Variational Dependent Multi-output Gaussian Process Dynamical Systems

Jing Zhao and Shiliang Sun

Department of Computer Science and Technology, East China Normal University
500 Dongchuan Road, Shanghai 200241, P. R. China
jzhao2011@gmail.com, slsun@cs.ecnu.edu.cn

Abstract. This paper presents a dependent multi-output Gaussian process (GP) for modeling complex dynamical systems. The outputs are dependent in this model, which is largely different from previous GP dynamical systems. We adopt convolved multi-output GPs to model the outputs, which are provided with a flexible multi-output covariance function. We adapt the variational inference method with inducing points for approximate posterior inference of latent variables. Conjugate gradient based optimization is used to solve parameters involved. Besides the temporal dependency, the proposed model also captures the dependency among outputs in complex dynamical systems. We evaluate the model on both synthetic and real-world data, and encouraging results are observed.

Keywords: Gaussian process, variational inference, dynamical system, multi-output modeling

1 Introduction

Dynamical systems are widespread in machine learning applications. Multi-output time series such as motion capture data and video sequences are typical examples of these systems. Modeling complex dynamical systems has a number of challenges such as only time as inputs, nonlinear mapping from time to observations, large data sets and possible dependency among multiple outputs. Gaussian processes (GPs) provide an elegant method for modeling nonlinear mappings in the Bayesian nonparametric learning framework [15]. Some extensions of GPs have been developed in recent years, which aim to solve these challenges.

Lawrence [9, 10] proposed the GP latent variable model (GP-LVM) as a nonlinear extension of the probabilistic principal component analysis [18]. GP-LVM can provide a visualization of high dimensional data by optimizing the latent variables with the maximum a posterior (MAP) solution. To overcome the difficulty of time and storage complexities for large data sets, some approximate methods, e.g., sparse GP [11] have been proposed for learning GP-LVM. By adding a Markov dynamical prior on the latent space, GP-LVM is extended to the GP dynamical model (GPDM) [21, 22] which is able to model nonlinear dynamical systems. GPDM captures the variability of outputs by constructing the variance of outputs with different parameters.

Instead of seeking a MAP solution for the latent variables as in the former methods, Titsias and Lawrence [20] introduced a variational Bayesian method for training GP-LVM. This method computes a lower bound of the logarithmic marginal likelihood by variationally integrating out the latent variables that appear nonlinearly in the inverse kernel matrix of the model. It was built on the method of variational inference with inducing points [19, 16]. This Bayesian GP-LVM was later adapted to multi-view learning [5] through introducing a softly shared latent space. Similarly, Damianou et al. [6] extended the Bayesian GP-LVM by imposing a dynamical prior on the latent space to the variational GP dynamical system (VGPDS). Park et al. [14] developed an almost direct application of VGPDS to phoneme classification. Besides variational approaches, expectation propagation based methods [7] are also capable of conducting approximate inference in Gaussian process dynamical systems (GPDS).

However, all the models mentioned above for GPDS ignore the dependency among multiple outputs, which usually assume that the outputs are conditionally independent. Actually, modeling the dependency among outputs is necessary in many applications such as sensor networks, geostatistics and time-series forecasting, which helps to make better predictions. Indeed, there are some recent works that explicitly considered the dependency of multiple outputs in GPs [3, 2, 23]. Latent force models (LFMs) [3] are a recent state-of-the-art modeling framework, which can model multi-output dependencies. Later, a series of extensions of LFMs were presented such as linear, nonlinear, cascaded and switching dynamical LFMs [1]. People also gave sequential inference methods for LFMs [8]. Álvarez and Lawrence [2] employed convolution processes to account for the correlations among outputs to construct a convolved multiple outputs GP (CMOGP) which can be regarded as a specific case of LFMs. Wilson et al. [23] combined neural networks with GPs to construct a GP regression network (GPRN). However, CMOGP and GPRN are neither introduced nor directly suitable for dynamical system modeling. When a dynamical prior is imposed, marginalizing over the latent variables is needed, which can be very challenging.

This paper proposes a variational dependent multi-output GP dynamical system (VDM-GPDS). The convolved process covariance function [2] is employed to capture the dependency among all the data points across all the outputs. To learn VDM-GPDS, we first approximate the latent functions in the convolution processes, and then variationally marginalize out the latent variables in the model. This leads to a convenient lower bound of the logarithmic marginal likelihood, which is then maximized by the scaled conjugate gradient method to find out the optimal parameters. Our model is applicable to general dependent multi-output dynamical systems rather than being specially tailored to a particular application. We adapt the model to different applications and obtain promising results.

2 The Proposed Model

Suppose we have multi-output time series data $\{\mathbf{y}_n, t_n\}_{n=1}^N$, where $\mathbf{y}_n \in \mathbb{R}^D$ is an observation at time $t_n \in \mathbb{R}^+$. We assume that there are low dimensional latent variables that govern the generation of the observations and a GP prior for the latent variables conditional on time captures the dynamical driving force of the observations, as in Damianou et al. [6]. However, a large difference compared with their work is that we explicitly model the dependency among the outputs through convolution processes [2].

