -D_d f_d(t) + \sum_{r=1}^{R} S_{d,r} u_r(t), \qquad (7)$$

where $f_d(t)$ is the dth output signal. Each $u_r(t)$ can be either a smooth latent force that is assigned a GP prior with covariance function $k_r(t,t')$ or a white noise force that has a GP prior with covariance function $\delta(t-t')$. That is, we have a composition of R latent forces, where R_s of them correspond to smooth latent forces and R_o correspond to white noise processes. The intuition behind this combination of input forces is that the smooth part can be used to represent medium/long term trends that cause a departure from the mean of the output processes, whereas the stochastic part is related to short term fluctuations around the mean. A model with $R_s = 1$ and $R_o = 0$ was proposed by Lawrence *et al.* (2007) to describe protein transcription regulation in a single input motif (SIM) gene network.

Solving the differential equation (7), we obtain

$$f_d(t) = e^{-D_d t} f_{d0} + \sum_{r=1}^{R} S_{d,r} \int_0^t e^{-D_d(t-z)} u_r(z) dz,$$

where f_{d0} arises from the initial condition. This model now is a special case of the multioutput regression model discussed in sections 1 and 2 where each output signal $y_d(t) = f_d(t) + \epsilon$ has a mean function $e^{-D_d t} f_{d0}$ and each model kernel $G_{d,r}(\mathbf{x})$ is equal to $S_{d,r} e^{-D_d(t-z)}$. The above model can be viewed as a stochastic latent force model (SLFM) following the work of Álvarez *et al.* (2009a).

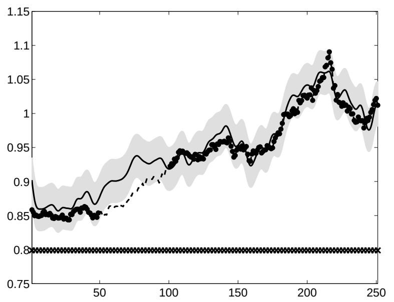
Latent market forces

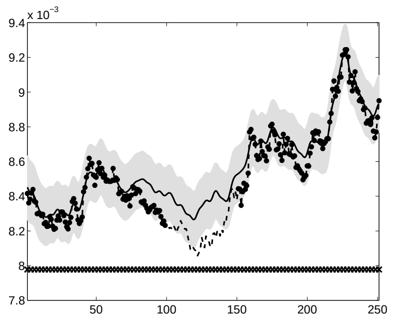
The application considered is the inference of missing data in a multivariate financial data set: the foreign exchange rate w.r.t. the dollar of 10 of the top international currencies (Canadian Dollar [CAD], Euro [EUR], Japanese Yen [JPY], Great British Pound [GBP], Swiss Franc [CHF], Australian Dollar [AUD], Hong Kong Dollar [HKD], New Zealand Dollar [NZD], South Korean Won [KRW] and Mexican Peso [MXN]) and 3 precious metals (gold [XAU], silver [XAG] and platinum [XPT]).[8] We considered all the data available for the calendar year of 2007 (251 working days). In this data there are several missing values: XAU, XAG and XPT have 9, 8 and 42 days of missing values respectively. On top of this, we also introduced artificially long sequences of missing data. Our objective is to model the data and test the effectiveness of the model by imputing these missing points. We removed a test set from the data by extracting contiguous sections from 3 currencies associated with very different geographic locations: we took days 50–100 from CAD, days 100–150 from JPY and days 150–200 from AUD. The remainder of the data comprised the training set, which consisted of 3051 points, with the test data containing 153 points. For preprocessing we removed the mean from each output and scaled them so that they all had unit variance.

It seems reasonable to suggest that the fluctuations of the 13 correlated financial time-series are driven by a smaller number of latent market forces. We therefore modelled the data with up to six latent forces which could be noise or smooth GPs. The GP priors for the smooth latent forces are assumed to have a Gaussian covariance function, $k_{u_r u_r}(t,t') = (1/\sqrt{2\pi\ell_r^2}) \exp(-((t-t')^2)/2\ell_r^2)$, where the hyperparameter ℓ_r is known as the lengthscale.

We present an example with $R = 4$. For this value of R, we consider all the possible combinations of R_o and R_s. The training was performed in all cases by maximizing the variational bound using the scale conjugate gradient algorithm until convergence was achieved. The best performance in terms of achieving the highest value for F_V was obtained for $R_s = 1$ and $R_o = 3$. We compared against the LMC model for different values of the latent functions in that framework. While our best model resulted in an standardized mean square error of 0.2795, the best LMC (with R=2) resulted in 0.3927. We plotted predictions from the latent market force model to characterize the performance when filling in missing data. In figure 4 we show the output signals obtained using the model with the highest bound ($R_s = 1$ and $R_o = 3$) for CAD, JPY and AUD. Note that the model performs better at capturing the deep drop in AUD than it does for the fluctuations in CAD and JPY.

