DAG-GPs: Learning Directed Acyclic Graph Structure For Multi-Output Gaussian Processes

Anonymous authors
Paper under double-blind review

Abstract

Multi-output Gaussian processes (MOGPs) introduce correlations between outputs, but are subject to negative transfer, where learned correlations associate an output with another that is actually unrelated, leading to diminished predictive accuracy. Negative transfer may be countered by structuring the MOGP to follow conditional independence statements so that independent outputs do not correlate, but to date the imposed structures have been hand selected for specific applications. We introduce the DAG-GP model, which linearly combines latent Gaussian processes so that MOGP outputs follow a directed acyclic graph structure. Our method exposes a deep connection between MOGPs and Gaussian directed graphical models, which has not been explored explicitly in prior work. We propose to learn the graph from data prior to training the MOGP, so that correlations between outputs are only introduced when justified. Automated structure learning means no prior knowledge of the conditional independence between outputs is required, and training multiple MOGPs to identify the best structure is no longer necessary. Graph structure is learned by applying existing structure learning algorithms developed for graphical models to a downselected set of MOGP training data. Experiments on real world data sets show that with sufficiently expressive kernels, prediction error and likelihood are improved when using the DAG-GP model compared to state of the art exact MOGP methods.

1 Introduction

Multi-output Gaussian processes (MOGPs) apply Gaussian processes to tasks with multiple output dimensions. Correlations are learned between outputs, so that changes in one output dimension may be used to improve prediction accuracy in another. Existing approaches to MOGPs allow for arbitrary correlations between outputs, but this expressiveness leads to a risk of negative transfer, where random correlations resulting from small sample statistics are incorrectly identified during training as correlations between output dimensions. Particularly in examples with small amounts of training data and complex covariance kernels, negative transfer leads to poor predictive performance and underestimation of prediction errors. The effects of negative transfer can be mitigated by selecting an MOGP structure that controls which correlations are possible, but there currently exists no principled method for structure selection.

In the graphical models community, highly connected graphical models also tend to fit to noise, leading to diminished predictive performance (Freno & Trentin, 2011). Structure learning algorithms, that identify structurally simple models with high likelihood, have been developed to better learn a model of the data. The learned models enforce conditional independence statements between variables, and are more robust to noise, resulting in improved predictive accuracy.

We present a method for limiting negative transfer in exact MOGPs by using techniques from graphical model structure learning. Our approach, referred to as the DAG-GP model, enforces a directed acyclic graph (DAG) structure between output variables of a multi-output Gaussian process. We are able to apply algorithms developed for graphical models to learn the graph structure from data, limiting inter-output correlations to those that are substantial, and removing the need to train and test on different MOGP structures. Our work is the first to identify the utility of applying graphical model structure learning to MOGPs. With sufficiently complex kernels, we show that the DAG-GP model outperforms state of the art MOGP techniques on exact inference tasks on real life data sets.

2 Related work

A frequently taken approach to multi-output Gaussian process regression is to combine the outputs of multiple independent latent single output Gaussian processes, either linearly or through convolution (Álvarez et al., 2012; Liu et al., 2018). Linear combination models including the intrinsic coregionalization model (Journel & Huijbregts, 1978), the linear model of coregionalization (Goovaerts et al., 1997), and the semiparametric latent factor model (SLFM) (Teh et al., 2005) all represent outputs as local mixtures of the latent processes. The methods differ in the number of latent processes used and whether the latent processes share kernels. Sharing latent processes between outputs leads to correlations between outputs, while the use of multiple kernels leads to correlations over multiple spatial scales for each output. Extensions have explored applying higher rank combinations of the latent processes (Bonilla et al., 2008), allowing correlated latent processes (Vargas-Guzmán et al., 2002), and adding an additional latent process unique to each output (Nguyen et al., 2014). In all cases, predictive performance is strongly dependent upon choices of the number of latent processes and how they are mixed. Determination of the optimal model requires training all possibilities and testing. In contrast, our use of structure learning may be viewed as an automated method of selecting which latent processes contribute to which output variables.

Convolutional methods instead construct output processes by convolving latent processes with a smoothing kernel. Initial approaches used a single white noise process (Ver Hoef & Barry, 1998; Melkumyan & Ramos, 2011), while more recent extensions have used mixtures of multiple latent processes (Boyle & Frean, 2005) and allowed latent processes different from white noise (Alvarez & Lawrence, 2009; Lawrence et al., 2007). Convolution allows non-local effects such as time delays to be modeled, but this leads to more hyperparameters to train, and a greater potential for negative transfer if the number of latent processes is not carefully controlled.

Our method is most similar to a class of methods we refer to as Gaussian process autoregressors, where outputs are considered sequentially and are computed using transformations of previous outputs, leading to a Bayesian network structure. However, work to date has used one of a fully connected network (Requeima et al., 2019), a single parent for each output (Kennedy & O'Hagan, 2000), or a bipartite network (Leen et al., 2012). Choices of structure are application driven, designed to transfer information between outputs according to known relationships. Our approach has the same goal of controlling information flow, but we generalize on these methods by allowing any DAG structure between outputs and learning that structure directly from the data. The possible structures include those used in the previously mentioned work, but typically they are found to differ. Gaussian processes autoregressors bear similarity to deep Gaussian processes (Damianou & Lawrence, 2013), but deep GPs use the outputs of latent GPs as inputs to another layer of GPs, rather than networks between output dimensions.

Finally, a related body of work has considered inference on Gaussian processes defined over trees and general undirected graphs (Sudderth et al., 2004; Wainwright et al., 2001; Venkitaraman et al., 2020). These methods improve prediction capability when using small training sets. However, in these cases the graphical model is imposed over the input space for GPs with a single output variable, whereas we consider a graphical model imposed between multiple output variables.

