Optimize to Generalize in Gaussian Processes: An Alternative Objective Based on the Rényi Divergence

Abstract

We introduce an alternative closed-form objective function α-ELBO for parameter estimation in the Gaussian process ($\mathcal{GP}$) based on the Rényi α-divergence. We use a decreasing temperature parameter α to iteratively deform the objective function during optimization. Ultimately, our objective function converges to the exact likelihood function of $\mathcal{GP}$. At early stages of optimization, α-ELBO can be viewed as a regularizer that smoothens out some unwanted critical points. At late stages, α-ELBO recovers the exact likelihood function that guides the optimizer to solutions that best explain the observed data. Theoretically, we derive an upper bound of the Rényi divergence under the proposed objective and derive convergence rates for a class of smooth and non-smooth kernels. Case studies on a wide range of real-life engineering applications demonstrate that our proposed objective is a practical alternative that offers improved prediction performance over several state of the art inference techniques.

Keywords: Gaussian Process, Rényi Divergence, Annealing, Convergence, Engineering Applications.

1 Introduction

The Gaussian process ($\mathcal{GP}$, also known as kriging) is a collection of random variables, any finite number of which has a joint Gaussian distribution (Sacks et al., 1989; Currin et al., 1991). It is widely used to reconstruct functions based on their scattered observations. In literature, $\mathcal{GP}$s were originally used to tackle regression problems in meteorology (Thompson, 1956; Daley, 1993), geostatistics (Matheron, 1973; Journel and Huijbregts, 1978) and spatial statistics (Ripley, 1981).

Over the past two decades, $\mathcal{GP}$ theory and its application has seen great success in various statistics areas. These include experimental design (Krishna et al., 2020; Gramacy and Apley, 2015; Joseph et al., 2019), Bayesian optimization (Snoek et al., 2012; Rana et al., 2017; Wang et al., 2019), computer experiments and calibration (Kennedy and O'Hagan, 2001; Plumlee et al., 2020; Plumlee, 2019; Sung et al., 2020; Gramacy, 2020; Zhang et al., 2021), reliability (Jones and Johnson, 2009; Wei et al., 2018; Alshraideh and Khatatbeh, 2014), reinforcement learning and bandits (Srinivas et al., 2009) and recently deep learning (Damianou and Lawrence, 2013; Bui et al., 2016; Matthews et al., 2018). Indeed, this success is due to the many desirable properties $\mathcal{GP}$s possess, such as their uncertainty quantification capability and highly flexible model priors where prior knowledge can often be readily accommodated in the mean and covariance function. This progress was also observed on a theoretical level. Matthews et al. (2018) prove that a fully connected, feedforward network will converge to a $\mathcal{GP}$ as the network width goes to infinity. This exciting work has brought upon many insightful connections between $\mathcal{GP}$s and deep neural networks (Jacot et al., 2018; Yang, 2019). Chen et al. (2020) show that mini-batch stochastic gradient descent can be applied in correlated settings, specifically within a $\mathcal{GP}$; a result that allowed scaling $\mathcal{GP}$s far beyond what is currently possible. Wang et al. (2019) derived uniform error bounds for $\mathcal{GP}$s trained using a Matérn kernel. These bounds were then used to find generalization bounds for Bayesian optimization and sequential experimental design (Martinez-Cantin, 2014; Yue and Kontar, 2020b; Tuo and Wang, 2020).

In this paper, we focus on parameter estimation within $\mathcal{GP}$s. **Our overarching goal is to help guide $\mathcal{GP}$ parameter estimation to better solutions that have improved generalization power to new data**. We also highlight the ability to use our model real-life engineering applications.

1.1 *Result Overview and Motivational Example*

Hereby, we start by presenting our main result. **Detailed notation will be further highlighted in the coming sections.**

Given a set of observations $\boldsymbol{Y}$, parameter estimation in $\mathcal{GP}$ is mainly done through optimizing the well-known marginal log-likelihood function $\log p(\boldsymbol{Y})$. However, $\log p(\boldsymbol{Y})$ is often a rugged objective with multiple critical solutions; each with a specific interpretation of the data (Sacks et al., 1989; Currin et al., 1991). Inspired by simulated annealing in optimization (Rose et al., 1990; Dowsland and Thompson, 2012), we propose an iterative estimation procedure based an alternative objective that is a lower bound to $\log p(\boldsymbol{Y})$. More specifically, our alternative objective aims at minimizing the Rényi α-divergence between the true and approximated posterior.

Let $\phi(\boldsymbol{Y}|\cdot,\cdot)$ denote a multivariate Gaussian density for the set of observations $\boldsymbol{Y}$ and

$|\cdot|$ a determinant operator. The alternative objective is given as

$$\mathcal{L}_\alpha(q^*) = \log\{\phi(\boldsymbol{Y}|\boldsymbol{0}, \sigma_\epsilon^2 I + (1-\alpha)\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} + \alpha\boldsymbol{Q})\} + \log|\boldsymbol{I} + \frac{1-\alpha}{\sigma_\epsilon^2}(\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} - \boldsymbol{Q})|^{\frac{-\alpha}{2(1-\alpha)}},$$

where $\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}}$ is the full covariance matrix, σ_ϵ^2 is the noise parameter, $\boldsymbol{Q} = \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{U}}\boldsymbol{K}_{\boldsymbol{U},\boldsymbol{U}}^{-1}\boldsymbol{K}_{\boldsymbol{U},\boldsymbol{f}}$ is a Nyström low-rank approximation of the exact covariance matrix $\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}}$ and $\boldsymbol{U}$ is a collection of latent variables.

Here $\alpha \in [0, 1)$ is a temperature parameter. During the optimization process, we gradually decrease α from 0.99 to 0. **When $\alpha = 0$, we recover the log-marginal likelihood function $\log p(\boldsymbol{Y})$ of the $\mathcal{GP}$.** Besides, $\mathcal{L}_\alpha$ contains a rich family of $\mathcal{GP}$ inference models including variational inference (VI) (when $\alpha \to 1$). At early stages of our optimization when α is close to 1, $\mathcal{L}_\alpha$ has decreased dependence on the exact data likelihood which leads to a smoother loss surface that prevents the optimizer from getting stuck early on, at bad solutions. At late stages, as $\alpha \to 0$, $\mathcal{L}_\alpha$ gradually converges to the original marginal likelihood function and guides the optimizer to solutions that best explain the data. As we will highlight in details in Sec. (4), titled: why use $\mathcal{L}_\alpha$, this new objective intrinsically restricts the complexity of the estimated posterior density and hence offers a trade-off between the model fit and complexity of the latent variable estimators. This trade-off is controlled by α. At early stages, when $\alpha \to 1$, the objective will tend to smooth-out sharp valleys, allowing the solution be driven to locations of high mass. Indeed, it is no surprise to $\mathcal{GP}$ practitioners that very often when optimizing the the marginal likelihood, the noise parameter is estimated to be either very large or very small leading to predictions that follow the mean curve or performing noiseless interpolation. This is an example of anomalies that we aim to help the optimizer avoid early on.