[8]Data is available at `http://fx.sauder.ubc.ca/data.html`.


(a) CAD: Real data and prediction


(b) JPY: Real data and prediction

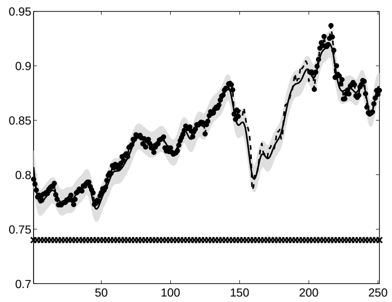

(c) AUD: Real data and prediction

Figure 4: Predictions from the model with $R_s = 1$ and $R_o = 3$ are shown as solid lines for the mean and grey bars for error bars at 2 standard deviations. For CAD, JPY and AUD the data was artificially held out. The true values are shown as a dotted line. Crosses on the x-axes of all plots show the locations of the inducing inputs.

7 Conclusions

We have presented a variational approach to sparse approximations in convolution processes. Our main focus was to provide efficient mechanisms for learning in multiple output Gaussian processes when the latent function is fluctuating rapidly. In order to do so, we have introduced the concept of inducing function, which generalizes the idea of inducing point, traditionally employed in sparse GP methods. The approach extends the variational approximation of Titsias (2009) to the multiple output case. Using our approach we can perform efficient inference on latent force models which are based around SDEs, but also contain a smooth driving force. Our approximation is flexible enough and has been shown to be applicable to a wide range of data sets, including high-dimensional ones.

Acknowledgements

We would like to thank Edwin Bonilla for his valuable feedback with respect to the datasets in section 5. Also to the authors of Bonilla *et al.* (2008) who kindly made the compiler dataset available. DL has been partly financed by the Spanish government through CICYT projects TEC2006-13514-C02-01 and TEC2009-14504-C02-01, and the CONSOLIDER-INGENIO 2010 Program (Project CSD2008-00010). MA and NL have been financed by a Google Research Award "Mechanistically Inspired Convolution Processes for Learning" and MA, NL and MT have been financed by EPSRC Grant No EP/F005687/1 "Gaussian Processes for Systems Identification with Applications in Systems Biology".

References

Mauricio A. Álvarez and Neil D. Lawrence. Sparse convolved Gaussian processes for multi-output regression. In *NIPS 21*, pages 57–64. MIT Press, 2009.

Mauricio A. Álvarez, David Luengo, and Neil D. Lawrence. Latent Force Models. In *JMLR: W&CP 5*, pages 9–16, 2009.

Mauricio A. Álvarez, David Luengo, Michalis K. Titsias, and Neil D. Lawrence. Variational inducing kernels for sparse convolved multiple output Gaussian processes. Technical report, School of Computer Science, University of Manchester, 2009. Available at `http://arxiv.org/abs/0912.3268`.

Edwin V. Bonilla, Kian Ming Chai, and Christopher K. I. Williams. Multi-task Gaussian process prediction. In *NIPS 20*, pages 153–160. MIT Press, 2008.

Phillip Boyle and Marcus Frean. Dependent Gaussian processes. In *NIPS 17*, pages 217–224. MIT Press, 2005.

Lehel Csató and Manfred Opper. Sparse representation for Gaussian process models. In *NIPS 13*, pages 444–450. MIT Press, 2001.

Pierre Goovaerts. *Geostatistics For Natural Resources Evaluation*. Oxford University Press, 1997.

David M. Higdon. Space and space-time modelling using process convolutions. In *Quantitative methods for current environmental issues*, pages 37–56. Springer-Verlag, 2002.

Neil D. Lawrence, Guido Sanguinetti, and Magnus Rattray. Modelling transcriptional regulation using Gaussian processes. In *NIPS 19*, pages 785–792. MIT Press, 2007.

Miguel Lázaro-Gredilla and Aníbal Figueiras-Vidal. Inter-domain Gaussian processes for sparse inference using inducing features. In *NIPS 22*, pages 1087–1095. MIT press, 2010.

Joaquin Quiñonero Candela and Carl Edward Rasmussen. A unifying view of sparse approximate Gaussian process regression. *JMLR*, 6:1939–1959, 2005.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

Alex Rogers, M. A. Osborne, S. D. Ramchurn, S. J. Roberts, and N. R. Jennings. Towards real-time information processing of sensor network data using computationally efficient multi-output Gaussian processes. In *Proc. Int. Conf. on Information Proc. in Sensor Networks (IPSN 2008)*, 2008.

Matthias Seeger, Christopher K. I. Williams, and Neil D. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In *Proc. 9th Int. Workshop on Artificial Intelligence and Statistics*, Key West, FL, 3–6 January 2003.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *NIPS 18*. MIT Press, 2006.

Yee Whye Teh, Matthias Seeger, and Michael I. Jordan. Semiparametric latent factor models. In *AISTATS 10*, pages 333–340, 2005.

Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *JMLR: W&CP 5*, pages 567–574, 2009.