3 Preliminaries

3.1 Gaussian processes

Gaussian processes (GPs) are continuous valued regressors that model any finite set of input/output pairs $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, $\boldsymbol{x}_i \in \mathbb{R}^{D_x}$, $y_i \in \mathbb{R}$ as drawn from a N-dimensional Gaussian distribution (Williams & Rasmussen, 2006). The GP $f \sim \mathcal{GP}(m, k)$ is specified through its mean function $m(\cdot)$ and covariance kernel $k(\cdot, \cdot)$. For input and output vectors $\boldsymbol{X} = [\boldsymbol{x}_1 \dots \boldsymbol{x}_n]^T$ and $\boldsymbol{Y} = [y_1 \dots y_n]^T$, the output vector is modeled as drawn from the GP plus independent noise,

$$f(\boldsymbol{X}) \sim \mathcal{N}(m(\boldsymbol{X}), k(\boldsymbol{X}, \boldsymbol{X})), \tag{1}$$

$$\boldsymbol{Y} = f(\boldsymbol{X}) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}), \tag{2}$$

where $[k(\boldsymbol{X}, \boldsymbol{X})]_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. It is typical to shift the output such that $m(\boldsymbol{X}) = 0$, and we proceed under this assumption for the rest of this paper.

Prediction of outputs $\boldsymbol{Y}_*$ at inputs $\boldsymbol{X}_*$ is accomplished using a conditional Gaussian distribution,

$$f(\boldsymbol{X}_*) \mid \boldsymbol{X}, \boldsymbol{Y} \sim \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{K}_*), \tag{3}$$

$$\boldsymbol{\mu}_* = k(\boldsymbol{X}_*, \boldsymbol{X}) \left[k(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 \boldsymbol{I}\right]^{-1} \boldsymbol{Y}, \tag{4}$$

$$\boldsymbol{K}_* = k(\boldsymbol{X}_*, \boldsymbol{X}_*) - k(\boldsymbol{X}_*, \boldsymbol{X}) \left[k(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 \boldsymbol{I}\right]^{-1} k(\boldsymbol{X}, \boldsymbol{X}_*). \tag{5}$$

Gaussian process covariance kernels are selected so that covariance matrix is always positive definite. Popular kernels include the radial basis function (RBF) kernel

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \theta^2 \exp\left(-\frac{1}{2} \sum_{d=1}^{D_x} \frac{(\boldsymbol{x}_{i,d} - \boldsymbol{x}_{j,d})^2}{l_d^2}\right) \tag{6}$$

and the spectral mixture (SM) kernel (Wilson & Adams, 2013)

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{s=1}^{S} \theta_s \prod_{d=1}^{D_x} \exp\left(-\frac{(\boldsymbol{x}_{i,d} - \boldsymbol{x}_{j,d})^2}{2 l_{s,d}^2}\right) \cos\left(\nu_{s,d}(\boldsymbol{x}_{i,d} - \boldsymbol{x}_{j,d})\right). \tag{7}$$

Kernel hyperparameters and noise are selected to maximize $\log p(\boldsymbol{Y})$.

3.2 Multi-output gaussian processes

Multi-output Gaussian processes (MOGPs) extend GPs to treat multivariate output, where $\boldsymbol{y}_i \in \mathbb{R}^{D_y}$. An MOGP expresses all outputs as a single ND_y-dimensional Gaussian distribution,

$$f(\boldsymbol{X}) \sim \mathcal{N}(0, \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})), \quad f(\boldsymbol{X})^T = \left[f_1(\boldsymbol{X})^T \ \ldots \ f_{D_y}(\boldsymbol{X})^T\right], \tag{8}$$

where f_m corresponds to the predictions for output dimension m. $\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})$ may be considered to be composed of D_y^2 blocks of size $N \times N$. The block of $\boldsymbol{K}_{m,n}$ corresponding to outputs m and n is specified through a cross covariance kernel,

$$\boldsymbol{K}_{m,n}(\boldsymbol{X}, \boldsymbol{X}) = k_{mn}(\boldsymbol{X}, \boldsymbol{X}). \tag{9}$$

The kernels are defined such that the overall covariance matrix is positive definite. Then, dimension m of the outputs follow $\boldsymbol{Y}_m = f_m(\boldsymbol{X}) + \boldsymbol{\epsilon}_m$, where $\boldsymbol{\epsilon}_m \sim \mathcal{N}(0, \sigma_m^2 \boldsymbol{I})$.

3.3 Constructing MOGPs from latent processes

Linear MOGP methods use multiple independent single-output GPs that are combined to introduce correlations between the output dimensions. The semiparametric latent factor model (SLFM) models each output as the weighted linear combination of Q independent latent processes u_q. The Collaborative MOGP (CoGP) model adds an additional process w_m specific to each output, so that

$$f_m(\boldsymbol{x}_i) = w_m(\boldsymbol{x}_i) + \sum_{q=1}^{Q} \lambda_{mq} u_q(\boldsymbol{x}_i), \tag{10}$$