As a motivational example, we create a simple illustrative example. We generate data from a $\mathcal{GP}$ with a radial basis kernel with length parameter 0.1, variance parameter 1.5 and noise parameter 0.01. In Figure 1, we plot $\mathcal{L}_\alpha$ with respect to the length parameter (x-axis) and the variance parameter (y-axis), by fixing the noise parameter to be 0.01. Let us first start with Figure 1(d) where $\alpha = 0$. This represents the exact $\mathcal{GP}$. It can be seen that there are two maxima: $(0.1, 1.5)$ and $(11, 0.4)$. The former one is the global maximal point. If one initializes an optimizer near $(11, 0.4)$, then the optimizer will converge to this local optimal solution and result in suboptimal predictions. On the other hand, for a larger α, the surface of $\mathcal{L}_\alpha$ is smoother and benign. In the example with $\alpha = 0.99$, if one starts with

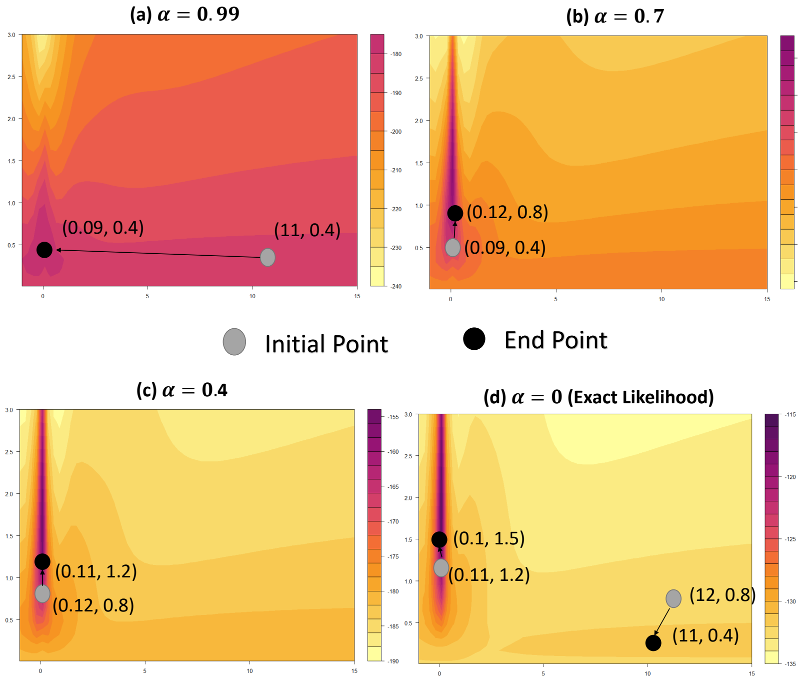

Figure 1: Loss Surfaces of $\mathcal{L}_\alpha$ and trajectories of optimizer with respect to different α. We first set $\alpha = 0.99$ and initialize optimizer in $(11, 0.4)$. The optimizer will move to the optimal solution $(0.09, 0.4)$ after several iterations. As we decrease α, the optimizer will finally converge to the global optimum $(0.1, 1.5)$ (Plot (d)). On the other hand, if we start with $\alpha = 0$ and initial point $(12, 0.8)$, the optimizer will move around the local optimal solution $(11, 0.4)$.

a point close to $(11, 0.4)$, then the optimizer will directly move to the region that contains the maximizer $(0.09, 0.4)$. As we decrease α, the optimizer will move around this region and finally converge to the global optimum $(0.1, 1.5)$. This example conveys a piece of key information: at early stages our proposed objective $\mathcal{L}_\alpha$ is a smoother that smoothens out some unwanted local stationary points and this can help guide to optimal solutions that best explain the data at late stages.

From a theoretical aspect, we show that the Rényi divergence from the true $\mathcal{GP}$ posterior can be made arbitrarily small and derive convergence rates of our bound under a smooth and non-smooth kernel (refer to Sec. 6). More importantly, we illustrate the superior performance of our proposed objective over state-of-the-art $\mathcal{GP}$ approaches on a wide variety of engineering applications and datasets. This demonstrates that $\mathcal{L}_\alpha$ can be a competitive objective function that $\mathcal{GP}$ practitioners can directly use.

1.2 *Organization*

We organize the paper as follows. In Sec. 2, we briefly review related background knowledge. We then provide the Rényi variational objective for $\mathcal{GP}$s in Sec. 3, and its underlying motivation in Sec. 4. The optimization algorithm and predictive distribution are presented in Sec. 5. Theoretical properties of our objective are investigated in Sec. 6. In Sec. 7, we conduct a detailed literature review. In Sec. 8 we provide numerical experiments over a range of engineering applications to demonstrate the advantages of our method. Our experiments include a traffic, battery, house electric, bike and turbofan engine dataset. We conclude our paper in Sec. 9. We note that we defer most derivations to the appendix and only highlight the main results.

2 Notation and Brief Review

We start by introducing some notations and briefly review the Gaussian process. Assume we have collected N training data points $\boldsymbol{Y} = [y_i]_{i=1}^N$ with corresponding D-dimensional inputs $\boldsymbol{X} = [\boldsymbol{x}_i]_{i=1}^N$, where $y_i \in \mathbb{R}$ and $\boldsymbol{x}_i \in \mathbb{R}^D$. We decompose the output as $y_i = f(\boldsymbol{x}_i) + \epsilon_i$, where $f(\cdot)$ is a $\mathcal{GP}$ and $\epsilon_i(\cdot)$ denotes additive noise with zero mean and σ_ϵ^2 variance. The $\mathcal{GP}$ places a prior over functions such that $p(\boldsymbol{f}|\boldsymbol{X}) = \phi(\boldsymbol{f}|\boldsymbol{0}, \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}})$ where $\boldsymbol{f} = [f_1, ..., f_N]$ is a vector of latent function values $f_i = f(\boldsymbol{x}_i)$ and $\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} := \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})$ is a covariance matrix whose entries are determined by a covariance function $k(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta})$ parameterized through $\boldsymbol{\theta}$. Here we note that for notational simplicity we assume zero mean $\mathcal{GP}$ and hereon we neglect conditioning on the input.

Often the end goal of a $\mathcal{GP}$ is to predict output $\boldsymbol{f}^*$ given new inputs $\boldsymbol{X}^*$. To do so, the predictive distribution is attained via $p(\boldsymbol{f}^*|\boldsymbol{Y}) = \int p(\boldsymbol{f}^*|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{Y})d\boldsymbol{f}$ and is given as

$$\boldsymbol{f}^*|\boldsymbol{Y} \sim \mathcal{N}\big(\boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{f}}[\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} + \sigma_\epsilon^2 \boldsymbol{I}]^{-1}\boldsymbol{Y}, \boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{f}^*} - \boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{f}}[\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} + \sigma_\epsilon^2 \boldsymbol{I}]^{-1}\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}^*}\big).$$

Here $p(\boldsymbol{f}^*|\boldsymbol{f})$ is the conditional prior derived from the $\mathcal{GP}$ prior

$$\boldsymbol{f}, \boldsymbol{f}^* \sim \mathcal{N}\Bigg(\mathbf{0}, \begin{pmatrix} \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} & \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}^*} \\ \boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{f}} & \boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{f}^*} \end{pmatrix}\Bigg)$$

and $p(\boldsymbol{f}|\boldsymbol{Y})$ is the posterior of $\boldsymbol{f}$.

