

Multi-stage process modeling using Gaussian Processes

Saksham Kiroriwal

Kognitive Industrielle Systeme (KIS)
Fraunhofer IOSB, Germany
saksham.kiroriwal@iosb.fraunhofer.de

Abstract

In multi-stage processes, dependence on noisy observations of the intermediates is a problem to overcome to predict the outputs accurately. This requires a Multi-Stage Gaussian Process (MSGP)- a modeling idea to incorporate such intermediate observations, considering various observation likelihoods effectively. The MSGP may further boost predictive performance by indirectly observing the multi-stage process by adopting Directed Acyclic Graph (DAG) architecture and Variational Inference (VI) methods. Such a model would use the prior information and increase the accuracy of inference, making Bayesian optimization and prediction effective in situations where one can hardly make direct observations.

1 Introduction

In most practical applications, processes are not confined to one stage but usually involve many stages with associated intermediate outputs. This, in turn, calls for a more sophisticated modeling approach to capturing the details of the real-world system. Traditional approaches to modeling nonlinear input-output relationships typically use single black box GPs [12, 18, 4]. More recently, the work by [2, 1, 13, 11] showed improved modeling and optimization results when the intermediate observations were included.

Most existing GPNs adopt a graph structure, where every node represents a process stage with recorded intermediate outputs. Current models often assume independent training of nodes. This is restrictive and can only deal with noise-free observations, not noisy observations. We propose a new conceptual framework, called the Multi-Stage Gaussian Process (MSGP), which will be able to incorporate the noisy intermediate observations robustly to improve inference. By extending established inference methods from Stochastic Variational GP (SVGP) [7]. MSGP will likely facilitate more effective Bayesian optimization. This paper discusses the broader idea of MSGP, why it is required, and how it could be achieved.

2 Multi-stage process

In complex systems, a multi-stage stochastic process, denoted as \mathbf{M} , is composed of B interconnected subprocesses $M_{(1)}, \dots, M_{(B)}$. Each subprocess $M_{(b)}$ can be expressed using function $\mathbf{g}_{(b)}$. Each subprocess accepts some adjustable parameters $\mathbf{s}_{(b)}$ and outputs from parent subprocesses $M_{\rho(b)}$. The process function operation yields outputs $\tilde{\mathbf{g}}_{(b)}$. These outputs are further influenced by stochastic noise $\tilde{\eta}_{(b)}$. This noise follows a distribution $p(\eta_{(b)}|\mathbf{s}_{(b)})$, capturing the inherent randomness of each subprocess. An example process can be shown as

$$\tilde{\mathbf{g}}_{(1)} = \mathbf{g}_{(1)}(\mathbf{s}_{(1)}) + \tilde{\eta}_{(1)}; \tilde{\mathbf{g}}_{(2)} = \mathbf{g}_{(2)}(\mathbf{s}_{(2)}, \tilde{\mathbf{g}}_{(1)}) + \tilde{\eta}_{(2)},$$

where $\tilde{\eta}_{(b)} \sim p(\eta_{(b)}|\mathbf{s}_{(b)}) \forall \{1, \dots, B\}$.

Another way to express the stochasticity of the process is using a function space. Using this definition, the process \mathbf{M} can be redefined as

$$\tilde{\mathbf{g}}_{(1)} \sim p(\mathbf{g}_{(1)}; \mathbf{s}_{(1)}); \tilde{\mathbf{g}}_{(2)} \sim p(\mathbf{g}_{(2)}; \{\mathbf{s}_{(2)}, \tilde{\mathbf{g}}_{(1)}\}).$$

Often, the subprocess outputs are observed indirectly, producing observed outputs $\tilde{\mathbf{t}}_{(b)}$, rendering the true outputs latent. This indirect observation can be modeled using a likelihood function. The challenge lies in modeling these final outputs using the adjustable inputs and indirect observations through varied likelihoods while leveraging the known data generation structure of the process \mathbf{M} .