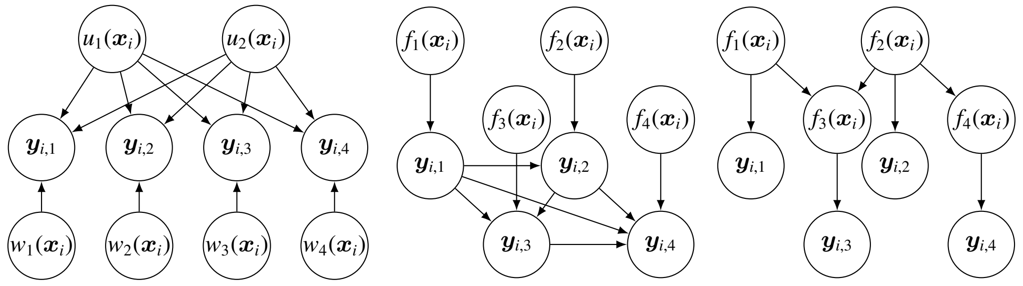
where each $w_m \sim \mathcal{GP}(0, \bar{k}_m)$ and $u_q \sim \mathcal{GP}(0, \tilde{k}_q)$. The CoGP model is visualized in figure 1a. The covariance kernel between outputs is

$$k_{mn}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \delta_{mn} \bar{k}_m(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{q=1}^{Q} \lambda_{mq} \lambda_{nq} \tilde{k}_q(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{11}$$

where δ_{mn} is the Kronecker delta function. In particular, when using RBF kernels for all latent processes, each kernel k_{mn} encodes correlations over $Q + 1$ length scales in each input dimension with the CoGP and Q length scales with the SLFM.

Convolutional processes instead model each output as a convolution of the latent processes with a smoothing kernel function. Álvarez & Lawrence (2011) recommend a general purpose kernel in which the kernel smoothing function and latent covariances follow a Gaussian distribution. In this case, the kernels may be expressed as follows, with matrices $\boldsymbol{S}_m$, $\boldsymbol{S}_n$, and $\boldsymbol{S}_q$

$$k_{mn}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{q=1}^{Q} \lambda_{mq} \lambda_{nq} \frac{|\boldsymbol{S}_m|^{\frac{1}{4}} |\boldsymbol{S}_n|^{\frac{1}{4}}}{\sqrt{|\boldsymbol{S}_m + \boldsymbol{S}_n + \boldsymbol{S}_q|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^T (\boldsymbol{S}_m + \boldsymbol{S}_n + \boldsymbol{S}_q)^{-1} (\boldsymbol{x}_i - \boldsymbol{x}_j)\right). \tag{12}$$



(a) CoGP model (b) GPAR model (c) DAG-GP model

Figure 1: Comparison of MOGP model structures.

4 Gaussian processes autoregressors and DAG-GPs

MOGPs that rely on combinations of latent processes suffer from negative transfer (Leen et al., 2012), where outputs that are not related are correlated in the model. False correlations between outputs can reduce MOGP predictive performance in two ways; they can lead to incorrect mean predictions as patterns in one output dimension are projected onto another, and they can reduce variance estimates below truth as noise is found to be explained by other outputs.

Negative transfer can be combated by controlling information flow between outputs. Gaussian process autoregressors allow for control over correlations by forming each output as a transformation of previous outputs. We focus on local linear transformations, because they have been shown to be effective (Leen et al., 2012), and they permit inference to be performed exactly. When applied to the process f_m, the MOGP is constructed as

$$f_m(\boldsymbol{x}_i) = u_m(\boldsymbol{x}_i) + \sum_{n \in pa(m)} \lambda_{mn} f_n(\boldsymbol{x}_i). \tag{13}$$

where $u_m \sim \mathcal{GP}(0, \tilde{k}_m)$ and $pa(m)$ is the parent set of index m, consisting of all indices that are combined to produce f_m.

In some previously applied methods, linear transformations are instead applied directly to the outputs $\boldsymbol{y}_{i,m}$. For example, the GPAR model (Requeima et al., 2019) in figure 1b connects outputs using parent sets containing all previously solved output dimensions. We focus with DAG-GPs on networks connecting the processes f_m as in (13), which is visualized in figure 1c. This model captures noise that can be specific to a single output dimension, and in our experiments performs better than applying a DAG structure between raw outputs. This is also the more general model, because the structure in figure 1b can be recovered by setting $\epsilon_m = 0$ and adding an independent noise to the kernel $\tilde{k}_m$. However, this choice does prevent optimizing the MOGP as D_y separate single-output GPs, as is performed by (Kennedy & O'Hagan, 2000) and (Requeima et al., 2019).

4.1 The DAG-GP model

Through appropriate selection of parent sets so that no process depends linearly on itself, the model in (13) may be naturally interpreted as a Directed Acyclic Graph (DAG) between $f_m(\boldsymbol{x})$. Critically, the DAG implies the existence of conditional independence relationships between the output dimensions. The DAG-GP model selects the appropriate conditional independence relationships - corresponding to appropriate DAG structure - that prevent negative transfer between outputs in the MOGP. This corresponds to choosing which parent sets will appear in the structural equation model.

Previously applied autoregressive models may be interpreted as DAG-GP models with a DAG structure selected based on assumptions specific to the application. These methods have not explicitly considered their model to be the construction of a DAG, and have not applied insights from the study of graphical models. With the DAG-GP model we propose to use graphical model structure learning to obtain conditional independence statements directly from the data, in order to formulate a DAG structure between the outputs that is appropriate for any problem. As a result, the DAG-GP model may be used regardless of whether an appropriate structure between output dimensions is already known.

Let the set of non-descendants $nd(m)$ consists of the indices of all processes that cannot be reached by following directed edges from $f_m(\boldsymbol{x})$. Then the DAG-GP model satisfies all conditional independence relationships of the form $f_m(\boldsymbol{X}) \perp\!\!\!\perp f_{nd(m)}(\boldsymbol{X}) \mid f_{pa(m)}(\boldsymbol{X})$ in addition to $f_m(\boldsymbol{x}_i) \perp\!\!\!\perp f_{nd(m)}(\boldsymbol{x}_i) \mid f_{pa(m)}(\boldsymbol{x}_i)$ for any $\boldsymbol{x}_i$. These results follow immediately from the conditional independence statements followed by a distribution that factors according to DAG, plus the equivalence between a linear structural equation model (13) and a Gaussian DAG (Pearl et al., 2000). We provide a more formal proof in the appendix.