Our model is a four-layer GP dynamical system. Here $\mathbf{t} \in \mathbb{R}^N$ represents the input variables in the first layer. Matrix $X \in \mathbb{R}^{N\times Q}$ represents the low dimensional latent variables in the second layer with element $x_{nq} = x_q(t_n)$. Similarly, matrix $F \in \mathbb{R}^{N\times D}$ denotes the latent variables in the third layer, with element $f_{nd} = f_d(\mathbf{x}_n)$ and matrix $Y \in \mathbb{R}^{N\times D}$ denotes the observations in the fourth layer whose nth row corresponds to $\mathbf{y}_n$. The model is composed of an independent multi-output GP mapping from $\mathbf{t}$ to X, a dependent multi-output GP mapping from X to F, and a linear mapping from F to Y.

Specifically, for the first mapping, $\mathbf{x}$ is assumed to be a multi-output GP indexed by time t similarly to Damianou et al. [6], that is $x_q(t) \sim \mathcal{GP}(0, \kappa_x(t,t'))$, $q = 1, ..., Q$, where individual components of the latent function $\mathbf{x}(t)$ are independent sample paths drawn from a GP with a certain covariance function $\kappa_x(t,t')$ parameterized by $\boldsymbol{\theta}_x$. There are several commonly used covariance functions such as the squared exponential covariance function (RBF) and Matern 3/2 function [6]. Given the above assumption, we have

$$p(X|\mathbf{t}) = \prod_{q=1}^{Q} p(\mathbf{x}_q|\mathbf{t}) = \prod_{q=1}^{Q} \mathcal{N}(\mathbf{x}_q|\mathbf{0}, \mathbf{K}_{\mathbf{t},\mathbf{t}}), \tag{1}$$

where $\mathbf{K}_{\mathbf{t},\mathbf{t}}$ is the covariance matrix. The covariance matrix may be constructed with any of the above covariance functions according to different applications.

For the second mapping, we assume that $\mathbf{f}$ is another multi-output GP indexed by $\mathbf{x}$, whose outputs are dependent, that is $f_d(\mathbf{x}) \sim \mathcal{GP}(0, \kappa_{f_d,f_{d'}}(\mathbf{x}, \mathbf{x}'))$, $d, d' = 1, ..., D$, where $\kappa_{f_d,f_{d'}}(\mathbf{x}, \mathbf{x}')$ is a convolved process covariance function which can capture the dependency among all the data points across all the outputs with parameters $\boldsymbol{\theta}_f = \{\{\Lambda_k\}, \{P_d\}, \{S_{d,k}\}\}$. The detailed formulation of $\kappa_{f_d,f_{d'}}(\mathbf{x}, \mathbf{x}')$ will be given in Sect. 2.1. From the conditional dependency among the latent variables $\{f_{nd}\}_{n=1,d=1}^{N,D}$, we have

$$p(F|X) = p(\mathbf{f}|X) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}_{\mathbf{f},\mathbf{f}}), \tag{2}$$

where $\mathbf{f}$ is a shorthand for $[\mathbf{f}_1^\top, ..., \mathbf{f}_D^\top]^\top$ and $\mathbf{K}_{\mathbf{f},\mathbf{f}}$ sized $ND\times ND$ is the covariance matrix in which the elements are calculated by $\kappa_{f_d,f_{d'}}(\mathbf{x}, \mathbf{x}')$.

The third mapping, which is from f_{nd} to the observation y_{nd} can be written as $y_{nd} = f_{nd} + \epsilon_{nd}$, where $\epsilon_{nd} \sim \mathcal{N}(0, \beta^{-1})$. Thus, we get

$$p(Y|F) = \prod_{d=1}^{D} \prod_{n=1}^{N} \mathcal{N}(y_{nd}|f_{nd}, \beta^{-1}). \tag{3}$$

Given the above setting, the graphical model for the proposed VDM-GPDS on the training data $\{\mathbf{y}_n, t_n\}_{n=1}^N$ can be depicted as Fig. 1. From (1), (2) and (3), the joint probability distribution for the VDM-GPDS model is given by

$$p(Y, F, X|\mathbf{t}) = p(\mathbf{f}|X) \prod_{d=1}^{D} \prod_{n=1}^{N} p(y_{nd}|f_{nd}) \prod_{q=1}^{Q} p(\mathbf{x}_q|\mathbf{t}). \tag{4}$$

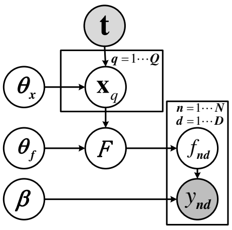

Fig. 1. The graphical model for VDM-GPDS.

2.1 Convolved Process Covariance Function

Since the outputs in our model are dependent, we need to capture the correlations among all the data points across all the outputs. Bonilla et al. [4] and Luttinen and Ilin [12] used a Kronecker product covariance matrix, which is very limited and actually a special case of some general covariances when covariances calculated from output dimensions and inputs are independent. In this paper, we use a more general and flexible model in which these two covariances are not separated. In particular, the convolution processes [2] are employed to model the latent function $F(X)$.