Given the predictive distribution, it is clear that good parameter estimation of $(\sigma_\epsilon, \boldsymbol{\theta})$ is imperative to $\mathcal{GP}$s. Perhaps the most popular approach for parameter estimation is by directly maximizing the well-known marginal log-likelihood function $p(\boldsymbol{Y}) = \int p(\boldsymbol{Y}|\boldsymbol{f})p(\boldsymbol{f})d\boldsymbol{f}$. Given that $\boldsymbol{Y}|\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{f}, \sigma_\epsilon^2 \boldsymbol{I})$ the log-marginal likelihood can be written as

$$\mathcal{L}_{marginal} \coloneqq \log p(\boldsymbol{Y}) = \log \phi(\boldsymbol{Y}|\mathbf{0}, \sigma_\epsilon^2 I + \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}}). \tag{1}$$

However, under most well developed covariance functions, $\mathcal{L}_{marginal}$ is highly non-linear and non-convex. Recall the motivating example provided in the Introduction section. As a result, $\mathcal{GP}$s are vulnerable to obtaining parameter estimates with bad generalization power. For instance, it is not uncommon to have critical points in the objective that interpret data as pure noise. The goal of this work is to provide an alternative objective that encourages parameter estimates with improved generalization power to new data.

3 An Alternative $\mathcal{GP}$ Objective

3.1 *Overview*

Instead of directly maximizing the marginal likelihood, we offer an alternative objective that iteratively converges to it. We start by introducing this alternative objective and then discuss its advantages and motivation in Sec. 4.

We follow the general philosophy of variational inference which turns inference into an optimization problem, where an optimal density (q^*), relative to some distance measure, is chosen from a distributional family $(\mathcal{Q})$ to approximate a target distribution - here the posterior of a $\mathcal{GP}$. To do so, we augment our probability space by M continuous latent variables $\boldsymbol{U} = [u(\boldsymbol{z}_i)]_{i=1}^M$ observed at inputs $\boldsymbol{\mathcal{Z}} = [\boldsymbol{z}_1, \cdots, \boldsymbol{z}_M]^T$. We assume that $\boldsymbol{U}$ are drawn from the same $\mathcal{GP}$ prior $p(\boldsymbol{f})$. $\boldsymbol{\mathcal{Z}}$ may be a subset of the input $(\boldsymbol{X})$ or some free parameters, often referred to as pseudo-inputs or inducing points (Snelson and Ghahramani, 2006), to be optimized over.

Notice that from the augmented joint model $p(\boldsymbol{Y}, \boldsymbol{f}, \boldsymbol{U})$ we still reach the same marginal likelihood in (1) through marginalization

$$p(\boldsymbol{Y}) = \int p(\boldsymbol{Y}, \boldsymbol{f}, \boldsymbol{U}) d\boldsymbol{f} d\boldsymbol{U} = \int p(\boldsymbol{Y}|\boldsymbol{f}) p(\boldsymbol{f}|\boldsymbol{U}) p(\boldsymbol{U}) d\boldsymbol{f} d\boldsymbol{U}$$

where $p(\boldsymbol{f}|\boldsymbol{U}) = \phi(\boldsymbol{f}|\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{U}} \boldsymbol{K}_{\boldsymbol{U},\boldsymbol{U}}^{-1} \boldsymbol{U}, \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} - \boldsymbol{Q})$ and $\boldsymbol{Q} = \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{U}} \boldsymbol{K}_{\boldsymbol{U},\boldsymbol{U}}^{-1} \boldsymbol{K}_{\boldsymbol{U},\boldsymbol{f}})$. This hints to the fact that one may let $p(\boldsymbol{U})$ be a distribution that adds some level of flexibility to the model.

3.2 *The α-ELBO*

Exploiting this added flexibility we now take a variational route to approximate the joint posterior over the latent variables $p(\boldsymbol{f}, \boldsymbol{U}|\boldsymbol{Y})$. We use the Rényi's α-divergence as a distance measure. This in turn will lead to an our proposed objective.

The Rényi's α-divergence, first proposed in (Rényi et al., 1961), is a distance measure between two probability density functions (p and q) of a continuous random variable.

$$D_\alpha[q||p] = \frac{1}{\alpha - 1} \log \int q(\boldsymbol{w})^\alpha p(\boldsymbol{w})^{1-\alpha} d\boldsymbol{w}, \alpha \in [0, 1).$$

This divergence contains a rich family of distance measures such as KL-divergence, Bhattacharyya coefficient and χ^2-divergence. Also, $D_\alpha[q||p]$ is continuous and non-decreasing on $\alpha \in [0, 1)$.

In the context of $\mathcal{GP}$s, our goal is to find an optimal posterior density $q^\star$ over the latent variables $\boldsymbol{f}, \boldsymbol{U}$ belonging to some distributional family $\mathcal{Q}$, by minimizing the Rényi α-divergence between the variational density $q(\boldsymbol{f}, \boldsymbol{U})$ and the target posterior $p(\boldsymbol{f}, \boldsymbol{U}|\boldsymbol{Y})$

$$q^*(\boldsymbol{f}, \boldsymbol{U}) \coloneqq \underset{q(\boldsymbol{f},\boldsymbol{U}) \in \mathcal{Q}}{\arg\min}\, D_\alpha[q(\boldsymbol{f}, \boldsymbol{U})||p(\boldsymbol{f}, \boldsymbol{U}|\boldsymbol{Y})].$$

Through some algebraic manipulations (Appendix A.1), one can find that

$$D_\alpha[q(\boldsymbol{f}, \boldsymbol{U})||p(\boldsymbol{f}, \boldsymbol{U}|\boldsymbol{Y})] = \log p(\boldsymbol{Y}) - \mathcal{L}_\alpha(q)\,. \tag{2}$$

Such that

$$\mathcal{L}_\alpha(q) \coloneqq \frac{1}{1-\alpha} \log \mathbb{E}_{q(\boldsymbol{f},\boldsymbol{U})} \left[\left(\frac{p(\boldsymbol{f}, \boldsymbol{U}, \boldsymbol{Y})}{q(\boldsymbol{f}, \boldsymbol{U})} \right)^{1-\alpha} \right]. \tag{3}$$

Following (3) and since $D_\alpha[q||p] \geq 0$, we have that $\mathcal{L}_\alpha(q) \leq \mathcal{L}_{marginal}$ is a lower bound on the log-marginal likelihood and maximizing $\mathcal{L}_\alpha(q)$ will equivalently minimize $D_\alpha[q||p]$.

In order to maximize, $\mathcal{L}_\alpha(q)$, one first needs to find the optimal density. To this end, we exploit a mean-field assumption $q(\boldsymbol{f}, \boldsymbol{U}) = p(\boldsymbol{f}|\boldsymbol{U})q(\boldsymbol{U})$. This in turn poses $q(\boldsymbol{U})$ as the variational density to be optimized.