We wish to incorporate a network-based modeling approach to avoid augmenting intermediate observations with inputs and not letting the input dimensionality blow up. This work proposes a Multi-Stage Gaussian Process (MSGP) to infer outputs using indirect observations by treating the latent outputs as a joint normal distribution.

3 Background

In this section, we discuss the existing Gaussian Process framework for a single black-box modeling approach where the inputs $\{\mathbf{s}_{(b)}\} \forall b \in \{1, \dots, B\}$ are concatenated to model the final output $\mathbf{t} = \mathbf{t}_{(B)}$.

3.1 Gaussian Process

A *Gaussian process* (GP) can be viewed as a probability distribution over functions defined on the input domain $\mathcal{S} \subseteq \mathbb{R}^{D_s}$. Conventional GP is used to model scalar values observations and use scalar values functions sampled from the function space. Concretely, it assigns a multivariate normal distribution to the set of function values evaluated at any finite collection of inputs. Formally, one writes

$$g(\mathbf{s}) \sim \mathcal{GP}\left(m(\mathbf{s}), k(\mathbf{s}, \mathbf{s}')\right), \quad (3.1)$$

where $m(\cdot) : \mathbb{R}^{D_s} \rightarrow \mathbb{R}$ is the mean function, and $k(\cdot, \cdot') : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}$ is the covariance (kernel) function.

Posterior Inference

Suppose we observe R data points $\{\mathbf{s}_r, t_r\}_{r=1}^R$, where each $\mathbf{s}_r \in \mathbb{R}^{D_s}$ denotes an input location and $t_r \in \mathbb{R}$ is the corresponding observed output. Let $\mathbf{S} = \{\mathbf{s}_r\}_{r=1}^R$ and $\mathbf{t} = \{t_r\}_{r=1}^R$. We place a GP prior over the latent function values $\mathbf{g} = \{g_r\}_{r=1}^R$, encoded as

$$p(\mathbf{g}; \mathbf{S}) = \mathcal{N}\left(m(\mathbf{S}), k(\mathbf{S}, \mathbf{S}')\right), \quad (3.2)$$

and define the likelihood for each observation t_r as $p(t_r|g_r)$. Using Bayes' rule, the posterior over the latent function values, given all observations, is

$$p(\mathbf{g} | \mathbf{t}; \mathbf{S}) = \frac{p(\mathbf{t} | \mathbf{g}) p(\mathbf{g}; \mathbf{S})}{\int p(\mathbf{t} | \mathbf{g}) p(\mathbf{g}; \mathbf{S}) d\mathbf{g}}. \quad (3.3)$$

This posterior encapsulates how the observed data update the prior GP assumptions.

Marginal Likelihood and Hyperparameter Optimization

The kernel $k(\cdot, \cdot')$ and the mean function $m(\cdot)$ usually include hyperparameters (e.g., length-scale, signal variance). These are often optimized by maximizing the *marginal log-likelihood* (MLL) of the observed data:

$$\mathcal{L}_{\text{GP}} = \sum_{r=1}^R \log \mathbb{E}_{p(g_r; \mathbf{s}_r)} \left[p(t_r | g_r) \right]. \quad (3.4)$$

In the special case of a Gaussian likelihood with additive noise variance σ_t^2 , the marginal distribution $p(\mathbf{t}|\mathbf{S})$ becomes $\mathcal{N}(m(\mathbf{S}), k(\mathbf{S}, \mathbf{S}') + \sigma_t^2 \mathbf{I})$, leading to a closed-form expression for Equation 3.4. However, solving for the exact posterior in this scenario requires $\mathcal{O}(R^3)$ operations due to the inversion of an $R \times R$ covariance matrix.

This cubic complexity presents computational challenges when R is large or non-Gaussian likelihoods are used. In those cases, one typically employs approximations such as sparse GPs or variational methods to reduce the computational load while retaining the desirable properties of Gaussian process models [15].