Now we introduce how to construct the convolved process covariance functions. Using latent functions $\{u_k(\mathbf{x})\}_{k=1}^K$ and smoothing kernels $\{G_{d,k}(\mathbf{x})\}_{d=1,k=1}^{D,K}$, $f_d(\mathbf{x})$ is supposed to be expressed through a convolution integral,

$$f_d(\mathbf{x}) = \sum_{k=1}^{K} \int_X G_{d,k}(\mathbf{x} - \tilde{\mathbf{x}}) u_k(\tilde{\mathbf{x}}) d\tilde{\mathbf{x}}. \tag{5}$$

The smoothing kernel is assumed to be Gaussian and formulated as $G_{d,k}(\mathbf{x}) = S_{d,k}\mathcal{N}(\mathbf{x}|\mathbf{0}, P_d)$, where $S_{d,k}$ is a scalar value that depends on the output index d and the latent function index k, and P_d is assumed to be diagonal. The latent process $u_k(\mathbf{x})$ is assumed to be Gaussian with covariance function

$$\kappa_k(\mathbf{x}, \mathbf{x}') = \mathcal{N}(\mathbf{x} - \mathbf{x}'|\mathbf{0}, \Lambda_k). \tag{6}$$

Thus, the covariance between $f_d(\mathbf{x})$ and $f_{d'}(\mathbf{x}')$ is

$$\kappa_{f_d, f_{d'}}(\mathbf{x}, \mathbf{x}') = \sum_{k=1}^{K} S_{d,k} S_{d',k} \mathcal{N}(\mathbf{x}|\mathbf{x}', P_d + P_{d'} + \Lambda_k). \tag{7}$$

The covariance between $f_d(\mathbf{x})$ and $u_k(\mathbf{x}')$ is

$$\kappa_{f_d,u_k}(\mathbf{x},\mathbf{x}') = S_{d,k}\mathcal{N}\left(\mathbf{x}-\mathbf{x}'|\mathbf{0}, P_d + \Lambda_k\right). \tag{8}$$

These covariance functions will be used for approximate inference in Sect. 3.

3 Inference and Optimization

The fully Bayesian learning for our model requires maximizing the logarithm of the marginal likelihood

$$p(Y|\mathbf{t}) = \int p(Y|F)p(F|X)p(X|\mathbf{t})dXdF. \tag{9}$$

Note that the integration w.r.t X is intractable, because X appears nonlinearly in the inverse of the matrix $\mathbf{K}_{\mathbf{f},\mathbf{f}}$. We attempt to make approximations for (9).

To begin with, we approximate $p(F|X)$ which is constructed by convolution process $f_d(\mathbf{x})$ in (5). Similarly to Álvarez and Lawrence [2], a generative approach is used to approximate $f_d(\mathbf{x})$ as follows. We first draw a sample, $\mathbf{u}_k(Z) = [u_k(\mathbf{z}_1), ..., u_k(\mathbf{z}_M)]^\top$, where $Z = \{\mathbf{z}_m\}_{m=1}^M$ are introduced as a set of input vectors for $u_k(\tilde{\mathbf{x}})$ and will be learned as parameters. We next sample $u_k(\tilde{\mathbf{x}})$ from the conditional prior $p(u_k(\tilde{\mathbf{x}})|\mathbf{u}_k)$. According to the above generating process, $u_k(\tilde{\mathbf{x}})$ in (5) can be approximated by the expectation $\mathcal{E}(u_k(\tilde{\mathbf{x}})|\mathbf{u}_k)$. Let $U = \{\mathbf{u}_k\}_{k=1}^K$ and $\mathbf{u} = [\mathbf{u}_1^\top, ..., \mathbf{u}_K^\top]^\top$. We get the probability distribution of $\mathbf{f}$ conditional on $\mathbf{u}, X, Z$ as follows

$$p(\mathbf{f}|\mathbf{u}, X, Z) = \mathcal{N}(\mathbf{f}|\mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}, \mathbf{K}_{\mathbf{f},\mathbf{f}} - \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}), \tag{10}$$

where $\mathbf{K}_{\mathbf{f},\mathbf{u}}$ is the cross-covariance matrix between $f_d(\mathbf{x})$ and $u_k(\mathbf{z})$ with element $\kappa_{f_d,u_k}(\mathbf{x},\mathbf{x}')$ in (8), block-diagonal matrix $\mathbf{K}_{\mathbf{u},\mathbf{u}}$ is the covariance matrix between $u_k(\mathbf{z})$ and $u_k(\mathbf{z}')$ with element $\kappa_k(\mathbf{x},\mathbf{x}')$ in (6), and $\mathbf{K}_{\mathbf{f},\mathbf{f}}$ is the covariance matrix between $f_d(\mathbf{x})$ and $f_{d'}(\mathbf{x}')$ with element $\kappa_{f_d,f_{d'}}(\mathbf{x},\mathbf{x}')$ in (7). Therefore, $p(F|X)$ is approximated by $p(\mathbf{f}|X,Z){=}\int p(\mathbf{f}|\mathbf{u},X,Z)p(\mathbf{u}|Z)d\mathbf{u}$ and $p(Y|\mathbf{t})$ is converted to

$$p(Y|\mathbf{t}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u}, X, Z)p(\mathbf{u}|Z)p(X|\mathbf{t})dFdUdX, \tag{11}$$

where $p(\mathbf{u}|Z) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{u},\mathbf{u}})$ and $\mathbf{y} = [\mathbf{y}_1^\top, ..., \mathbf{y}_D^\top]^\top$. It is worth noting that (11) is still intractable as the integration w.r.t X remains difficult.