Under this mean-field assumption (See Appendix A.2),

$$\begin{aligned} \mathcal{L}_\alpha(q) &= \frac{1}{1-\alpha} \log \int p_\alpha(\boldsymbol{Y}|\boldsymbol{U}) q(\boldsymbol{U})^\alpha p(\boldsymbol{U})^{1-\alpha} d\boldsymbol{U} \\ &= \frac{1}{1-\alpha} \log \mathbb{E}_{q(\boldsymbol{U})} p_\alpha(\boldsymbol{Y}|\boldsymbol{U}) q(\boldsymbol{U})^{\alpha-1} p(\boldsymbol{U})^{1-\alpha}, \end{aligned} \tag{4}$$

where $p_\alpha(\boldsymbol{Y}|\boldsymbol{U}) = \int p(\boldsymbol{f}|\boldsymbol{U}) p(\boldsymbol{Y}|\boldsymbol{f})^{1-\alpha} d\boldsymbol{f}$. We slightly abuse notation as $p_\alpha(\boldsymbol{Y}|\boldsymbol{U})$ is not a probability density anymore. Our goal next goal is to find

$$q^*(\boldsymbol{U}) \coloneqq \arg\min_{q(\boldsymbol{U})} D_\alpha[p(\boldsymbol{f}|\boldsymbol{U})q(\boldsymbol{U})||p(\boldsymbol{f}, \boldsymbol{U}|\boldsymbol{Y})] = \operatorname*{argmax}_{q(\boldsymbol{U})} \mathcal{L}_\alpha(q).$$

Fortunately this can be optimally solved in closed form (See Appendix A.3)

$$q^*(\boldsymbol{U}) \propto p_\alpha(\boldsymbol{Y}|\boldsymbol{U})^{1/(1-\alpha)} p(\boldsymbol{U}). \tag{5}$$

Now, plugging in $q^*(\boldsymbol{U})$ to (4), we reach our final result, $\mathcal{L}_\alpha(q^*)$ that denotes the lower bound under the optimal q and is given as (see Appendix A.4)

$$\begin{aligned} &\mathcal{L}_\alpha(q^*) = \log \int p_\alpha(\boldsymbol{Y}|\boldsymbol{U})^{1/(1-\alpha)} p(\boldsymbol{U}) d\boldsymbol{U} = \\ &\qquad \log \Big\{ \mathcal{N}\big(\boldsymbol{0}, \sigma_\epsilon^2 I + (1-\alpha)\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} + \alpha \boldsymbol{Q}\big) \Big\} + \log \big| \boldsymbol{I} + \frac{1-\alpha}{\sigma_\epsilon^2} (\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} - \boldsymbol{Q}) \big|^{\frac{-\alpha}{2(1-\alpha)}}. \end{aligned} \tag{6}$$

Hereon we refer to our bound as the Rényi $\mathcal{GP}$ or the α-ELBO as it is an evidence lower bound (ELBO) on the marginal log-likelihood.

Through scrutinizing (6) we directly observe that the new lower bound unifies components from the exact covariance ($\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}}$) and an approximate covariance ($\boldsymbol{Q} = \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{U}} \boldsymbol{K}_{\boldsymbol{U},\boldsymbol{U}}^{-1} \boldsymbol{K}_{\boldsymbol{U},\boldsymbol{f}}$), also known as the Nyström approximation. Interestingly, when $\alpha = 0$, the marginal log-likelihood is recovered, $\mathcal{L}_0 = \mathcal{L}_{marginal} = \log p(\boldsymbol{Y})$. While, for $\alpha \to 1$, we recover the traditional VI bound obtained from maximizing a KL divergence distance measure. This indeed is a direct consequence of the fact that $\lim_{\alpha \to 1} D_\alpha[p||q] = KL[p||q]$ (Titsias and Lawrence, 2010; Tran et al., 2015; Liu et al., 2018; Yue and Kontar, 2020a).

4 Why use $\mathcal{L}_\alpha$

In this section we shed light on the advantages of our alternative bound $\mathcal{L}_\alpha$.

4.1 *A Regularization Perspective*

To understand why $\mathcal{L}_\alpha(q^*)$ is a promising objective, we first define the lower bounds below. Here $\mathcal{L}_{VI} = \lim_{\alpha\to 1}\mathcal{L}_\alpha(q)$ while $\mathcal{L}_{Jensen}$ is obtained from a direct Jensen's inequality on $\mathcal{L}_\alpha(q)$ (see appendix B).

$$\mathcal{L}_\alpha(q) = \frac{1}{1-\alpha}\log \mathbb{E}_{q(\boldsymbol{U})}\left[p_\alpha(\boldsymbol{Y}|\boldsymbol{U})q(\boldsymbol{U})^{\alpha-1}p(\boldsymbol{U})^{1-\alpha}\right]$$

$$\mathcal{L}_{Jensen} = \underbrace{\frac{1}{1-\alpha}\int q(\boldsymbol{U})\log p_\alpha(\boldsymbol{Y}|\boldsymbol{U})d\boldsymbol{U}}_{\text{Model fit}}\underbrace{-KL[q(\boldsymbol{U})||p(\boldsymbol{U})]}_{\text{Prior regularization}}$$

$$\mathcal{L}_{VI} = \lim_{\alpha\to 1}\mathcal{L}_\alpha(q) = \mathbb{E}_{q(\boldsymbol{f},\boldsymbol{U})}\left[\log p(\boldsymbol{Y}|\boldsymbol{f})\right] - KL[q(\boldsymbol{U})||p(\boldsymbol{U})]$$

where $\mathcal{L}_{marginal} \geq \mathcal{L}_\alpha(q) \geq \mathcal{L}_{Jensen} \geq \mathcal{L}_{VI}$ holds true for any $\alpha \in [0,1)$. We focus on $\mathcal{L}_\alpha(q)$ as $\mathcal{L}_\alpha(q^*)$ is a by-product from optimizing $\mathcal{L}_\alpha(q)$. Note that $KL[q(\boldsymbol{U})||p(\boldsymbol{U})] = KL[q(\boldsymbol{U},\boldsymbol{f})||p(\boldsymbol{U},\boldsymbol{f})]$ since $q(\boldsymbol{f},\boldsymbol{U}) = p(\boldsymbol{f}|\boldsymbol{U})q(\boldsymbol{U})$. Also, $p(\boldsymbol{Y}|\boldsymbol{f}) = p(\boldsymbol{Y}|\boldsymbol{f},\boldsymbol{U})$.

One can directly observe that the bounds mirror the trade-off between the likelihood and prior. For instance, in $\mathcal{L}_{Jensen}$, the first term denotes the model fit and encourages the density of the latent variables to place probability mass on configurations that best explain the observed data; this often induces **a rugged objective with many local critical points each with a specific interpretation of the data**. Whereas the KL term is a regularizer that encourages latent variables close to the prior class. Intrinsically this regularization restricts the complexity of the estimated posterior density and hence offers a trade-off between the fit and complexity of the latent variable estimators. Here α plays the role of controlling this enforced regularization and is data dependent. This in turn allows data to speak for themselves. Note that in a $\mathcal{GP}$, the prior is imposed via the kernel. At early stages of optimization, prior regularization is encouraging kernel hyper-parameters that satisfy both the prior class while at the meantime suiting the observed data. Ultimately, we are optimizing the marginal likelihood to get a good model fit.