Learning structure from data has been successful in the graphical models community. Constructing a DAG has been shown to improve prediction of unobserved data by preventing negative transfer of information between unrelated variables. This has lead to development of a number of structure learning techniques for DAGs, and we are able to leverage these methods directly to learn a DAG structure to be used in the DAG-GP.

4.2 Comparison of DAG-GPs and linear MOGP models

Define $\boldsymbol{A}$ as the $D_y \times D_y$ matrix such that $\boldsymbol{A}_{m,n} = \lambda_{mn}$ where defined and 0 otherwise, and let $\boldsymbol{B} = (\boldsymbol{I} - \boldsymbol{A})^{-1}$. Then

$$f(\boldsymbol{x}_i) = \boldsymbol{B}u(\boldsymbol{x}_i), \tag{14}$$

and covariance kernels are constructed using

$$k_{mn}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{d=1}^{D_y} \boldsymbol{B}_{m,d}\boldsymbol{B}_{n,d}\tilde{k}_d(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{15}$$

From (14) and (15) it is clear that any DAG-GP is equivalent to an SLFM instance with D_y latent processes and appropriately selected linear parameters. The difference is that in the DAG-GP model, the elements of $\boldsymbol{B}$ cannot be selected arbitrarily, they must be consistent with construction from fewer values of λ_{mn}. There are QD_y coefficients connecting the latent processes to the outputs in the SLFM, while in our experiments, each output averages two parents or fewer in the DAG-GP model, leading to fewer coefficients to be optimized compared to the SLFM with a large number of latent processes. Since the SLFM is more general, it reaches a higher marginal likelihood during training. The success of the DAG-GP model reveals that the additional marginal likelihood arises from correlations that are learned from noise, and reduce predictive accuracy in the MOGP.

A consequence of having fewer connecting coefficients is that the number of latent processes in k_{mn} is at most $max(|an(m)|, |an(n)|)$ for $m \neq n$, and at most $|an(m)| + 1$ for $m = n$, where the set of ancestors $an(m)$ consists of the indices of all processes that can reach $f_m(\boldsymbol{x})$ by following directed edges. Processes with few ancestors in the DAG have simpler kernels, with fewer effective length scales. For example, for the DAG-GP in figure 1c, k_{11} and k_{13} are diffent multiples of $\tilde{k}_1$, while k_{33} is a weighted sum of $\tilde{k}_1$, $\tilde{k}_2$, and $\tilde{k}_3$. Since kernels are effectively simpler for processes with few ancestors, we recommend kernels with mixtures of more than one length scale in DAG-GPs. Our experiments show that this is critical to achieving strong predictive performance.

5 Structure selection

In the graphical models community, DAG structure is typically optimized through score-based or constraint-based methods. Score-based methods maximize data likelihood subject to a penalization on graph complexity, while constraint-based methods identify conditional independence statements from the data and construct a consistent graph. DAG-GP structure can be selected through application of these methods.

In principle two frequently used score-based methods, the Akaike Information Criterion (AIC) (Akaike, 1974) and the Bayesian Information Criterion (BIC) (Schwarz et al., 1978) can be used directly as objectives during MOGP optimization. AIC and BIC for DAGs are defined below, where $|E|$ is the number of edges in the corresponding DAG

$$AIC = \max\{2\log p(\boldsymbol{Y})\} - 2|E| \tag{16}$$

$$BIC = \max\{2\log p(\boldsymbol{Y})\} - |E|\ln(N). \tag{17}$$

Optimization of structure could be performed simultaneously with DAG-GP hyperparameter optimization, with the highest scoring DAG-GP selected. However, this approach significantly increases MOGP training complexity, and has a high risk of converging to locally optimum structure unless a DAG-GP is trained for every possible DAG. Instead, we learn the structure from data before training Gaussian process parameters, so training only a single DAG-GP is required.

We face two challenges in applying DAG structure learning algorithms to DAG-GPs. First, structure learning algorithms typically rely on the assumption that samples from the DAG are independent and identically distributed. Independence does not hold in DAG-GPs, where data correlation is fundamental. Second, we only have access to $\boldsymbol{Y}$, which are noisy observations of $f(\boldsymbol{X})$.

5.1 Generating approximately i.i.d. training data

We make use of structure learning algorithms developed for i.i.d. data by using the conditional independence in DAG-GPs at each specific input, i.e. $f_m(\boldsymbol{x}_i) \perp\!\!\!\perp f_{nd(m)}(\boldsymbol{x}_i) \mid f_{pa(m)}(\boldsymbol{x}_i)$. Large classes of kernels of interest, including the RBF and SM kernels used in this paper, are stationary and decay rapidly with distance between inputs. By downselecting training data so that all inputs are separated by a distance that is much larger than the length scale in such kernels, the training data at each input may be treated as approximately i.i.d. draws from a D_y-dimensional Gaussian DAG. Existing structure learning algorithms may then be applied to the downselected data set.

In practice, the length scale of correlations is unknown prior to training, and excessive downselection decreases structure learning accuracy. In our experiments, we found strong performance using all training data or by selecting data to be separated by twice the average closest neighbor distance. The success of this method, despite significant correlations between data at different inputs, suggests that additional data for structure learning is preferable to strict insistence on independence.