Then, we introduce a lower bound of the $\log p(Y|\mathbf{t})$. We construct a variational distribution $q(F,U,X|Z)$ to approximate the distribution $p(F,U,X|Y,\mathbf{t})$ and compute the Jensen's lower bound on the $\log p(Y|\mathbf{t})$ as

$$\mathcal{L} = \int q(F,U,X|Z)\log\frac{p(Y,F,U,X|\mathbf{t},Z)}{q(F,U,X|Z)}dXdUdF. \tag{12}$$

The variational distribution is assumed to be factorized as

$$q(F,U,X|Z) = p(\mathbf{f}|\mathbf{u},X,Z)q(\mathbf{u})q(X). \tag{13}$$

$p(\mathbf{f}|\mathbf{u}, X, Z)$ in (13) is the same as the second term in (11), which will be eliminated during the variational computation. $q(\mathbf{u})$ is an approximation to $p(\mathbf{u}|X, Y)$, which is arguably Gaussian by maximizing the variational lower bound [6, 20]. $q(X)$ is an approximation to $p(X|Y)$, which is assumed to be a product of independent Gaussian distributions $q(X) = \prod_{q=1}^{Q} \mathcal{N}(\mathbf{x}_q|\boldsymbol{\mu}_q, S_q)$.

After some calculations and simplifications, the optimal lower bound becomes

$$\begin{aligned}\mathcal{L} =& \log\left[\frac{\beta^{\frac{ND}{2}}|\mathbf{K}_{\mathbf{u},\mathbf{u}}|^{\frac{1}{2}}}{(2\pi)^{\frac{ND}{2}}|\beta\psi_2+\mathbf{K}_{\mathbf{u},\mathbf{u}}|^{\frac{1}{2}}} \exp\{-\frac{1}{2}\mathbf{y}^\top W\mathbf{y}\}\right] \\ & -\frac{\beta\psi_0}{2} + \frac{\beta}{2}\mathrm{Tr}(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\psi_2) - \mathbf{KL}[q(X)||p(X|\mathbf{t})],\end{aligned} \tag{14}$$

where $W = \beta I - \beta^2\psi_1(\beta\psi_2 + \mathbf{K}_{\mathbf{u},\mathbf{u}})^{-1}\psi_1^\top$, $\psi_0 = \mathrm{Tr}(\langle\mathbf{K}_{\mathbf{f},\mathbf{f}}\rangle_{q(X)})$, $\psi_1 = \langle\mathbf{K}_{\mathbf{f},\mathbf{u}}\rangle_{q(X)}$ and $\psi_2 = \langle\mathbf{K}_{\mathbf{u},\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{u}}\rangle_{q(X)}$. $\mathbf{KL}[q(X)||p(X|\mathbf{t})]$ defined by $\int q(X)\log\frac{q(X)}{p(X|\mathbf{t})}dX$ is

$$\begin{aligned}\mathbf{KL}[q(X)||p(X|\mathbf{t})] =& \frac{Q}{2}\log|\mathbf{K}_{\mathbf{t},\mathbf{t}}| - \frac{1}{2}\sum_{q=1}^{Q}\log|S_q| \\ & + \frac{1}{2}\sum_{q=1}^{Q}[\mathrm{Tr}(\mathbf{K}_{\mathbf{t},\mathbf{t}}^{-1}S_q) + \mathrm{Tr}(\mathbf{K}_{\mathbf{t},\mathbf{t}}^{-1}\boldsymbol{\mu}_q\boldsymbol{\mu}_q^\top)] + const.\end{aligned} \tag{15}$$

Note that although the lower bound in (14) and the one in VGPDS [6] look similar, they are essentially distinct and have different meanings. In particular, the variables U in this paper are the samples of the latent functions $\{u_k(\mathbf{x})\}_{k=1}^{K}$ in the convolution process while in VGPDS they are samples F. Moreover, the covariance functions of F involved in this paper are multi-output covariance functions while VGPDS adopts single-output covariance functions. As a result, our model is more flexible and challenging.

3.1 Computation of ψ_0, ψ_1, ψ_2

Recall that the lower bound (14) requires computing the statistics $\{\psi_0, \psi_1, \psi_2\}$. We now detail how to calculate them. ψ_0 is a scalar that can be calculated as

$$\psi_0 = \sum_{n=1}^{N}\sum_{d=1}^{D}\int \kappa_{f_d,f_d}(\mathbf{x}_n, \mathbf{x}_n)\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_n, S_n)\, d\mathbf{x}_n = \sum_{d=1}^{D}\sum_{k=1}^{K}\frac{NS_{d,k}S_{d,k}}{(2\pi)^{\frac{Q}{2}}|2P_d + \Lambda_k|^{\frac{1}{2}}}. \tag{16}$$

ψ_1 is a $V \times W$ matrix whose elements are calculated as[1]

$$(\psi_1)_{v,w} = \int \kappa_{f_d,u_k}(\mathbf{x}_n, \mathbf{z}_m)\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_n, S_n)d\mathbf{x}_n = S_{d,k}\mathcal{N}(\mathbf{z}_m|\boldsymbol{\mu}_n, P_d + \Lambda_k + S_n), \tag{17}$$

[1] We borrow the density formulations to express ψ_1 as well as ψ_2.