The trade-off role of α is critical for generalization to new data (Schölkopf et al., 2002). Indeed, $\mathcal{GP}$ literature has shown that inference via $\mathcal{L}_{marginal}$ can be advantageous on some datasets and VI on others (Lalchand and Rasmussen, 2019; Wang et al., 2019; Chen et al., 2020; Wang et al., 2019). The reason lies in the low rank approximation term $\boldsymbol{Q}$ in $\mathcal{L}_{VI}$ and $\mathcal{L}_{\alpha}$. (Stein, 2014) has proved that a low rank approximation works well when the underlying $\mathcal{GP}$ is smooth (e.g., with the square exponential kernel). Under this scenario, $\mathcal{L}_{VI}$ sometimes work better than $\mathcal{L}_{marginal}$ due to its regularization property. On the other hand, if the underlying $\mathcal{GP}$ is highly non-smooth, $\mathcal{L}_{marginal}$ typically triumphs over $\mathcal{L}_{VI}$.

$\mathcal{L}_{\alpha}$ can achieve the best of both worlds through iterative deforming the objective function, starting from a smooth objective with strong prior regularization and leading to one that emphasizes model fit.

4.2 *A Fractional Posterior Perspective*

Another insight on the advantages of $\mathcal{L}_{\alpha}$ is through fractional posteriors (often referred to as tempered or inexact posteriors) where a likelihood is raised to a some fractional power. Indeed, fractional posteriors have gained renewed interest in recent years within Bayesian statistics due to their empirical success in improving generalization and their robustness to model mis-specifications (Bhattacharya et al., 2019; Miller and Dunson, 2018; Grünwald, 2012). For instance, Miller and Dunson (2018) show that raising likelihood to a well-chosen power induces robustness to a mismatch between the model used and the true generating data process. Specifically, under specific regularity conditions, they show that fractional likelihoods are asymptotically equivalent to conditioning on having the empirical distribution of the observed data close to the empirical distribution of data sampled from the model, with respect to a relative entropy distance.

In our context, using (5), our posterior over the latent variables is given as

$$q^*(\boldsymbol{f}, \boldsymbol{U}) = p(\boldsymbol{f}|\boldsymbol{U})q^*(\boldsymbol{U}) \propto p(\boldsymbol{f}, \boldsymbol{U})\big[\int p(\boldsymbol{f}|\boldsymbol{U})p(\boldsymbol{Y}|\boldsymbol{f})^{1-\alpha}d\boldsymbol{f}\big]^{\frac{1}{1-\alpha}}.$$

Notice that the data likelihood $p(\boldsymbol{Y}|\boldsymbol{f})$ is raised to the power of $1-\alpha$. As $\alpha \to 1$, the likelihood $p(\boldsymbol{Y}|\boldsymbol{f})$ will be flattened and its impact reduced. Therefore, as $\alpha \to 1$, the induced regularization will prefer q to be more spread out across configurations of the hidden variables and not only concentrated around ones that best explain the observed data. This decreased dependence on the likelihood also leads to a smoother loss surface where bad local critical points (often anomalies in the data) are smoothened out. Therefore, when α value is large, the optimizer will not get stuck in those bad local optima and move to a region that contains desirable optimal solutions. As we iteratively decrease α, the surface of $\mathcal{L}_\alpha(q^*)$ will be closer to the marginal likelihood function and the optimizer will finally converge to a solution with strong generalization power.

5 Computation & Prediction

5.1 *Optimization*

We optimize (6) with respect to $(\sigma_\epsilon, \boldsymbol{\theta})$ and optionally $\boldsymbol{\mathcal{Z}}$. During the optimization process, we gradually decrease α from a large value (e.g., 0.99) to 0. This means we start with the variational objective function and end in the original exact likelihood function. Here we emphasize that α is not a tuning parameter. Instead, it is a temperature that decreases over iterations.

5.2 *Computation*

The recent work of Chen et al. (2020) theoretically shows that mini-batch stochastic gradient descent (SGD) can be used for optimizing $\mathcal{GP}$s. This result in turn allows scaling $\mathcal{GP}$s to very large data size regimes. For instance, in their work, a $\mathcal{GP}$ with one million data points can be trained within half an hour on a standard laptop. In the experimental section, we use SGD to estimate all parameters. Here we detail the SGD procedure. Let $\boldsymbol{\theta}_{total}$ be all model parameters that need to be optimized. Denote by ξ the set of indices corresponding to a subset of training data and $\boldsymbol{X}_\xi, \boldsymbol{Y}_\xi$ the respective subset of inputs and outputs indexed by ξ. We can obtain the stochastic gradient of $\mathcal{L}_\alpha(q^*)$ as

$$g(\boldsymbol{\theta}_{total};\xi) := \frac{1}{2}\boldsymbol{Y}_\xi^T \boldsymbol{\Xi}_\xi^{-1}\frac{d\boldsymbol{\Xi}_\xi}{d\boldsymbol{\theta}_{total}}\boldsymbol{\Xi}_\xi^{-1}\boldsymbol{Y}_\xi - \frac{1}{2}\operatorname{Tr}\left(\boldsymbol{\Xi}_\xi^{-1}\frac{d\boldsymbol{\Xi}_\xi}{d\boldsymbol{\theta}_{total}}\right) - \frac{\alpha}{2(1-\alpha)}\operatorname{Tr}\left(A_\xi^{-1}\frac{dA_\xi}{d\boldsymbol{\theta}_{total}}\right),$$

where

$$\begin{aligned}\boldsymbol{\Xi}_\xi &:= \sigma_\epsilon^2 \boldsymbol{I}_\xi + (1-\alpha)\boldsymbol{K}(\boldsymbol{X}_\xi,\boldsymbol{X}_\xi) + \alpha \boldsymbol{K}(\boldsymbol{X}_\xi,\boldsymbol{U})\boldsymbol{K}^{-1}(\boldsymbol{U},\boldsymbol{U})\boldsymbol{K}(\boldsymbol{U},\boldsymbol{X}_\xi),\\ A_\xi &:= \boldsymbol{I}_\xi + \frac{1-\alpha}{\sigma_\epsilon^2}(\boldsymbol{K}(\boldsymbol{X}_\xi,\boldsymbol{X}_\xi) - \boldsymbol{K}(\boldsymbol{X}_\xi,\boldsymbol{U})\boldsymbol{K}^{-1}(\boldsymbol{U},\boldsymbol{U})\boldsymbol{K}\left(\boldsymbol{U},\boldsymbol{X}_\xi\right)).\end{aligned}$$