5.2 Choice of structure learning method

A number of constraint-based methods have been proposed to perform structure learning on data corrupted by observation noise. However, the shortcomings of these methods range from strict restrictions on structure (Anandkumar et al., 2013), to requirements that the noise model be known (Saeed et al., 2020), to a requirement that the number of leaf nodes in the DAG to be known (Zhang et al., 2017). For our score-based experiment, we follow the suggestion of Zhang et al. (2017) for Gaussian DAGs with unknown noise, which applies the deterministic PC algorithm and requires an estimate for the number of leaf nodes found by first optimizing BIC.

For the training set sizes we consider, independence tests tend to be imprecise. We have greater success applying score-based methods, where we test with AIC, BIC, and BGe (Geiger & Heckerman, 1994). The presence of measurement noise means the maximum likelihood of the DAG must be computed numerically. Combined structure learning and likelihood maximization can be performed for AIC and BIC through the MS-EM algorithm (Friedman, 1997), but the complexity of structure learning is still significantly increased compared to when the DAG variables are observable, and BGe, which assumes a fully observable DAG, cannot be applied. Analysis by Zhang et al. (2017) suggests that the covariance of noisy output is similar to the covariance of $f(\boldsymbol{x}_i)$ unless any σ_m^2 is comparable in magnitude to the variance of $f_m(\boldsymbol{x}_i)$. In the problems where effective learning can be performed, we expect the noise variance to be small compared to the variance of $f_m(\boldsymbol{x}_i)$, so we apply score-based algorithms to the noisy data directly. Experiments with this method show strong results, with negligible time required for structure learning.

6 Experiments

6.1 Description

We tested DAG-GPs against an independent process model, SLFM, CoGP, the convolutional approach (Conv), and GPAR. For latent process models, we ran the experiments with multiple numbers of latent processes, and present the highest likelihood and lowest error achieved. We compared DAG-GPs with structure selected through deterministic PC (DPC), maximization of AIC, BIC, BGe, and unpenalized likelihood (Full). DAGs were learned with all data, and with data separated by two

Table 1: Mean negative log likelihood (NLL) and 2-norm error (Err) for RBF and SM2 kernels. Bolded entries indicate lowest result in each column.

	ANDROMEDA				**EXCHANGE**				**JURA**			
	RBF		**SM2**		**RBF**		**SM2**		**RBF**		**SM2**	
METHOD	**NLL**	**ERR**	**NLL**	**ERR**	**NLL**	**ERR**	**NLL**	**ERR**	**NLL**	**ERR**	**NLL**	**ERR**
Independent	4.995	0.628	4.464	0.583	3.328	0.418	2.696	0.390	15.248	2.006	14.978	1.891
SLFM	**3.574**	0.636	3.613	0.569	2.670	0.423	2.027	0.384	13.822	1.880	13.963	1.823
CoGP	4.060	0.656	4.248	0.617	**1.993**	**0.379**	1.884	0.380	14.158	1.873	13.899	1.810
Conv	3.813	**0.557**	-	-	2.390	0.392	-	-	15.003	**1.857**	-	-
GPAR	3.826	0.630	3.520	0.595	2.343	0.412	2.019	0.393	**13.662**	1.917	13.443	1.888
DAG Full	3.842	0.693	3.522	0.588	2.710	0.408	1.941	0.392	14.092	1.951	14.663	1.903
DAG AIC	4.122	0.701	**3.174**	**0.515**	2.605	0.404	**1.861**	**0.375**	14.067	1.922	13.514	1.831
DAG BIC	4.122	0.702	**3.174**	**0.515**	2.557	0.408	1.881	0.382	13.960	1.912	13.584	**1.783**
DAG BGe	4.176	0.722	3.386	0.547	2.759	0.418	1.946	0.379	14.010	1.921	**13.374**	1.789
DAG DPC	4.437	0.717	3.241	0.532	2.971	0.416	2.002	0.378	14.511	2.069	13.792	1.842

and four times the average nearest neighbor distance, with lowest NLL and error presented. Tests were performed using both the RBF kernel and the spectral mixture kernel with two components (SM2), except with the convolutional method, as there is no closed form expression for convolutions with spectral mixture kernels.

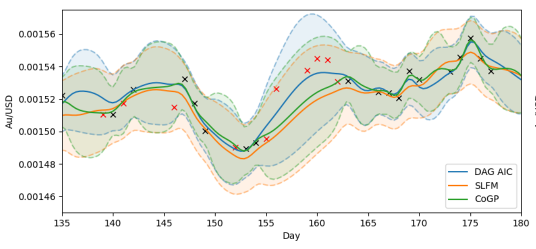
In each experiment, all outputs at a random subset of inputs were selected for training in four separate runs. The task was to jointly predict all outputs at remaining inputs. Data was normalized to mean 0 and standard deviation 1. Score-based DAGs were learned using GOBNILP (Cussens et al., 2017), and MOGP optimization was performed using BFGS. We present the 2-norm error of the predicted mean (Err) and the negative log-likelihood (NLL) of the normalized testing data averaged across all runs. Our initial experiment was performed on the Andromeda data set (Hatzikos et al., 2008) as collected by Spyromitros-Xioufis et al. (2016). The data consists of a daily average of water quality variables observed by an underwater vehicle across 49 days. We trained on temperature, conductivity, salinity, and oxygen concentration data from 45 of the 49 days. Our second test used a time series of daily exchange rate values against USD across 2007. We modeled the exchange rates of silver, gold, and the currencies CAD, EUR, JPY, and GBP. 209 complete data entries were available, of which 150 were used for training. Finally, we used the Jura data set as an example of multiple input dimensions and a larger number of output dimensions. The data set records concentrations of 7 elements sampled at various x,y locations in the Swiss Jura (Goovaerts et al., 1997). We trained on 150 random locations out of 259, and simultaneously modeled all 7 outputs.[1]