where $V = N \times D$, $W = M \times K$, $d = \lfloor \frac{v-1}{N} \rfloor + 1$, $n = v - (d-1)N$, $k = \lfloor \frac{w-1}{M} \rfloor + 1$ and $m = w - (k-1)M$. Here the symbol "$\lfloor\rfloor$" means rounding down. ψ_2 is a $W \times W$ matrix whose elements are calculated as

$$\begin{aligned}(\psi_2)_{w,w'} &= \sum_{d=1}^{D}\sum_{n=1}^{N} \int \kappa_{f_d,u_k}(\mathbf{x}_n, \mathbf{z}_m)\kappa_{f_d,u_{k'}}(\mathbf{x}_n, \mathbf{z}_{m'})\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_n, S_n)d\mathbf{x}_n \\ &= \sum_{d=1}^{D}\sum_{n=1}^{N} S_{d,k}S_{d,k'}\mathcal{N}\left(\mathbf{z}_m|\mathbf{z}_{m'}, 2P_d + \mathit{\Lambda}_k + \mathit{\Lambda}_{k'}\right)\mathcal{N}\left(\frac{\mathbf{z}_m + \mathbf{z}_{m'}}{2}|\boldsymbol{\mu}_n, \mathit{\Sigma}_{\psi_2}\right),\end{aligned} \tag{18}$$

where $k = \lfloor \frac{w-1}{M} \rfloor + 1$, $m = w - (k-1)M$, $k' = \lfloor \frac{w'-1}{M} \rfloor + 1$, $m' = w' - (k'-1)M$ and $\mathit{\Sigma}_{\psi_2} = (P_d + \mathit{\Lambda}_k)^\top(2P_d + \mathit{\Lambda}_k + \mathit{\Lambda}_{k'})^{-1}(P_d + \mathit{\Lambda}_{k'}) + S_n$.

3.2 Conjugate Gradient Based Optimization

The parameters involved in (14) include the model parameters $\{\beta, \boldsymbol{\theta}_x, \boldsymbol{\theta}_f\}$ and the variational parameters $\{\{\boldsymbol{\mu}_q, S_q\}_{q=1}^Q, Z\}$. In order to reduce the variational parameters to be optimized and speed up convergence, we reparameterize the variational parameters $\boldsymbol{\mu}_q$ and S_q as $\bar{\boldsymbol{\mu}}_q$ and $\bar{\mathit{\Lambda}}_q$, respectively, as in Opper and Archambeau [13] and Damianou et al. [6]. The corresponding transformations are $S_q = (\mathbf{K}_{\mathbf{t},\mathbf{t}}^{-1} + \bar{\mathit{\Lambda}}_q)^{-1}$ and $\boldsymbol{\mu}_q = \mathbf{K}_{\mathbf{t},\mathbf{t}}\bar{\boldsymbol{\mu}}_q$. All the parameters are jointly optimized by the scaled conjugate gradient method to maximize the lower bound in (14).

4 Prediction

4.1 Prediction with Only Time

In the Bayesian framework, we need to compute the posterior distribution of the predicted outputs $Y_* \in \mathbb{R}^{N_* \times D}$ on some given time instants $\mathbf{t}_* \in \mathbb{R}^{N_*}$. With the parameters and time $\mathbf{t}_*$ omitted, the posterior density is given by

$$p\left(Y_*|Y\right) = \int p\left(Y_*|F_*\right)p\left(F_*|X_*, Y\right)p\left(X_*|Y\right)dF_*dX_*, \tag{19}$$

where $F_* \in \mathbb{R}^{N_* \times D}$ denotes the set of latent variables (the noise-free version of Y_*) and $X_* \in \mathbb{R}^{N_* \times Q}$ denotes the latent variables in the low dimensional space.

The distribution $p\left(F_*|X_*, Y\right)$ is approximated by the variational distribution

$$q(\mathbf{f}_*|X_*) = \int p(\mathbf{f}_*|\mathbf{u}, X_*)q(\mathbf{u})d\mathbf{u}, \tag{20}$$

where $\mathbf{f}_*^\top = [\mathbf{f}_{*1}^\top, ..., \mathbf{f}_{*D}^\top]$, and $p(\mathbf{f}_*|\mathbf{u}, X_*)$ is Gaussian expressed as $\mathcal{N}(\mathbf{f}_*|\mathbf{K}_{\mathbf{f}_*,\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}, \mathbf{K}_{\mathbf{f}_*,\mathbf{f}_*} - \mathbf{K}_{\mathbf{f}_*,\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}_*})$. Since the optimal setting for $q(\mathbf{u})$ is Gaussian, $q(\mathbf{f}_*|X_*)$ is Gaussian that can be computed analytically.

The distribution $p\left(X_*|Y\right)$ is approximated by the variational distribution $q\left(X_*\right)$ formulated as $q\left(X_*\right) = \mathcal{N}\left(\boldsymbol{\mu}_{X_*}, \boldsymbol{\Sigma}_{X_*}\right)$, where $\boldsymbol{\mu}_{X_*}$ is composed of column vector $\boldsymbol{\mu}_{\mathbf{x}_{*q}}$ with $\boldsymbol{\mu}_{\mathbf{x}_{*q}} = \mathbf{K}_{\mathbf{t}_*,\mathbf{t}}\mathbf{K}_{\mathbf{t},\mathbf{t}}^{-1}\boldsymbol{\mu}_q$ and block-diagonal matrix $\boldsymbol{\Sigma}_{X_*}$ has diagonal element $\boldsymbol{\Sigma}_{\mathbf{x}_{*q}}$ with $\boldsymbol{\Sigma}_{\mathbf{x}_{*q}} = \mathbf{K}_{\mathbf{t}_*,\mathbf{t}_*} - \mathbf{K}_{\mathbf{t}_*,\mathbf{t}}\mathbf{K}_{\mathbf{t},\mathbf{t}}^{-1}(\mathbf{K}_{\mathbf{t},\mathbf{t}_*} - S_q\mathbf{K}_{\mathbf{t},\mathbf{t}}^{-1}\mathbf{K}_{\mathbf{t},\mathbf{t}_*})$.

