Constrained Gaussian Process for Signal Integrity applications using Variational Inference

Thong Nguyen[#], Bobi Shi[#], Hanzhi Ma[*], Er-ping Li[*], Andreas Cangellaris[#], Jose Schutt-Aine[#]
[#]University of Illinois Urbana-Champaign, USA
[*]Zhejang University, China
{tnnguye3, bobishi2, cangella, jesa}@illinois.edu, {mahanzhi, 4liep}@zju.edu.cn

***Abstract* — Surrogate modeling with Gaussian Process is effective for problems where data is expensive to query. By construction, a vanilla Gaussian Process model uses a Gaussian likelihood whose support is $\mathbb{R}$. This means the resulted model could generate non-physical values in certain cases. For instance, a negative-valued eye height in high-speed channel simulation can be generated. In this paper, a beta likelihood is used to enforce the non-negative constraint of the underlying mapping. Due to the non-Gaussian likelihood, the regression model is no longer analytical, the posterior is intractable and approximated using variational Bayesian inference. A channel simulation example is used to demonstrate that the approximate Gaussian Process approach successfully avoids generating negative eye heights when used in a Monte-Carlo simulation.**

***Keywords* — Gaussian Process, Variational Inference, High-speed Channel simulation, Signal Integrity, Surrogate modeling**

I. Introduction

A Gaussian Process (GP) [1] is a stochastic process, any finite number of samples of which follow a multi-dimensional Gaussian distribution. In [2], [3], exact GPs have been introduced and used as surrogate models for different applications such as RF microwave filter and digital high-speed link. Since exact GPs assume a Gaussian noise model, the marginal likelihood is also Gaussian, the predictive distribution can be obtained analytically. For problems where the output values are far away from non-physical values such as modeling the bandpass filter in [2], there is little concern for non-physical values generated by the GPs. However, for problems such as eye diagram prediction in channel simulations, the eye width and eye height are bounded between zero and a maximum value, the chance of getting values violating these bounds are much more significant and could be problematic for Monte-Carlo simulations done on the surrogate model. One remedy to resolve the non-physical values when sampling from the surrogate model is to discard them from the sample pool before doing any statistical analysis. In this paper, as another remedy for this problem, we propose to enforce bound constraints by using the beta likelihood. Since the beta likelihood is not conjugate to Gaussian prior, the posterior distribution is intractable and must be approximated. We choose to implement the approximation using variational Bayes.

The paper is structured as follows: Section II reviews the theory of GP in Bayesian framework and variational Bayes theory. Section III presents an example where a constrained GP was used to predict the eye height of a high-speed link. Compared with the original GP, the constrained GP used in this example generates output values that are well bounded as intended, Section IV discusses the result and potential extension of this work.

II. Variational Inference for Gaussian Process Regression

A. Gaussian Process Regression (GPR)

In exact single-output GP regression, given a set of data $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(i)}, y^{(i)}\right), i = 1, 2, ..., N\right\}$ of N pairs of $d-$dimensional vector-valued input $\boldsymbol{x}^{(i)} \in \mathbb{R}^d$ and function-valued output $y^{(i)} \in \mathcal{Y} \subset \mathbb{R}$ such that:

$$y = f(\boldsymbol{x}) + \epsilon \tag{1}$$

where $\epsilon \sim \mathcal{N}\left(0, \sigma^2\right)$ is the Gaussian observation noise, GP makes prediction, y_*, on a test point $\boldsymbol{x_*}$ by sampling from the posterior

$$p(y_* | \boldsymbol{x}_*, \mathcal{D}) = \int_\theta p(y_* | \boldsymbol{x}_*, \boldsymbol{\theta})\, p(\boldsymbol{\theta} | \mathcal{D})\, \mathrm{d}\boldsymbol{\theta} \tag{2}$$

where $\boldsymbol{\theta}$ is the hyper-parameter vector. The hyper-parameter posterior is given by Bayes' rule

$$p(\boldsymbol{\theta} | \mathcal{D}) = \frac{p(\mathcal{D} | \boldsymbol{\theta})\, p(\boldsymbol{\theta})}{\int_\theta p(\mathcal{D} | \boldsymbol{\theta})\, p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}} \tag{3}$$