6.2 Results

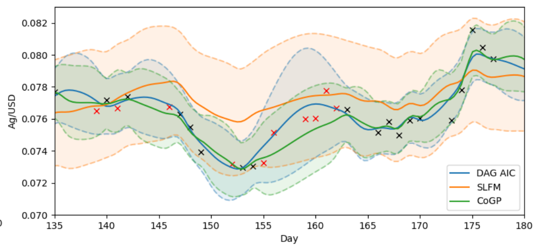
Mean results in table 1 show the DAG-GP model leads to lower average prediction errors and negative log likelihood when using the SM2 kernel, but is not competitive when limited to RBF kernels. We hypothesize this to be a consequence of the less expressive kernels in output dimensions with few ancestors, as discussed in section 4.2. The simple RBF kernels are unable to capture a covariance with multiple length scales that arises from real data, and this deficiency is worse than the potential for negative transfer. Once more flexible kernels are used, the DAG-GP model with sparse DAG structure shows lower negative log likelihood and mean error when compared to fully connected DAG structures and to other state of the art methods. The alternate approaches typically have higher marginal likelihood on the training data than DAG-GPs, but the correlations that are learned do not generalize to unseen data. Based on our experimental results, we recommend the use of AIC as a scoring criterion, as it leads to the lowest NLL and error in the Andromeda and Exchange experiments, and performs competitively in the Jura dataset.

The benefit of the learned DAG structure in the Exchange data set is shown in figures 2a and 2b, which display learned GPs with SM2 kernels against a crest in gold (Au) and silver (Ag) prices

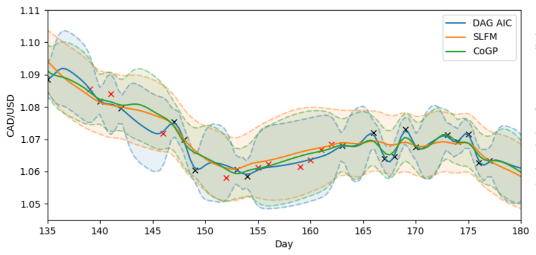
[1]Data available at http://mulan.sourceforge.net/datasets-mtr.html (Andromeda), http://fx.sauder.ubc.ca (Exchange), https://sites.google.com/site/goovaertspierre/pierregoovaertswebsite/download/jura-data (Jura)



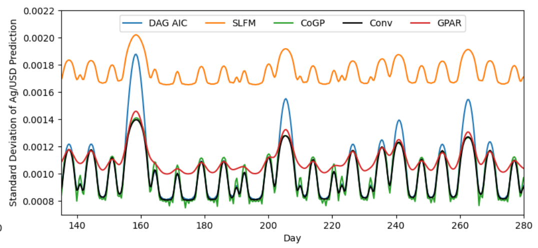
(a) Excerpt from prediction of Au/USD



(b) Excerpt from prediction of Ag/USD



(c) Excerpt from prediction of CAD/USD



(d) Standard deviation of prediction of Ag/USD

Figure 2: Comparison of MOGP performance on the Exchange data set. Black/red crosses indicate training/testing data.

between days 155 and 163. Using the DAG-GP model, the crest is learned for silver because of the troughs on either side, and silver is a direct parent of gold, passing on the prediction of a similar crest. Figure 2c shows CAD, which is learned to be an ancestor of gold and silver in the DAG, so no crest is predicted. In the SLFM and CoGP models, all outputs share generating latent processes, and as a result the crest is smoothed over, because it appears in some but not all outputs.

Outside of selected areas like the crest, the predicted means of the DAG-GP model are similar to those predicted using the CoGP and convolutional models with optimal numbers of latent processes. The DAG-GP model reaches this result using 6 latent processes and 12 linear parameters λ_{mn}. In contrast, the optimized CoGP model requires 10 latent processes with 24 linear parameters, and the convolutional model requires 6 latent processes and 36 linear parameters. Furthermore, determination of the optimal number of latent processes required training and testing multiple CoGP and convolutional models, which was not necessary when using the DAG-GP model.

In the vicinity of training data, the DAG-GP model produces a similar predicted variance profile to the more complex CoGP and convolutional models, as shown in figure 2d. Further away from training data, the predicted variance of the simpler DAG-GP model more rapidly rises. The larger variance is physical, as it better matches the deviation of the test data from the predicted mean, resulting in higher test data likelihood.

7 Conclusions

MOGP models are subject to negative transfer of information between output dimensions that are truthfully conditionally independent. To counteract negative transfer, we propose the DAG-GP model, in which a Gaussian directed acyclic graph is constructed between the outputs. Previous MOGP methods have used arbitrarily correlated outputs or hand-coded structures, but ours is the first to allow outputs to follow an arbitrary DAG, and to learn that DAG from the data. The DAG structure is learned by downselecting training data to produce an approximately independent and identically distributed data set, to which an existing structure learning algorithm is directly applied. Structure learning is fast compared to the time required to train an MOGP, and it removes the need to train on multiple structures like other MOGP methods. When using covariance kernels with multiple length scales, experiments on real world data sets showed that the DAG-GP model uses fewer parameters, better shares trends between outputs when they only appear in some dimensions, and constructs more realistic confidence bounds, thereby decreasing mean prediction error and increasing likelihood over state of the art MOGP methods.

References

Hirotugu Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

Mauricio Alvarez and Neil D Lawrence. Sparse convolved gaussian processes for multi-output regression. In *Advances in neural information processing systems*, pp. 57–64, 2009.