However, the integration of $q(\mathbf{f}_*|X_*)$ w.r.t $q(X_*)$ is not analytically feasible. Following Damianou et al. [6], we give the expectation of $\mathbf{f}_*$ as $\mathcal{E}(\mathbf{f}_*)$ and its element-wise autocovariance as vector $\mathcal{C}(\mathbf{f}_*)$ whose $(\tilde{n}\times d)$th entry is $\mathcal{C}(f_{\tilde{n}d})$ with $\tilde{n}=1,...,N_*$ and $d=1,...,D$.

$$\begin{aligned} \mathcal{E}(\mathbf{f}_*) &= \psi_{1*}\mathbf{b}, \\ \mathcal{C}(f_{\tilde{n}d}) &= \mathbf{b}^\top(\psi_{2\tilde{n}}^d - (\psi_{1\tilde{n}}^d)^\top\psi_{1\tilde{n}}^d)\mathbf{b} + \psi_{0*}^d - \mathrm{Tr}\left[(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} - (\mathbf{K}_{\mathbf{u},\mathbf{u}} + \beta\psi_2)^{-1})\psi_{2*}^d\right], \end{aligned} \tag{21}$$

where $\psi_{1*} = \langle\mathbf{K}_{\mathbf{f}_*,\mathbf{u}}\rangle_{q(X_*)}$, $\mathbf{b} = \beta(\mathbf{K}_{\mathbf{u},\mathbf{u}} + \beta\psi_2)^{-1}\psi_1^\top\mathbf{y}$, $\psi_{1\tilde{n}}^d = \langle\mathbf{K}_{f_{\tilde{n}d},\mathbf{u}}\rangle_{q(\mathbf{x}_{\tilde{n}})}$, $\psi_{2\tilde{n}}^d = \langle\mathbf{K}_{\mathbf{u},f_{\tilde{n}d}}\mathbf{K}_{f_{\tilde{n}d},\mathbf{u}}\rangle_{q(\mathbf{x}_{\tilde{n}})}$, $\psi_{0*}^d = \mathrm{Tr}(\langle\mathbf{K}_{\mathbf{f}_{*d},\mathbf{f}_{*d}}\rangle_{q(X_*)})$, $\psi_{2*}^d = \langle\mathbf{K}_{\mathbf{u},\mathbf{f}_{*d}}\mathbf{K}_{\mathbf{f}_{*d},\mathbf{u}}\rangle_{q(X_*)}$. Since Y_* is the noisy version of F_*, the expectation and element-wise autocovariance of Y_* are $\mathcal{E}(\mathbf{y}_*) = \mathcal{E}(\mathbf{f}_*)$ and $\mathcal{C}(\mathbf{y}_*) = \mathcal{C}(\mathbf{f}_*) + \beta^{-1}\mathbf{1}_{N_*D}$, where $\mathbf{y}_*^\top = [\mathbf{y}_{*1}^\top, ..., \mathbf{y}_{*D}^\top]$.

4.2 Prediction with Time and Partial Observations

In this case which is referred as reconstruction, we need to predict Y_*^m which represents the outputs on missing dimensions, given Y_*^{pt} which represents the outputs observed on partial dimensions. The posterior density of Y_*^m is given by

$$p(Y_*^m|Y_*^{pt},Y) = \int p(Y_*^m|F_*^m)p(F_*^m|X_*,Y_*^{pt},Y)p(X_*|Y_*^{pt},Y)dF_*^m dX_*. \tag{22}$$

$p(X_*|Y_*^{pt},Y)$ is approximated by $q(X_*)$ whose parameters need to be optimized for the sake of considering the partial observations Y_*^{pt}. This requires maximizing a new lower bound of $\log p(Y_*^{pt},Y)$ which can be computed analogously to (14). Moreover, parameters of the new variational distribution $q(X,X_*)$ are jointly optimized because of the coupling of X and X_*. Then the marginal distribution $q(X_*)$ is obtained from $q(X,X_*)$. Note that multiple sequences where X_* and X are independent, only the separated variational distribution $q(X_*)$ is optimized.

5 Experiment

5.1 Synthetic Data

In this section, we evaluate our method on synthetic data generated from a complex dynamical system. The latent variables X are independently generated by the Ornstein-Uhlenbeck (OU) process

$$dx_q = -\gamma x_q dt + \sqrt{\sigma^2}dW, \quad q=1,...,Q. \tag{23}$$

The outputs Y are generated through a multi-output GP

$$y_d(\mathbf{x}) \sim \mathcal{GP}(0, \kappa_{f_d,f_{d'}}(\mathbf{x},\mathbf{x}')), \quad d,d'=1,...,D, \tag{24}$$

where $\kappa_{f_d,f_{d'}}(\mathbf{x},\mathbf{x}')$ employs the convolution process with one latent function. In this paper, the number of the latent functions in (5) is set to one, i.e., $K=1$, which is also the common setting used in Álvarez and Lawrence [2]. We sample the synthetic data by two steps. First we use the differential equation with parameters $\gamma = 0.5$, $\sigma = 0.01$ to sample $N = 200$, $Q = 2$ latent variables at time interval $[-1, 1]$. Then we sample $D = 4$ dimensional outputs, each of which has 200 observations through the multi-output GP with parameters $S_{1,1} = 1$, $S_{2,1} = 2$, $S_{3,1} = 3$, $S_{4,1} = 4$, $P_1 = [5, 1]^\top$, $P_2 = [5, 1]^\top$, $P_3 = [3, 1]^\top$, $P_4 = [2, 1]^\top$ and $\Lambda = [4, 5]^\top$. In addition, white Gaussian noise is added to each output.