3.2 Stochastic Variational Gaussian Process

When the data size grows large or the observation model is non-Gaussian, optimizing a Gaussian process (GP) by directly maximizing the MLL in Equation 3.4 becomes prohibitive [20]. As a remedy, *Stochastic Variational Gaussian Process* (SVGP) [7, 8] provides a principled approximation scheme that addresses both scenarios.

Inducing Points and Their Distribution

SVGP introduces a finite set of inducing locations $\mathbf{J} = \{\mathbf{j}_k\}_{k=1}^K$, where each $\mathbf{j}_k \in \mathbb{R}^{D_s}$ is drawn from the same domain as the observed inputs $\{\mathbf{s}_r\}$. The corresponding GP function values, called inducing points, at these pseudo-inputs are $\mathbf{w} = \{w_k\}_{k=1}^K$. Their prior distribution under the GP is

$$p(\mathbf{w}; \mathbf{J}) = \mathcal{N}\left(m(\mathbf{J}), k(\mathbf{J}, \mathbf{J}')\right), \quad (3.5)$$

which complements the exact GP prior in Equation 3.2. To facilitate variational inference, one also specifies a Gaussian variational distribution for these inducing variables,

$$q(\mathbf{w}) = \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{w}}, \boldsymbol{\Sigma}_{\mathbf{w}}\right).$$

Variational Approximation

Because \mathbf{w} and \mathbf{g} (the GP values at the observed inputs) come from the same underlying GP, one can write their joint Gaussian distribution by combining Equations 3.2 and 3.5 (see also [14]). Based on this joint model, the SVGP framework defines a variational posterior for \mathbf{g} as follows:

$$\begin{aligned} q(\mathbf{g}; \mathbf{S}, \mathbf{J}) &= \int p(\mathbf{g} | \mathbf{w}; \mathbf{S}, \mathbf{J}) q(\mathbf{w}) d\mathbf{w} \\ &= \mathcal{N}\left(\boldsymbol{\mu}_{q(\mathbf{g})}, \boldsymbol{\Sigma}_{q(\mathbf{g})}\right), \end{aligned} \quad (3.6)$$

where the closed-form expressions for the mean and covariance are given by the following formulae

$$\begin{aligned} \boldsymbol{\mu}_{q(\mathbf{g})} &= m(\mathbf{S}) + \boldsymbol{\alpha}(\mathbf{S})^\top \left(\boldsymbol{\mu}_{\mathbf{w}} - m(\mathbf{J})\right), \\ \boldsymbol{\Sigma}_{q(\mathbf{g})} &= k(\mathbf{S}, \mathbf{S}) - \boldsymbol{\alpha}(\mathbf{S})^\top \left(k(\mathbf{J}, \mathbf{J}) - \boldsymbol{\Sigma}_{\mathbf{w}}\right) \boldsymbol{\alpha}(\mathbf{S}), \\ \boldsymbol{\alpha}(\mathbf{S}) &= k(\mathbf{J}, \mathbf{J})^{-1} k(\mathbf{J}, \mathbf{S}). \end{aligned} \quad (3.7)$$

By conditioning only on K inducing points (with $K \ll R$), we are able to keep the computational costs tractable. This is even the case for large datasets or complex likelihood models.

Variational Objectives

To optimize both the GP hyperparameters and the variational parameters, one maximizes an Evidence Lower BOund (ELBO) [8]:

$$\mathcal{L}_{\text{SVGP, ELBO}} = \sum_{r=1}^R \mathbb{E}_{q(g_r)} [\log p(t_r | g_r)] - \beta \text{KL}\left(q(\mathbf{w}) \parallel p(\mathbf{w})\right),$$

where KL is the Kullback–Leibler divergence. Monte Carlo estimation is typically used to approximate the expectation term.

Predictive Log Likelihood

Alternatively, the Parametric Predictive GP Regressor (PPGPR) [10] proposes maximizing a Predictive Log Likelihood (PLL):

$$\mathcal{L}_{\text{SVGP, PLL}} = \sum_{r=1}^R \log \mathbb{E}_{q(g_r)} \left[p(t_r | g_r) \right] - \beta \text{KL}\left(q(\mathbf{w}) \parallel p(\mathbf{w})\right),$$

and reports improved predictive performance in some settings. However, the required expectation does not admit a closed-form solution for non-conjugate likelihoods, and various approximations must be employed [8, 9].