In implementations, the denominator of (3), a.k.a the *evidence*, is the biggest challenge. It is often a high dimensional integral, hence, intractable. For exact GPs, (3) was never calculated, because everything was assumed Gaussian, the analytical form of $p(\boldsymbol{\theta} | \mathcal{D})$ is given by [1], [2]

$$p(\boldsymbol{y_*} | \boldsymbol{x}_*, \mathcal{D}) = \mathcal{N}(\boldsymbol{\mu_*}, \boldsymbol{\Sigma_*}) \tag{4}$$

where

$$\boldsymbol{\mu_*} = \boldsymbol{K_{tr}} \left(\boldsymbol{K_{rr}} + \sigma^2 \boldsymbol{I}\right)^{-1} \boldsymbol{y} \tag{4a}$$

$$\boldsymbol{\Sigma_*} = \boldsymbol{K_{tt}} - \boldsymbol{K_{tr}} \left(\boldsymbol{K_{rr}} + \sigma^2 \boldsymbol{I}\right)^{-1} \boldsymbol{K_{tr}} \tag{4b}$$

$\boldsymbol{K_{tt}}$, $\boldsymbol{K_{tr}}$ and $\boldsymbol{K_{rr}}$ are the kernel matrices whose i, j-th element is calculated by evaluating a kernel function, $k(\cdot, \cdot) : \mathbb{R}^d \to \mathbb{R}$, using 2 data points, $K^{(ij)} = k\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right)$. The subscript $\boldsymbol{t}$ stands for test data while $\boldsymbol{r}$ stands for training data. The first and second subscript in a kernel matrix indicate which data set the i^{th} and j^{th} data point come from, respectively.

Popular kernel functions can be found in [1], [4]. The marginal likelihood

$$p\left(\mathcal{D}|\,\boldsymbol{\theta}\right)=\prod_{i=1}^{N}p\left(y^{(i)}\Big|\,\boldsymbol{x},\boldsymbol{\theta}\right) \tag{5}$$

is maximized to find the hyper-parameters of the GP.

B. Variational Inference with non-Gaussian GP

In probabilistic modeling, when a likelihood function is non-analytical, approximation methods are needed to arrive at the solution for the hyper-parameter learning. The work in [5] proposes to use variational inference (VI) as an alternative for traditional approximation algorithms such as expected maximization. Let q be a variational distribution [5], [6]. In the following, subscript $*$ refers to the predictive points. The log-likelihood of the GP prediction is

$$\begin{aligned}\log p\left(\boldsymbol{y}_{*}|\,\boldsymbol{\theta}\right) &= \log\int p\left(\boldsymbol{y}_{*},\boldsymbol{f}_{*}|\,\boldsymbol{\theta}\right)\mathrm{d}\boldsymbol{f}_{*}\\ &\geq \int q\left(\boldsymbol{f}_{*}\right)\log\frac{p\left(\boldsymbol{y}_{*},\boldsymbol{f}_{*}|\,\boldsymbol{\theta}\right)}{q\left(\boldsymbol{f}_{*}\right)}\mathrm{d}\boldsymbol{f}_{*}\end{aligned} \tag{6}$$

Let

$$\mathcal{L}\left(\boldsymbol{\theta}\right)=\int q\left(\boldsymbol{f}_{*}\right)\log\frac{p\left(\boldsymbol{y}_{*},\boldsymbol{f}_{*}|\,\boldsymbol{\theta}\right)}{q\left(\boldsymbol{f}_{*}\right)}\mathrm{d}\boldsymbol{f}_{*} \tag{7}$$

be the lower bound of $\log p\left(y|\,\boldsymbol{\theta}\right)$. $\mathcal{L}\left(\theta\right)$ can be further rewritten as

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{\theta}\right) &= \int q\left(\boldsymbol{f}_{*}\right)\log\left[p\left(\boldsymbol{y}_{*}|\,\boldsymbol{\theta}\right)\frac{p\left(\boldsymbol{f}_{*}|\,\boldsymbol{y}_{*},\boldsymbol{\theta}\right)}{q\left(\boldsymbol{f}_{*}\right)}\right]\mathrm{d}\boldsymbol{f}_{*} &\text{(7a)}\\ &= \log p\left(\boldsymbol{y}_{*}|\,\boldsymbol{\theta}\right)\int q\left(\boldsymbol{f}_{*}\right)\mathrm{d}\boldsymbol{f}_{*}\\ &\quad+\int q\left(\boldsymbol{f}_{*}\right)\log\frac{p\left(f|\,\boldsymbol{y}_{*},\boldsymbol{\theta}\right)}{q\left(\boldsymbol{f}_{*}\right)}\mathrm{d}\boldsymbol{f}_{*} &\text{(7b)}\\ &= \log p\left(\boldsymbol{y}_{*}|\,\boldsymbol{\theta}\right)-KL\left(q\left(\boldsymbol{f}_{*}\right)\|\,p\left(\boldsymbol{f}_{*}|\,\boldsymbol{y}_{*},\boldsymbol{\theta}\right)\right) &\text{(7c)}\end{aligned}$$