Prediction Here we evaluate the performance of our method for predicting the outputs given only time compared with CMOGP, GPDM and VGPDS. We randomly select 50 points from each output for training with the remaining 150 points for testing. This is repeated for ten times. The latent variables X in VGPDS and VDM-GPDS with two dimensions are initialized by using principal component analysis on the observations. Moreover, the Matern 3/2 covariance function and 30 inducing points are used in VGPDS and VDM-GPDS.

Table 1 presents the averaged root mean square error (RMSE) with the standard deviation (std) for predictions. The best results are shown in bold. Since the data in this experiment are generated from a complex dynamical system that combines two GP mappings, CMOGP which consists of only one GP mapping can not capture the complexity well. Moreover, VDM-GPDS models the explicit dependency among the multiple outputs while GPDM and VGPDS does not. Therefore, our model gives the best performance among the four models as expected. Besides highest accuracies, VDM-GPDS also has the smallest variances. In addition, to verify the flexibility of VDM-GPDS, we do experiments on the independent output data which are generated analogously to Sect. 5.1. GPDM and VGPDS which do not make the assumption of output dependency is included as comparisons. The results are given in Table 2 where we can see that our model performs as well as VGPDS and significantly better than GPDM.

Table 1. Averaged RMSE (%) with std (%) for predictions on the dependent output data.

	CMOGP	GPDM	VGPDS	VDM-GPDS
y_1	1.75±0.38	1.70±0.18	1.51±0.31	$\mathbf{1.43 \pm 0.23}$
y_2	3.46±0.67	3.32±0.27	2.99±0.53	$\mathbf{2.82 \pm 0.35}$
y_3	5.19±0.99	4.83±0.28	4.24±0.85	$\mathbf{4.09 \pm 0.59}$
y_4	7.50±0.94	5.98±0.55	5.16±0.92	$\mathbf{5.00 \pm 0.60}$

Reconstruction In this part, we compare VDM-GPDS with the k-nearest neighbor best (k-NNbest) method which chooses the best k from $\{1, \ldots, 5\}$, CMOGP and VGPDS for recovering missing points given time and partially observed outputs. Here, we do not include the results of GPDM because that GPDM is not directly suitable for reconstructing some dimensions given data of other. We set $S_{4,1} = -4$ to generate data in this part, which makes that the

Table 2. Averaged RMSE (%) with std (%) for predictions on the independent output data.

	GPDM	VGPDS	VDM-GPDS
y_1	3.82±1.55	$\mathbf{2.18 \pm 0.06}$	2.21±0.06
y_2	3.45±1.70	2.06±0.19	$\mathbf{2.05 \pm 0.13}$
y_3	3.57±1.71	$\mathbf{1.68 \pm 0.09}$	1.72±0.12
y_4	7.10±1.28	4.48±0.23	$\mathbf{4.45 \pm 0.20}$

output y_4 be negatively correlated with the others. We remove all outputs y_1 or y_4 at time interval $[0.5, 1]$ from the 50 training points, resulting in 35 points as training data. Note that CMOGP considers all the present outputs as the training set while VGPDS and VDM-GPDS only consider the outputs at time interval $[-1, 0.5)$ as the training set. Table 3 shows the results with four methods for reconstructions on the missing points for y_1 and y_4. It indicates the superior performance of our model for the reconstruction task.

Table 3. Averaged RMSE (%) with std (%) for reconstructions on y_1 and y_4.

	k-NNbest	CMOGP	VGPDS	VDM-GPDS
y_1	1.87±0.62	1.90±0.31	1.49±0.94	$\mathbf{0.98 \pm 0.34}$
y_4	13.51±2.54	9.31±0.87	6.79±6.07	$\mathbf{5.56 \pm 1.88}$

5.2 Human Motion Capture Data

Here the sequences of runs/jogs from subject 35 in the CMU motion capture database are employed for the reconstruction task. We preprocess the data as in Lawrence [11], which leads to nine independent training sequences and one testing sequence. The average length of each sequence is 40 frames and the output dimension is 59.

The RBF kernel is adopted in this set of experiments to construct $\mathbf{K}_{\mathbf{t},\mathbf{t}}$ which is a block-diagonal matrix because the sequences are independent. We compare our model with the nearest neighbor in the angle space (NN) and the scaled space (NN sc.) [17] and VGPDS. For parameter optimization of VDM-GPDS and VGPDS, the maximum numbers of iteration steps are set to be identical.

Table 4 gives results of four methods. LS and LA correspond to the reconstructions on the right leg in the scaled space and angle space. Similarly, BS and BA correspond to the upper body in the same two spaces. Clearly, our model outperforms the other approaches. We conjecture that this is because VDM-GPDS effectively considers both the dynamical characteristics and the dependency among the outputs in the complex dynamical system. Since GPDM cannot reconstruct the missing outputs on some dimensions given the others as explained in Sect. 5.1. We do experiments according to Wang et al. [22] to reconstruct the missing frames $21 - 43$ on all dimensions of the test data. We get the RMSE for reconstruction: 0.7323 with VDM-GPDS versus 0.9448 with GPDM and 5.1099 with VDM-GPDS versus 7.8984 with GPDM in the scaled space and angle space, respectively. It turns out that our model also defeats GPDM.