3.3 Deep Gaussian Process

A *Deep Gaussian Process* (DGP) [3] extends standard Gaussian processes by stacking multiple GPs in a hierarchical structure. Consider a model with N layers, where the j^{th} layer outputs a vector $\mathbf{g}^{(j)} \in \mathbb{R}^{D_j}$. Each layer’s outputs serve as inputs to the subsequent layer. Formally, if $\mathbf{g}_r^{(j-1)}$ denotes the output of layer $j - 1$ for the r^{th} data point, then it becomes the input to layer j , yielding output $\mathbf{g}_r^{(j)}$. Unlike a single-layer GP, this hierarchical setup induces a non-Gaussian marginal distribution at the final layer, making closed-form inference based on the MLL intractable.

Doubly Stochastic Variational Inference

A well-known technique for approximate inference in DGPs is *doubly stochastic variational inference* [17], which generalizes the SVGP approach to multiple layers. As in single-layer SVGP, one introduces a set of inducing locations

and corresponding inducing points for each layer. For layer j , let $\mathbf{J}^{(j-1)} \in \mathbb{R}^{K(j) \times D_{j-1}}$ denote the inducing locations, with $K(j)$ being the number of inducing points for that layer and $\mathbf{W}^{(j)} \in \mathbb{R}^{K(j) \times D_j}$ the corresponding function values under the GP prior. By assuming independence across layers [17], each layer’s marginal depends only on the previous layer.

Following Equations 3.6 and 3.7, one writes the variational distribution at the final layer N as

$$q\left(\mathbf{g}_r^{(N)} \mid \mathbf{g}_r^{(N-1)}, \mathbf{J}^{(N-1)}\right) = \int \prod_{j=1}^N q\left(\mathbf{g}_r^{(j)} \mid \mathbf{g}_r^{(j-1)}, \mathbf{W}^{(j)}, \mathbf{J}^{(j-1)}\right) d\mathbf{g}_r^{(N-1)}, \quad (3.8)$$

where we treat $\mathbf{g}_r^{(0)} = \mathbf{s}_r$ as the original input, and $q(\mathbf{g}_r^{(N)})$ is understood as a distribution rather than a single scalar. Because this setup involves a nested composition of distributions, an exact evaluation of the likelihood term is not feasible. Instead, one resorts to Monte Carlo (MC) approximations for the inner expectations [17], sampling $\tilde{\mathbf{g}}_r^{(j)} \sim q(\mathbf{g}_r^{(j)})$ layer by layer.

Variational Objective

Given the layered structure, the ELBO for the DGP is derived by summing over all observations and penalizing the divergence between the variational and prior distributions for the inducing points of each layer. Specifically,

$$\begin{aligned} \mathcal{L}_{\text{DGP, ELBO}} &= \sum_{r=1}^R \mathbb{E}_{q(\mathbf{g}_r^{(N)})} \left[\log p\left(\mathbf{t}_r \mid \mathbf{g}_r^{(N)}\right) \right] \\ &\quad - \beta \sum_{j=1}^N \text{KL}\left(q(\mathbf{W}^{(j)}) \parallel p(\mathbf{W}^{(j)})\right). \end{aligned} \quad (3.9)$$

This objective can be optimized via gradient-based methods, with MC sampling used for the expectation term in non-conjugate settings. By repeatedly sampling through the layers—hence the phrase “doubly stochastic”—one obtains an effective approximate posterior that captures multiple levels of latent structure.