where

$$KL\left(q\left(\boldsymbol{f}_{*}\right)\|\,p\left(\boldsymbol{f}_{*}|\,\boldsymbol{y}_{*},\boldsymbol{\theta}\right)\right)=-\int q\left(\boldsymbol{f}_{*}\right)\log\frac{p\left(\boldsymbol{f}_{*}|\,\boldsymbol{y}_{*},\boldsymbol{\theta}\right)}{q\left(\boldsymbol{f}\right)}\mathrm{d}\boldsymbol{f} \tag{7d}$$

is the Kullback-Leibler (KL) divergence between q and p, a non-negative quantity, which measures the difference between $q\left(\boldsymbol{f}_{*}\right)$ and $p\left(\boldsymbol{f}_{*}|\,\boldsymbol{y}_{*},\boldsymbol{\theta}\right)$. Since $KL\left(q\left(\boldsymbol{f}_{*}\right)\|\,p\left(\boldsymbol{f}_{*}|\,\boldsymbol{y}_{*},\boldsymbol{\theta}\right)\right)\geq 0$. It is clear that $\log p\left(y|\,\boldsymbol{\theta}\right)\geq\mathcal{L}\left(\boldsymbol{\theta}\right)\ \forall\boldsymbol{\theta}$. Minimizing $KL\left(q\left(\boldsymbol{f}_{*}\right)\|\,p\left(\boldsymbol{f}_{*}|\,\boldsymbol{y}_{*},\boldsymbol{\theta}\right)\right)$ is same as minizing the gap between the true posterior and q, subsequently making q the best approximation for the true posterior. Using the result from [7]

$$q\left(\boldsymbol{f}_{*}\right)=\mathcal{N}\left(\boldsymbol{f}_{*}|\,\mathrm{m},\mathrm{S}\right) \tag{8}$$

where $\mathrm{m}\in\mathbb{R}^{N}$ and $\mathrm{S}\in\mathbb{R}^{N\times N}$ are called the variational parameters and are learnt during the training along with other hyper-parameters of the original GP kernel. In this paper, VI is implemented by Pyro [8], a probabilistic programming framework written in Python. Gradient calculations for the optimization process in Pyro are handled by automatic differentiation which is an advantage by saving time and effort to manually derive the gradient for each different function. This allows prototyping to happen much faster. Different model forms and distribution functions can be mixed and matched to create the best model.

In this paper, we will be using the scaled rational basis function (RBF) kernel for any GP models

$$k\left(\boldsymbol{x},\boldsymbol{x'}\right)=\eta\exp\left(\frac{\|\boldsymbol{x}-\boldsymbol{x'}\|_{2}^{2}}{2\ell^{2}}\right) \tag{9}$$

where η and ℓ are kernel hyper-parameters. We use a beta distribution for the noise model to enforce the bound constraint. A beta distribution of a random variable $0<z<1$ is parameterized by the shape parameters, $\alpha>0$ and $\beta>0$, given by

$$p\left(z|\,\alpha,\beta\right)=Beta\left(\alpha,\beta\right)=\frac{\Gamma\left(\alpha+\beta\right)}{\Gamma\left(\alpha\right)\Gamma\left(\beta\right)}z^{\alpha-1}\left(1-z\right)^{\beta-1} \tag{10}$$

where $\Gamma\left(n\right)=\left(n-1\right)!$ for $n\in\mathbb{Z}^{+}$ is the gamma function. Figure 1 shows the probability density function (pdf) of the beta distribution with different values of α and β. It can be seen that a beta distribution with $\alpha=\beta=2$ or $\alpha=2$, $\beta=5$ can be a good candidate for the noise model of GPs and maintain the bound constraint on the output value.