Table 4. The RMSE for reconstructions on the motion capture data.

	NN sc.	NN	CMOGP	VGPDS	VDM-GPDS
LS	0.8170	0.8493	1.1468	0.6502	**0.6379**
LA	6.7495	7.9441	13.5338	5.5356	**5.3026**
BS	1.0027	1.4018	3.5564	0.6569	**0.5961**
BA	5.6332	9.5748	5.0171	2.8108	**2.6033**

6 Conclusion

In this paper we have proposed a dependent multi-output GP for modeling complex dynamical systems. The convolved process covariance function is employed to model the dependency among all the data points across all the outputs. We adapt the variational inference method involving inducing points to our model so that the latent variables are variationally integrated out.

Modeling the possible dependency among multiple outputs can help to make better predictions. The effectiveness of the proposed model is empirically demonstrated. However, when the dimensionality of the output is very high, our model may take a long time to converge. This opens the possibility for future work to accelerate training for high dimensional dynamical systems.

Acknowledgments

This work is supported by NNSFC Projects 61370175 and 61075005, and Shanghai Knowledge Service Platform Project (No. ZF1213).

References

1. Álvarez, M.A., D.Luengo, Lawrence, N.D.: Linear latent force models using Gaussian processes. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 2693–2705 (2013)
2. Álvarez, M.A., Lawrence, N.D.: Computationally efficient convolved multiple output Gaussian processes. Journal of Machine Learning Research 12, 1459–1500 (2011)
3. Álvarez, M.A., Luengo, D., Lawrence, N.D.: Latent force models. In: Proceedings of the 12th International Conference on Articicial Intelligence and Statistics. pp. 9–16 (2009)
4. Bonilla, E.V., Chai, K.M., Williams, C.K.I.: Multi-task Gaussian process prediction. Advances in Neural Information Processing Systems 18, 153–160 (2008)
5. Damianou, A.C., Ek, C.H., Titsias, M.K., Lawrence, N.D.: Manifold relevance determination. In: Proceedings of the 29th International Conference on Machine Learning. pp. 145–152 (2012)
6. Damianou, A.C., Titsias, M.K., Lawrence, N.D.: Variational Gaussian process dynamical systems. Advances in Neural Information Processing Systems 24, 2510–2518 (2011)

7. Deisenroth, M.P., Mohamed, S.: Expectation propagation in Gaussian process dynamical systems. Advances in Neural Information Processing Systems 25, 2618–2626 (2012)
8. Hartikainen, J., Särkkä, S.: Sequential inference for latent force models. `http://arxiv.org/abs/1202.3730` (2012)
9. Lawrence, N.D.: Gaussian process latent variable models for visualisation of high dimensional data. Advances in Neural Information Processing Systems 17, 329–336 (2004)
10. Lawrence, N.D.: Probabilistic non-linear principal component analysis with Gaussian process latent variable models. Journal of Machine Learning Research 6, 1783–1816 (2005)
11. Lawrence, N.D.: Learning for larger dataset with the Gaussian process latent variable model. In: Proceedings of the 11th International Workshop on Artificial Intelligence and Statistics. pp. 243–250 (2007)
12. Luttinen, J., Ilin, A.: Efficient Gaussian process inference for short-scale spatio-temporal modeling. In: Proceedings of the 15th International Conference on Artificial Intelligence and Statistics. pp. 741–750 (2012)
13. Opper, M., Archambeau, A.: The variational Gaussian approximation revisited. Neural Computation 21, 786–792 (2009)
14. Park, H., Yun, S., Park, S., Kim, J., Yoo, C.D.: Phoneme classification using constrained variational Gaussian process dynamical system. Advances in Neural Information Processing Systems 22, 2015–2023 (2012)
15. Rasmussen, C.E., Williams, C.K.I.: Gaussian Process for Machine Learning. MIT Press (2006)
16. Sun, S.: A review of deterministic approximate inference techniques for Bayesian machine learning. Neural Computing and Applications 23, 2039–2050 (2013)
17. Taylor, G.W., Hinton, G.E., Roweis, S.: Modeling human motion using binary latent variables. Advances in Neural Information Processing Systems 17, 1345–1352 (2007)
18. Tipping, M.E., Bishop, C.M.: Probabilistic principal component analysis. Journal of the Royal Statistical Society 61, 611–622 (1999)
19. Titsias, M.K.: Variational learning of inducing variables in sparse Gaussian processes. In: Proceedings of the 12th International Conference on Artificial Intelligence and Statistics. pp. 567–574 (2009)
20. Titsias, M.K., Lawrence, N.D.: Bayesian Gaussian process latent variable model. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics. pp. 844–851 (2010)
21. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models. Advances in Neural Information Processing Systems 19, 1441–1448 (2006)
22. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models for human motion. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 283–398 (2008)
23. Wilson, A.G., Knowles, D.A., Ghahramani, Z.: Gaussian process regression networks. In: Proceedings of the 29th International Conference on Machine Learning. pp. 599–606 (2012)