4 Gaussian Process Networks and Related Work

Gaussian Process Networks (GPN)

The so-called *Gaussian Process Network* was first proposed in [5] and extended in [6] for the problem of learning a Bayesian network structure in a Gaussian process framework. These early formulations focus on structure learning about how nodes are connected in a directed acyclic graph (DAG) without considering inference, given noisy observations of intermediate subprocesses. A different line of work, *Gaussian Process Regression Networks* (GPRN), was proposed by [5, 21], where each node in a neural network–like graph and its outputs are modeled via linear combinations of node outputs in conjunction with Gaussian processes. However, GPRNs constitute yet another approach to multi-output regression. Thus, as they do not specifically consider intermediate observations, their problem setting cannot be similar to the one examined here.

A more recent usage of GPNs as *surrogate models* is discussed by [2, 1]. Each subprocess $M^{(b)}$ is treated as a Gaussian process:

$$g^{(b)}(\cdot) \sim \mathcal{GP}^{(b)}(m^{(b)}(\cdot), k^{(b)}(\cdot, \cdot')),$$

and the associated noisy measurement is given by the likelihood $p(t^b | g^{(b)})$. Because each node’s GP is assumed independent given its observed input-output pairs, one can employ standard closed-form marginal log-likelihood [15] to train each GP individually. Monte Carlo (MC) sampling propagates the final predictions through the network.

Despite being computationally convenient, this strategy leads to three main limitations:

1. **Noisy Observations vs. Latent Inputs:** In practice, the child nodes often depend on the *latent* parent outputs, not the noisy observations of the parent. However, the GPN framework provides the child nodes with the observed noisy values, valid only under noise-free conditions.
2. **Deterministic Input Assumption:** Every output of each parent node in GP is a distribution rather than a fixed number. At training, GPNs rely on their closed-form marginal log-likelihood, which depends on a

deterministic input provided to the child GP. For prediction, MC sampling is conducted instead, resulting in a discrepancy between what has been trained and what is inferred.

3. **Restriction to Gaussian Likelihood:** Since the approach relies on exact marginal log-likelihood, it is restricted to Gaussian observation models. This is too restrictive in practice where the noise could be non-Gaussian, and the real-world process might require amortized likelihoods due to high-dimensional intermediate observations.

Models in [19, 13] partially overcome the first issue by explicitly modeling latent values but still relying on independent node inference via exact marginal log-likelihood, leaving the second and third constraints unresolved. Moreover, GPN-based frameworks presented in [1, 19, 2] are mostly surrogate models for Bayesian optimization with intermediate data rather than general-purpose inference methods.

Gaussian Process Autoregressive Regression (GPAR)

An alternative GPN variant, known as *Gaussian Process Autoregressive Regression* (GPAR), has been introduced by [16]. GPAR sequentially models each observed output by treating previous outputs (in a fixed or greedily chosen order) as inputs to downstream GPs in an autoregressive fashion. Although GPAR demonstrates strong multi-output predictive performance, it does not fully resolve the second and third limitations above nor provides a scalable solution for selecting the order of outputs. Furthermore, when the underlying structure is a DAG (rather than a simple chain), it remains unclear how to accommodate more complex dependencies while retaining GPAR’s flexibility.

5 Discussion

Following the variational inference techniques from Section 3, we propose a multi-stage Gaussian process that can mitigate the major limitations of the current state-of-the-art GPN. Unlike the GPN, which uses the exact marginal log-likelihood while directly feeding the noisy observations from the parental

nodes into the child processes, the new model would incorporate a variational formulation capable of naturally handling stochastic outputs from parent nodes and non-Gaussian likelihoods. In particular, a variational posterior would be assigned for each subprocess using inducing points, similar to the case of SVGP or DGP. There are also hierarchical dependencies between layers, which thus allows MSGP to model observation noise disentangled from true latent outputs and use Monte Carlo samples for training and prediction phases. This ensures that the same way of treating uncertainty is followed throughout all the steps of the network. Besides, MSGP naturally could support high-dimensional intermediate observations by extending the VI methods for amortized inference strategies and avoiding rigid assumptions for closed-form marginal likelihood. It would thus provide a flexible, scalable surrogate modeling tool applicable to most real-world multi-stage systems where the intermediate measurements may be noisy, non-Gaussian, or both while keeping the desirable properties of Bayesian inference.