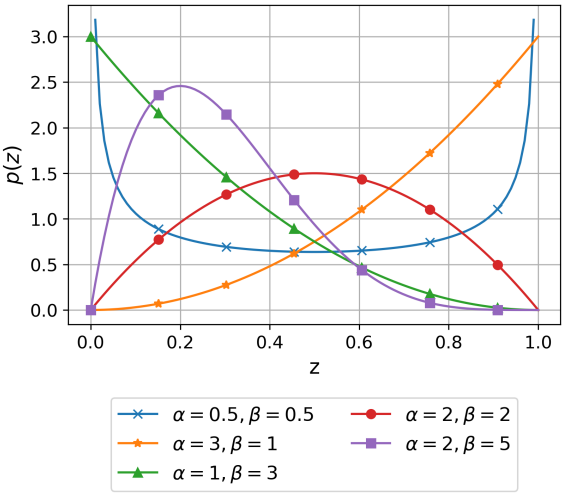

Fig. 1. Beta distribution parameterized by α and β.

There are two minor tweaks that need to be done so that (10) can be used in GP regression. First, since z is bounded between 0 and 1, the data needs to be normalized into this interval as well. This is trivial because normalization is often done as a pre-processing step to increase numerical stability for most regression algorithms anyway. As normalization is a bijective operation, i.e. it has perfect one-to-one correspondence between the actual data and the normalized values, it does not affect the meaning of the mapping between the input and output that needs to be learnt. Second, to link the noiseless GP, f, to (10), we define

$$\alpha=\sigma_{sig}\left(f\right)s,\ \ \beta=\left(1-\sigma_{sig}\left(f\right)\right)s \tag{11}$$

where $\sigma_{sig}\left(\cdot\right)$ is the sigmoid function and $s>0$ is a scaling factor. The main purpose of using $\sigma_{sig}\left(\cdot\right)$ is to convert f from

$\mathbb{R}$ into $(0,1)$. With α, β defined in (11), their domain is still $(0,\infty)$ yet f can affect them and they can be infered during the minimization of the ELBO, i.e. α, β are included in $\boldsymbol{\theta}$ above. In brief, for exact GPs, the likelihood of the data is given by

$$p\left(y|\, f\right) = \mathcal{N}\left(f, \sigma^2\right) \tag{12}$$

while for bounded GP, the likelihood is

$$p\left(y|\, f\right) = Beta\left(\sigma_{sig}\left(f\right)s, \left(1-\sigma_{sig}\left(f\right)\right)s\right) \tag{13}$$

III. Example

In this section, we present a familiar example of predicting the eye height of a high-speed channel. The channel is a die-to-die, coupled strip-line configuration on both (transmitter, TX, and receiver, RX) sides, built on an organic interposer. The link includes various micro-vias transitioning between different layers in an embedded bridge [9]. The inputs are the geometry of the link and the output is particularly the eye height. Equalizations are enabled on both sides; equalization settings are also one of the inputs. Figure 2 conceptualizes the setup. The output node labelled V_{RO} is where the eye diagram is measured and used as the output for the regression model.

Fig. 2. Conceptual channel simulation setup

As shown in [3], GP models under full Bayesian learning can converge with much fewer number of samples compared to other surrogate models. In this example, following the observation in [3] regarding the number of samples needed to train a GP, only a set of $N = 50$ samples were uniformly selected to train the GP models. Both exact and bounded GP use scaled RBF kernel as desribed in (9). Adam [10] is used as the optimizer. The same starting learning rate of 0.01 is used for both models to ensure fair comparison.

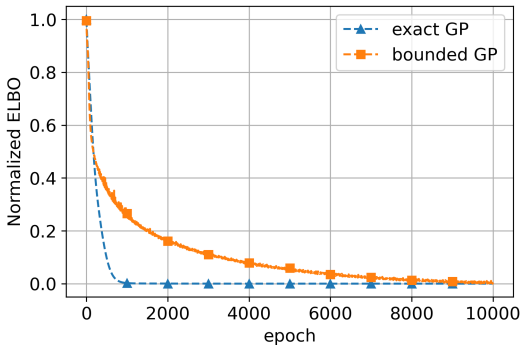

Fig. 3. Normalized training error

The training was done through 10,000 epoch until both models converge although the exact GP model converges long before the bound constrained GP did as shown in Figure 3. It is worth noting that the training errors shown in Figure 3 have been normalized to themselves for better comparison because the two models are derived from different probability distributions and the ELBO absolute values are not a repsentative metric for comparison.