Mauricio A Álvarez and Neil D Lawrence. Computationally efficient convolved multiple output gaussian processes. *The Journal of Machine Learning Research*, 12:1459–1500, 2011.

Mauricio A Álvarez, Lorenzo Rosasco, and Neil D Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, 4(3):195–266, 2012.

Animashree Anandkumar, Daniel Hsu, Adel Javanmard, and Sham Kakade. Learning linear bayesian networks with latent variables. In *International Conference on Machine Learning*, pp. 249–257, 2013.

Edwin V Bonilla, Kian M Chai, and Christopher Williams. Multi-task gaussian process prediction. In *Advances in neural information processing systems*, pp. 153–160, 2008.

Phillip Boyle and Marcus Frean. Dependent gaussian processes. In *Advances in neural information processing systems*, pp. 217–224, 2005.

James Cussens, Matti Järvisalo, Janne H Korhonen, and Mark Bartlett. Bayesian network structure learning with integer programming: Polytopes, facets and complexity. *Journal of Artificial Intelligence Research*, 58:185–229, 2017.

Andreas Damianou and Neil Lawrence. Deep gaussian processes. In *Artificial Intelligence and Statistics*, pp. 207–215, 2013.

Antonino Freno and Edmondo Trentin. *Hybrid Random Fields*, chapter 2.4. Springer, 2011.

Nir Friedman. Learning belief networks in the presence of missing values and hidden variables. In *Proc. 14th International Conference on Machine Learning*, pp. 125–133. Morgan Kaufmann, 1997.

Dan Geiger and David Heckerman. Learning gaussian networks. In *Uncertainty Proceedings 1994*, pp. 235–243. Elsevier, 1994.

Pierre Goovaerts et al. *Geostatistics for natural resources evaluation*. Oxford University Press on Demand, 1997.

Evaggelos V Hatzikos, Grigorios Tsoumakas, George Tzanis, Nick Bassiliades, and Ioannis Vlahavas. An empirical study on sea water quality prediction. *Knowledge-Based Systems*, 21(6): 471–478, 2008.

Andre G Journel and Charles J Huijbregts. *Mining geostatistics*, volume 600. Academic press London, 1978.

Marc C Kennedy and Anthony O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

Neil D Lawrence, Guido Sanguinetti, and Magnus Rattray. Modelling transcriptional regulation using gaussian processes. In *Advances in Neural Information Processing Systems*, pp. 785–792, 2007.

Gayle Leen, Jaakko Peltonen, and Samuel Kaski. Focused multi-task learning in a gaussian process framework. *Machine learning*, 89(1-2):157–182, 2012.

Haitao Liu, Jianfei Cai, and Yew-Soon Ong. Remarks on multi-output gaussian process regression. *Knowledge-Based Systems*, 144:102–121, 2018.

Arman Melkumyan and Fabio Ramos. Multi-kernel gaussian processes. In *Twenty-second international joint conference on artificial intelligence*, 2011.

Trung V Nguyen, Edwin V Bonilla, et al. Collaborative multi-output gaussian processes. In *UAI*, pp. 643–652, 2014.

Judea Pearl et al. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 2000.

James Requeima, William Tebbutt, Wessel Bruinsma, and Richard E Turner. The gaussian process autoregressive regression model (gpar). In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1860–1869, 2019.

Basil Saeed, Anastasiya Belyaeva, Yuhao Wang, and Caroline Uhler. Anchored causal inference in the presence of measurement error. In *Conference on Uncertainty in Artificial Intelligence*, pp. 619–628. PMLR, 2020.

Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

Eleftherios Spyromitros-Xioufis, Grigorios Tsoumakas, William Groves, and Ioannis Vlahavas. Multi-target regression via input space expansion: treating targets as inputs. *Machine Learning*, 104(1):55–98, 2016.

Erik B Sudderth, Martin J Wainwright, and Alan S Willsky. Embedded trees: Estimation of gaussian processes on graphs with cycles. *IEEE Transactions on Signal Processing*, 52(11):3136–3150, 2004.

YW Teh, M Seeger, and MI Jordan. Semiparametric latent factor models. In *AISTATS 2005-Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, 2005.

JA Vargas-Guzmán, AW Warrick, and DE Myers. Coregionalization by linear combination of nonorthogonal components. *Mathematical Geology*, 34(4):405–419, 2002.

Arun Venkitaraman, Saikat Chatterjee, and Peter Handel. Gaussian processes over graphs. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5640–5644. IEEE, 2020.

Jay M Ver Hoef and Ronald Paul Barry. Constructing and fitting models for cokriging and multivariable spatial prediction. *Journal of Statistical Planning and Inference*, 69(2):275–294, 1998.

Martin J Wainwright, Erik B Sudderth, and Alan S Willsky. Tree-based modeling and estimation of gaussian processes on graphs with cycles. In *Advances in neural information processing systems*, pp. 661–667, 2001.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Andrew Wilson and Ryan Adams. Gaussian process kernels for pattern discovery and extrapolation. In *International conference on machine learning*, pp. 1067–1075, 2013.

Kun Zhang, Mingming Gong, Joseph Ramsey, Kayhan Batmanghelich, Peter Spirtes, and Clark Glymour. Causal discovery in the presence of measurement error: Identifiability conditions. 2017.

A Conditional independence in the DAG-GP model

The conditional independence statements between outputs in the DAG-GP model follow from its representation in (13). We prove the conditional independence statements hold through the following theorem.