References

- [1] Virginia Aglietti et al. “Causal bayesian optimization”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 3155–3164.
- [2] Raul Astudillo and Peter Frazier. “Bayesian optimization of function networks”. In: *Advances in neural information processing systems* 34 (2021), pp. 14463–14475.
- [3] Andreas Damianou and Neil D Lawrence. “Deep gaussian processes”. In: *Artificial intelligence and statistics*. PMLR. 2013, pp. 207–215.
- [4] P. Frazier and Jialei Wang. “Bayesian optimization for materials design”. In: *arXiv: Machine Learning* (2015). DOI: 10.1007/978-3-319-23871-5_3.
- [5] Nir Friedman and Iftach Nachman. “Gaussian process networks”. In: *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*. UAI’00. Stanford, California: Morgan Kaufmann Publishers Inc., 2000, pp. 211–219. ISBN: 1558607099.

- [6] Enrico Giudice, Jack Kuipers, and Giusi Moffa. “A Bayesian take on Gaussian process networks”. In: *Advances in Neural Information Processing Systems* 36 (2024).
- [7] James Hensman, Nicolo Fusi, and Neil D Lawrence. “Gaussian processes for big data”. In: *arXiv preprint arXiv:1309.6835* (2013).
- [8] James Hensman, Alexander Matthews, and Zoubin Ghahramani. “Scalable variational Gaussian process classification”. In: *Artificial Intelligence and Statistics*. PMLR. 2015, pp. 351–360.
- [9] Martin Jankowiak, Geoff Pleiss, and Jacob Gardner. “Deep sigma point processes”. In: *Conference on uncertainty in artificial intelligence*. PMLR. 2020, pp. 789–798.
- [10] Martin Jankowiak, Geoff Pleiss, and Jacob Gardner. “Parametric gaussian process regressors”. In: *International conference on machine learning*. PMLR. 2020, pp. 4702–4712.
- [11] Saksham Kiroriwal et al. “Joint Parameter and State-Space Modelling of Manufacturing Processes using Gaussian Processes”. In: *IEEE International Conference on Industrial Informatics* (2024).
- [12] R. Kontar, Shiyu Zhou, and J. Horst. “Estimation and monitoring of key performance indicators of manufacturing systems using the multi-output Gaussian process”. In: *International Journal of Production Research* 55 (2017), pp. 2304–2319. DOI: 10.1080/00207543.2016.1237791.
- [13] Shunya Kusakawa et al. “Bayesian optimization for cascade-type multi-stage processes”. In: *Neural Computation* 34.12 (2022), pp. 2408–2431.
- [14] Felix Leibfried et al. “A tutorial on sparse Gaussian processes and variational inference”. In: *arXiv preprint arXiv:2012.13962* (2020).
- [15] Carl Edward Rasmussen. “Gaussian processes in machine learning”. In: *Summer school on machine learning*. Springer, 2003, pp. 63–71.
- [16] James Requeima et al. “The gaussian process autoregressive regression model (gpar)”. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 1860–1869.

- [17] Hugh Salimbeni and Marc Deisenroth. “Doubly stochastic variational inference for deep Gaussian processes”. In: *Advances in neural information processing systems* 30 (2017).
- [18] Syusuke Sano et al. “Application of Bayesian Optimization for Pharmaceutical Product Development”. In: *Journal of Pharmaceutical Innovation* (2019), pp. 1–11. DOI: 10.1007/s12247-019-09382-8.
- [19] Scott Sussex, Anastasiia Makarova, and Andreas Krause. “Model-based causal Bayesian optimization”. In: *arXiv preprint arXiv:2211.10257* (2022).
- [20] Michalis Titsias. “Variational learning of inducing variables in sparse Gaussian processes”. In: *Artificial intelligence and statistics*. PMLR. 2009, pp. 567–574.
- [21] Andrew Gordon Wilson, David A Knowles, and Zoubin Ghahramani. “Gaussian process regression networks”. In: *arXiv preprint arXiv:1110.4411* (2011).