Therefore, at each iteration t, a subset of training data is taken to update model parameters as

$$\boldsymbol{\theta}_{total}^{(t+1)} = \boldsymbol{\theta}_{total}^{(t)} - \eta^{(t)}(-g(\boldsymbol{\theta}_{total}^{(t)}; \xi^{(t)}))$$

where $\eta^{(t)}$ is the learning rate at iteration t. We summarize all aforementioned optimization procedures in Algorithm 1.

```
Algorithm 1: Optimization Algorithm
  Data: Number of iterations $T$, SGD learning rate schedule $\{\eta^{(t)}\}_{t=1}^{T}$, initial model
        parameter $\boldsymbol{\theta}_{total}^{(0)}$, $\alpha^{(0)} = 0.99$.
  for $t = 0 : (T-1)$ do
      $\alpha^{t+1} = \alpha^{t} - \frac{0.99}{T}$ (other decreasing schedules may be also used);
      Randomly sample a subset of data from $(\boldsymbol{X}, \boldsymbol{Y})$ and denote it as $\xi^{(t)}$;
      $\boldsymbol{\theta}_{total}^{(t+1)} = \boldsymbol{\theta}_{total}^{(t)} - \eta^{(t)}(-g(\boldsymbol{\theta}_{total}^{(t)}; \xi^{(t)}))$ ;
  end
  Return $\boldsymbol{\theta}_{total}^{(T)}$;
```

One drawback of Chen et al. (2020) is that in the prediction phase, using SGD implies only using a batch of the data to perform predictions. This is sub-optimal, specifically since prediction is a one-shot problem unlike the iterative procedure of learning parameters. To overcome this difficulty, we modify and employ the recently proposed algorithm - Blackbox Matrix-Matrix multiplication (Gardner et al., 2018; Wang et al., 2019). The BBMM is an efficient approach to optimize $\mathcal{L}_{marginal}$. This algorithm offers a fast way to calculate the predictive distribution using conjugate gradients (CG), pivoted Cholesky decomposition and parallel computing. Though BBMM is less efficient than SGD, we only require predictions once after parameter estimates are obtained. Therefore, BBMM is a viable method at the prediction stage. Indeed, in our experiments, BBMM acquires predictions within seconds. We defer the detailed BBMM algorithm into Appendix C.

5.3 *Prediction*

Upon estimating all parameters, the predictive distribution $f(\boldsymbol{x}^*)$, at a new input point $\boldsymbol{x}^*$, is given by $\mathcal{N}\left(\mu_{pred}(\boldsymbol{x}^*), \sigma^2_{pred}(\boldsymbol{x}^*)\right)$, where

$$\mu_{pred}(\boldsymbol{x}^*) = \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{X}) \left(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2_\epsilon I\right)^{-1} \boldsymbol{Y}$$

$$\sigma^2_{pred}(\boldsymbol{x}^*) = \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{X}) \left(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2_\epsilon \boldsymbol{I}\right)^{-1} \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{x}^*).$$

The above predictive equations are based on the exact Gaussian process which is eventually recovered as $\alpha \to 0$.

6 Theoretical Properties

In this section, we study the rate of convergence of our algorithm.

6.1 *A Data-dependent Upper Bound*

In order to derive convergence rates, we first need to obtain a data-dependent upper bound on the marginal likelihood. Titsias (2014) provides a bound based on the KL divergence. We can generalize this bound into (details in Appendix D.1)

$$\mathcal{L}_{upper} \geq \mathcal{L}_{marginal} = \log \frac{1}{|2\pi\boldsymbol{\Xi}|^{\frac{1}{2}}} - \frac{1}{2}\boldsymbol{Y}^T \left(\boldsymbol{\Xi} + \alpha \mathrm{Tr}(\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} - \boldsymbol{Q})\boldsymbol{I}\right)^{-1}\boldsymbol{Y}. \tag{7}$$

6.2 *Rate of Convergence*

Given the upper bound, we provide the rate of convergence of the proposed α-ELBO (Appendix D.2).

Theorem 1. *Suppose N data points are drawn independently from input distribution $p(\boldsymbol{x})$ and $k(\boldsymbol{x}, \boldsymbol{x}) \leq v_0, \forall \boldsymbol{x} \in \mathcal{X}$. For $\epsilon > 0$, with probability at least $1 - \delta$,*

$$D_\alpha[q||p] \leq \frac{\alpha}{2\delta(1-\alpha)} \log \left[1 + \frac{1-\alpha}{\sigma^2_\epsilon} \frac{[(M+1)C + 2Nv_0\epsilon]}{N}\right]^N + \alpha \frac{(M+1)C + 2Nv_0\epsilon}{2\delta\sigma^2_\epsilon} \frac{\|\boldsymbol{Y}\|^2}{\sigma^2_\epsilon}.$$

Furthermore, if $\boldsymbol{Y}$ is distributed according to a sample from the prior generative model, then with probability at least $1 - \delta$,

$$D_\alpha[p||q] \leq \alpha \frac{(M+1)C + 2Nv_0\epsilon}{2\delta\sigma^2_\epsilon} + \frac{1}{\delta} \frac{\alpha}{2(1-\alpha)} \log \left[1 + \frac{1-\alpha}{\sigma^2_\epsilon} \frac{[(M+1)C + 2Nv_0\epsilon]}{N}\right]^N.$$

where $C = N\sum_{m=M+1}^{\infty} \lambda_m$ and λ_m are the eigenvalues of the integral operator $\mathcal{K}$ associated to kernel and $p(\boldsymbol{x})$.

To see the value of Theorem 1, we will further simplify derived upper bounds by considering different kernels.

6.3 *Consequences*

Based on Theorem 1, we can derive the convergence rate for both smooth (e.g., the square exponential kernel) and non-smooth (e.g., Matérn) kernels.

6.3.1 *Smooth Kernel*

We will provide a convergence result with the square exponential (SE) kernel. The m-th eigenvalue of kernel operator is $\lambda_m = v\sqrt{2a/A}B^{m-1}$, where $a = 1/(4\sigma_\epsilon^2)$, $b = 1/(2\ell^2)$, $c = \sqrt{a^2 + 2ab}$, $A = a + b + c$ and $B = b/A$. ℓ is the length parameter, v is signal variance and σ_ϵ is the noise parameter. We can obtain $\sum_{m=M+1}^{\infty} \lambda_m = \frac{v\sqrt{2a}}{(1-B)\sqrt{A}}B^M$.

Corollary 2. *Suppose $\|\boldsymbol{Y}\|^2 \le RN$, where R is a constant. Fix $\gamma > 0$ and take $\epsilon = \frac{\delta\sigma_\epsilon^2}{vN^{\gamma+2}}$. Assume the input data is normally distributed and regression in performed with a SE kernel. With probability $1-\delta$,*

$$D_\alpha[p||q] \le 2\alpha\frac{R}{\sigma_\epsilon^2}\frac{1}{N^\gamma} + \frac{1}{\delta}\frac{\alpha}{2(1-\alpha)}\log\left[1 + (1-\alpha)\big(\frac{4\delta}{N^{\gamma+2}}\big)\right]^N,$$

when inference is performed with $M = \frac{(3+\gamma)\log N + \log\eta}{\log(B^{-1})}$, where $\eta = \frac{v\sqrt{2a}}{a\sqrt{A}\sigma_\epsilon^2\delta(1-B)}$.

This corollary is proved in Appendix D.3. It implies that the number of inducing points should be of order $\mathcal{O}(\log N)$ (i.e., sparse). In a high dimensional input space, following a similar proof, we can show that this order becomes $\mathcal{O}(\log^D N)$. Additionally, we can see that the upper bound goes to zero as $N \to \infty$. This implies $D_\alpha[p||q]$ can be arbitrarily small and the approximation of the true posterior becomes more accurate as sample size increases.