Theorem 1 (Theorem 1.4.1 of Pearl et al. (2000))**.** *Consider the set of possibly correlated random variables* $\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_D\}$*, jointly independent random variables* $\{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_D\}$*, and equations* $\{g_1, g_2, \dots, g_D\}$ *related by the structural equation model*

$$\mathbf{z}_m = g_m(\mathbf{z}_{pa(m)}, \mathbf{c}_m).$$

Construct a graph by drawing directed edges from each of the $\mathbf{z}_{pa(m)}$ to $\mathbf{z}_m$. If the resultant graph is acyclic, then the resultant distribution $p(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_D)$ admits the following conditional independence relationships for each m,

$$\mathbf{z}_m \perp\!\!\!\perp \mathbf{z}_{nd(m)} \mid \mathbf{z}_{pa(m)}.$$

Proof. The proof by Pearl et al. (2000) is based on d-separation in directed acyclic graphs. Instead, we present a proof based on the representation of the joint distribution.

Since each $\mathbf{z}_m$ depends only on its parents and an independent noise term $\mathbf{c}_m$, the distribution $p(\mathbf{z}_1, \mathbf{c}_1, \mathbf{z}_2, \mathbf{c}_2, \ldots, \mathbf{z}_D, \mathbf{c}_D)$ decomposes as

$$p(\mathbf{z}_1, \mathbf{c}_1, \mathbf{z}_2, \mathbf{c}_2, \ldots, \mathbf{z}_D, \mathbf{c}_D) = \prod_{m=1}^{D} p(\mathbf{z}_m, \mathbf{c}_m \mid \mathbf{z}_{pa(m)}). \tag{18}$$

$\mathbf{c}_m$ terms may be marginalized to yield

$$p(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_D) = \prod_{m=1}^{D} p(\mathbf{z}_m \mid \mathbf{z}_{pa(m)}). \tag{19}$$

The graph connecting all $\mathbf{z}_m$ is acyclic, so the parents of all nondescendants of $\mathbf{z}_m$ are disjoint from the descendants of $\mathbf{z}_m$. If $\mathbf{z}_{nd(m)}$ refers to all nondescendants that are not elements of $\mathbf{z}_{pa(m)}$, then the descendants of $\mathbf{z}_m$ can be marginalized to give

$$p(\mathbf{z}_m, \mathbf{z}_{pa(m)}, \mathbf{z}_{nd(m)}) = p(\mathbf{z}_m \mid \mathbf{z}_{pa(m)}) \prod_{i \in pa(m)} p(\mathbf{z}_i \mid \mathbf{z}_{pa(i)}) \prod_{j \in nd(m)} p(\mathbf{z}_j \mid \mathbf{z}_{pa(j)}), \tag{20}$$

$$p(\mathbf{z}_{pa(m)}, \mathbf{z}_{nd(m)}) = \prod_{i \in pa(m)} p(\mathbf{z}_i \mid \mathbf{z}_{pa(i)}) \prod_{j \in nd(m)} p(\mathbf{z}_j \mid \mathbf{z}_{pa(j)}). \tag{21}$$

Dividing (20) by (21) yields

$$p(\mathbf{z}_m | \mathbf{z}_{pa(m)}, \mathbf{z}_{nd(m)}) = p(\mathbf{z}_m | \mathbf{z}_{pa(m)}), \tag{22}$$

which is the required statement of conditional independence. □

Corollary 1.1. *In the DAG-GP model, $f_m(\boldsymbol{x}_i) \perp\!\!\!\perp f_{nd(m)}(\boldsymbol{x}_i) \mid f_{pa(m)}(\boldsymbol{x}_i)$ for any $\boldsymbol{x}_i$.*

Proof. The DAG-GP model uses a structural equation model $f_m(\boldsymbol{x}_i) = u_m(\boldsymbol{x}_i) + \sum_{n \in pa(m)} \lambda_{mn} f_n(\boldsymbol{x}_i)$ across an acyclic graph. This expression continues to hold with all $f(\boldsymbol{x}_j)$ marginalized for $\boldsymbol{x}_j \neq \boldsymbol{x}_i$ All $u_m(\boldsymbol{x}_i)$ are jointly independent, so the assumptions of theorem 1 apply with $\mathbf{z}_m = f_m(\boldsymbol{x}_i)$ and $\mathbf{c}_m = u_m(\boldsymbol{x}_i)$. □

Corollary 1.2. *In the DAG-GP model, $f_m(\boldsymbol{X}) \perp\!\!\!\perp f_{nd(m)}(\boldsymbol{X}) \mid f_{pa(m)}(\boldsymbol{X})$.*

Proof. Theorem 1 places no restriction on whether the random variables may be vector valued. $u_m(\boldsymbol{X})$ follow jointly independent Gaussian processes, and so are jointly independent random vectors. A structural equation model holds, given by

$$f_m(\boldsymbol{X}) = u_m(\boldsymbol{X}) + \sum_{n \in pa(m)} \lambda_{mn} \boldsymbol{I} f_n(\boldsymbol{X}). \tag{23}$$

The graph drawn between all $f_m(\boldsymbol{X})$ is acyclic, and so the assumptions of theorem 1 apply with $\mathbf{z}_m = f_m(\boldsymbol{X})$ and $\mathbf{c}_m = u_m(\boldsymbol{X})$. □

The conditional independence statements do not depend on the linearity of the structural equations used in the DAG-GP model. In the absence of linearity, the joint distribution is no longer Gaussian distributed and exact inference may not be possible.

The conditional independence statements in corollary 1.2 do not depend on the use of a local structural equation model. For example, $\lambda_{mn}\boldsymbol{I}$ in (23) may be replaced with a non-diagonal matrix in order to capture non-local correlations between an output and its parents. However in this case corollary 1.1, which is fundamental to learning DAG structure, would not apply.