Figure 4 shows the test results from two models compared to that from Monte-Carlo sampling. The eye height values have been normalized to the signal voltage swing, denoted by $\bar{y}$. Not only the bounded GP gives better prediction, its predictions also stay within the bound. The exact GP model generates many values that are negative and larger than the voltage swing ($\bar{y} > 1$) which both violate the physical meaning of the predicted eye height.

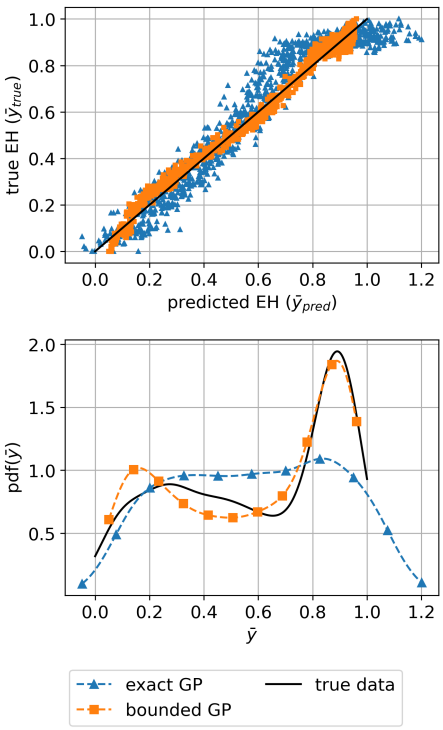

Fig. 4. Predictions from 2 GP models vs. true distribution

IV. Conclusion

To enforce physical constraints on GPR models, it is necessary to use a non-Gaussian likelihood such as the beta likelihood for GPs that generate bounded output values. This paper formulates GPR in the variational framework based on a similar work for classification problems [7] to handle the analytically intractability from using a non-Gaussian likelihood. The algorithm is verified on a high-speed die-to-die channel design. The vanilla exact GP [3] was trained along with the bounded GP presented in this paper with the same training data and training conditions. The result shows that the bounded GP outperforms the exact GP in accuracy and upholds the bound contraint well. The only trade-off is a longer training time for bounded GP. However, due to small training dataset,

the increase in training time is still well acceptable. The model is trained within a few minutes only.

For multi-output systems, multi-output GP should be used instead of multiple single-output GPs for faster convergence. A multi-output GP formulation extension of this work is underway and will be presented in future work along with more examples in both power signal integrity and RF/microwave applications.

ACKNOWLEDGMENT

This material is based upon work supported by the U.S Army Small Business Innovation Research (SBIR) Program office and the U.S. Army Research Office under Contract No.W911NF-16-C-0125.

REFERENCES

[1] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[2] T. Nguyen and J. Schutt-Aine, "Gaussian process surrogate model for variability analysis of RF circuits," in *2020 IEEE Electrical Design of Advanced Packaging and Systems (EDAPS)*, 2020, pp. 1–3.

[3] T. Nguyen, B. Shi, H. Ma, E.-P. Li, X. Chen, A. C. Cangellaris, and J. Schutt-Aine, "Comparative study of surrogate modeling methods for signal integrity and microwave circuit applications," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 11, no. 9, pp. 1369–1379, 2021.

[4] M. A. Álvarez, L. Rosasco, and N. D. Lawrence, "Kernels for vector-valued functions: A review," *Foundations and Trends in Machine Learning*, vol. 4, no. 3, pp. 195–266, 2012.

[5] B. Shi, T. Nguyen, and J. Schutt-Aine, "Variational inference approach to jitter decomposition in high-speed link," in *2020 IEEE 29th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*, 2020, pp. 1–3.

[6] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

[7] J. Hensman, A. Matthews, and Z. Ghahramani, "Scalable Variational Gaussian Process Classification," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Lebanon and S. V. N. Vishwanathan, Eds., vol. 38. San Diego, California, USA: PMLR, 09–12 May 2015, pp. 351–360. [Online]. Available: https://proceedings.mlr.press/v38/hensman15.html

[8] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N. D. Goodman, "Pyro: Deep Universal Probabilistic Programming," *Journal of Machine Learning Research*, 2018.

[9] R. Mahajan, R. Sankman, N. Patel, D.-W. Kim, K. Aygun, Z. Qian, Y. Mekonnen, I. Salama, S. Sharan, D. Iyengar, and D. Mallik, "Embedded multi-die interconnect bridge (emib) – a high density, high bandwidth packaging interconnect," in *2016 IEEE 66th Electronic Components and Technology Conference (ECTC)*, 2016, pp. 557–565.

[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980