6.3.2 *Non-smooth Kernel*

For the Matérn $r + \frac{1}{2}$, $\lambda_m \asymp \frac{1}{m^{2r+2}}$ kernel, where $\asymp$ means "asymptotically equivalent to", we can obtain $\sum_{m=M+1}^{\infty} \lambda_m = \mathcal{O}(\frac{1}{M^{2r+1}})$. Let $\sum_{m=M+1}^{\infty} \lambda_m \le A\frac{1}{M^{2r+1}}$. Then by Theorem

1, we have (Appendix D.4)

$$\alpha\frac{(M+1)N\sum_{m=M+1}^{\infty}\lambda_m+2Nv_0\epsilon}{2\delta\sigma_\epsilon^2}\frac{\|\mathbf{Y}\|^2}{\sigma_\epsilon^2}\leq\frac{\alpha R}{2\delta\sigma_\epsilon^4}\big(\frac{(M+1)N^2A}{M^{2r+1}}+2N^2v_0\epsilon\big).$$

Let $M=N^t$ (t will be clarified shortly) and $2rt-2\geq\gamma$, then $t\geq\frac{\gamma+2}{2r}$. Therefore, we have (Appendix D.4)

$$\frac{\alpha R}{2\sigma_\epsilon^4}\big(\frac{(M+1)N^2A}{M^{2r+1}}+2N^2v_0\epsilon\big)\leq\frac{\alpha R}{N^\gamma\sigma_\epsilon^2}+\frac{\alpha RA}{2\delta\sigma_\epsilon^4N^\gamma}.$$

Another term in the bound can also be simplified as

$$\frac{\alpha}{2(1-\alpha)}\log\left[1+\frac{1-\alpha}{\sigma_\epsilon^2}\frac{[(M+1)C+2Nv_0\epsilon]}{N}\right]^N\leq\frac{\alpha N}{2(1-\alpha)}\log\left[1+(1-\alpha)\big(\frac{A+2\delta}{\sigma_\epsilon^2N^{\gamma+2}}\big)\right].$$

It can be seen that we require more inducing points ($\mathcal{O}(N^t)$) when we are using non-smooth kernels and t decreases as we increase the smoothness (i.e., r) of the Matérn kernel.

7 Literature Overview

Though it is by no means an exhaustive list, recent advances in model estimation for Gaussian processes can be roughly split into five main trends (for further details see the recent survey in Liu et al. (2018)). **First**, sampling methods such as Markov chain Monte Carlo (MCMC) (Gramacy and Lian, 2012; Frigola et al., 2013; Hensman et al., 2015) and Hamiltonian Monte Carlo (Havasi et al., 2018) have been extensively studied. However, a sampling approximation is usually computationally intensive. Notably, a recent comparison study (Lalchand and Rasmussen, 2019) shows that variational inference (VI) can achieve remarkable performance compared to sampling approaches while the former has better theoretical properties and can be fitted into many existing efficient optimization frameworks. **Second**, expectation propagation (EP) (Deisenroth and Mohamed, 2012) is an iterative local message passing method designed for approximate Bayesian inference. Based on this approach, Bui et al. (2017) propose the power EP (PEP) framework to learn $\mathcal{GP}$s and demonstrate that PEP encapsulates a rich family of approximated $\mathcal{GP}$s such as FITC and DTC (Bui et al., 2017). Though accurate and promising, the EP family, in general, is not guaranteed to converge (Bishop, 2006). **Third**, variational inference is an approach to estimate probability densities through efficient optimization algorithms (Hoffman et al., 2013;

Hoang et al., 2015; Blei et al., 2017). It approximates intractable posterior distributions using a tractable distributional family $\mathcal{Q}$. This approximation in turn yields a lower bound that is optimized to learn model parameters. VI has caught the most attention compared to the other approximate inference algorithms due to ease of use and elegant theoretical properties. **Fourth**, there has been a recent push on utilizing GPU acceleration and distributed computing to optimize the log-marginal likelihood in $\mathcal{GP}$s. Such approaches leverage Blackbox Matrix-Matrix multiplication, distributed Cholesky factorization and kernel partitioning (Gardner et al., 2018; Wang et al., 2019). **Lastly**, some literature consider low rank or sparse approximation techniques (Gramacy and Haaland, 2016) and covariance tapering (Furrer et al., 2006; Kaufman et al., 2008). Besides those trends, some notable work also approximate Gaussian process using Vecchia's approximation method (Guinness, 2018) and stochastic partial differential equation approximation methods (Lindgren et al., 2011).

8 Experiments

We benchmark our model with recent state-of-the-art methods: (1) the exact inference procedure for $\mathcal{GP}$s (EGP) (Wang et al., 2019; Chen et al., 2020). This method directly optimizes the exact likelihood function $\mathcal{L}_{\text{marginal}}$. We use SGD to estimate parameters and use BBMM to obtain predictions; (2) the stochastic variational $\mathcal{GP}$ (SGP) (Hoffman et al., 2013; Hensman et al., 2013). This method performs stochastic VI to the exact $\mathcal{GP}$ and optimizes the derived variational lower bound; (3) the power expectation propagation (PEP) (Bui et al., 2017) with optimal tuning α_{PEP} values. Here please note that α_{PEP} is a tuning parameter.

8.1 *A Toy Example*

We first investigate the performance of our method on well-known simulated functions with 1,000 data points in various dimensions. Data is from the Virtual Library of Simulation Experiments (`http://www.sfu.ca/~ssurjano/index.html`). The testing functions are Gramacy & Lee function (GL, $D = 1$), Branin-Hoo function (BH, $D = 2$) and Griewank-D function (GD, $D \geq 2$). For each dataset, we randomly split 60% data as training sets and 40% as testing sets. We set the number of inducing points to be 50. Throughout the experiment, we use the Matérn kernel. For each function, we run our model 30 times with different initial parameters. For each repetition, during the optimization process, we gradually decrease α from 0.99 to 0. The performance of each model is measured by Root Mean Square Error (RMSE).

Table 1: Simulation Results. Outputs are standardized to be mean 0 and variance 1.

RMSE	GL	BH	GD ($D = 4$)
Rényi	**0.001(±0.000)**	**0.009(±0.002)**	**0.020(±0.003)**
EGP	0.003(±0.000)	0.017(±0.003)	0.027(±0.002)
SGP	0.002(±0.000)	0.014(±0.002)	0.033(±0.002)
PEP	0.002(±0.000)	0.018(±0.001)	0.029(±0.003)

We report results from Gramacy & Lee function, Branin-Hoo function and Griewank-4 function in Table 1. The results clearly indicate that our model has the smallest RMSE among all benchmark models. Here, we note that EGP outperforms the SGP in some cases while the opposite happens sometimes. This observation aligns with the conclusion made by Wang et al. (2019): SGP is not necessarily better than EGP and vice versa. Overall, our Rényi objective can deliver improvements compared to other benchmark models. This credits to the additional temperature parameter α. As we have illustrated previously (Sec. 4), the variational objective ($\alpha > 0$) acts like a smoother and more likely pushes optimizer to the global optimal solution of the exact likelihood function ($\alpha = 0$).

8.2 *Real Engineering Data*

Table 2: RMSE of all models on different datasets. The RMSE is calculated over 30 replications with different initial points. The NL values are reported in Appendix F.

Dataset	N	EGP	SGP	PEP (optimal α_{PEP})	Rényi
Bike	17,389	0.221 ± 0.003	0.305 ± 0.001	0.288 ± 0.009	$\mathbf{0.203 \pm 0.001}$
C-MAPSS	33,727	0.633 ± 0.051	0.597 ± 0.055	0.642 ± 0.083	$\mathbf{0.545 \pm 0.032}$
Protein	29,267	0.536 ± 0.012	0.577 ± 0.008	0.539 ± 0.012	$\mathbf{0.500 \pm 0.008}$
Traffic	48,204	0.125 ± 0.001	0.121 ± 0.003	0.124 ± 0.001	$\mathbf{0.119 \pm 0.000}$
Battery	104,046	0.194 ± 0.003	0.305 ± 0.005	0.242 ± 0.014	$\mathbf{0.155 \pm 0.001}$
House Electric	1,311,539	0.053 ± 0.000	0.088 ± 0.005	0.049 ± 0.007	$\mathbf{0.041 \pm 0.001}$

Table 3: RMSE of Rényi $\mathcal{GP}$ with different M values. The batch size is 1152.

House Electric	$M = 128$	$M = 256$	$M = 512$	$M = 1152$
RMSE	0.048 ± 0.001	0.045 ± 0.000	0.048 ± 0.001	0.043 ± 0.001

Table 4: Running Time Comparison (minutes)

Dataset	N	EGP	SGP	PEP	Rényi
House Electric (with learning inducing points)	1,311,539	66.2 ± 0.7	1269.3 ± 9.3	123.1 ± 8.2	129.6 ± 8.5
House Electric (without learning inducing points)	1,311,539	66.2 ± 0.7	64.2 ± 1.3	67.3 ± 1.1	69.6 ± 0.8

We benchmark the Rényi $\mathcal{GP}$ on a range of datasets that include the (1) *Bike*, (2) *Traffic* (3) *Protein* and (4) *House electric* datasets from the UCI data repository (Asuncion and Newman, 2007) (`https://archive.ics.uci.edu/ml/datasets.php`). (5) Battery data from the General Motors Onstar System and (6) *C-MAPSS* aircraft turbofan engines dataset provided by the National Aeronautics and Space Administration (NASA) (`https://ti.arc.nasa.gov/tech/dash/groups/pcoe/`).

The size of these datasets ranges from 17,389 to 1,311,539 data points. For each dataset, we randomly split 60% data as training sets and 40% as testing sets and replicate the experiment 30 times. All data are standardized to have mean 0 and variance 1.

Inference: For α-ELBO, SGP and PEP, we use SGD with batch size 1024 and $M = 1024$ to optimize all hyperparameters (this is the recommended setting for M in Wang et al. (2019); Bui et al. (2017)). For EGP, we use batch size 64 with learning rate 0.01. The number of epoch is set to be 100. **Prediction:** For mBCG algorithm, we use a diagonal-scaling-preconditioning-matrix to stabilize the algorithm and boost convergence speed (Takapoui and Javadi, 2016). In mBCG, the maximum number of iterations is set to be $10N$.

8.3 *Results and Discussion*

Experimental results are reported in Table 2. The performance of each model is measured by RMSE. The RMSE is calculated over 30 experiments with different initial points. We also report the negative loss (NL, or log-likelihood) in Appendix F. Based on Table 2, we can obtain some important insights. **First**, the results indicate that Rényi $\mathcal{GP}$ achieves the smallest RMSE among all benchmarks on all datasets ranging from small data regimes

($N \approx 17,000$) to large data regimes ($N \approx 1,300,000$). **Second**, EGP seems to outperform SGP on some data sets while the opposite happens on others; both while being inferior to PEP and α-ELBO. This confirms that neither marginal or sparse/variational inference is always superior over the other as the extent of regularization needed is data dependent. We also observe that PEP sometimes does not converge as seen in Table 2 where standard deviations for RMSE of PEP is sometimes large. This is not surprising as PEP is a heuristic that currently lacks theoretical backing. **Furthermore**, the advantages of our model become increasingly significant when the sample size increases. This reveals that controlling smoothness of the ELBO is necessary and promising when we have big and high dimensional data. **Lastly**, we report the running time of all models in Table 4. As shown in the Table, the training time for α-ELBO is comparable to PEP on the House Electric dataset with 1.3 million data points (note that all methods are trained with 100 epochs). They are slower than EGP since EGP does not need to learn the distribution of inducing points. When we uniformly distribute inducing points and do not optimize them, the running times are similar to EGP.

8.4 *Different Choice of* M

We conduct a sensitivity analysis with different M on the large scale House Electric dataset from the UCI repository. By employing SGD, the α-ELBO can be trained efficiently using one RTX-2080 GPU. Table 3 reports the RMSE of all models. We use Matérn 3/2 kernel (we also observed that the square exponential kernel delivered similar performance). The results confirm the benefits of the α-ELBO regardless of M.

9 Conclusion

We introduce an alternative objective for obtaining parameter estimates in $\mathcal{GP}$s, based on the Rényi α-divergence. This bound offers a structured balance between model-fit and prior regularization and therefore is capable of controlling the enforced regularization on the objective function. Through many case studies on engineering datasets and applications we demonstrate that our proposed objective is a practical alternative that offers improved prediction performance over several state of the art inference techniques.

References

Alshraideh, H. and E. Khatatbeh (2014). A gaussian process control chart for monitoring autocorrelated process data. *Journal of Quality Technology 46*(4), 317–322.

Asuncion, A. and D. Newman (2007). Uci machine learning repository.

Bhattacharya, A., D. Pati, Y. Yang, et al. (2019). Bayesian fractional posteriors. *Annals of Statistics 47*(1), 39–66.

Bishop, C. M. (2006). *Pattern recognition and machine learning.* springer.

Blei, D. M., A. Kucukelbir, and J. D. McAuliffe (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association 112*(518), 859–877.

Bui, T., D. Hernández-Lobato, J. Hernandez-Lobato, Y. Li, and R. Turner (2016). Deep gaussian processes for regression using approximate expectation propagation. In *International conference on machine learning*, pp. 1472–1481.

Bui, T. D., J. Yan, and R. E. Turner (2017). A unifying framework for gaussian process pseudo-point approximations using power expectation propagation. *The Journal of Machine Learning Research 18*(1), 3649–3720.

Chen, H., L. Zheng, R. Al Kontar, and G. Raskutti (2020). Stochastic gradient descent in correlated settings: A study on gaussian processes. *Neural Information Processing Systems*.

Currin, C., T. Mitchell, M. Morris, and D. Ylvisaker (1991). Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association 86*(416), 953–963.

Daley, R. (1993). *Atmospheric data analysis.* Number 2. Cambridge university press.

Damianou, A. and N. Lawrence (2013). Deep gaussian processes. In *Artificial Intelligence and Statistics*, pp. 207–215.

Deisenroth, M. and S. Mohamed (2012). Expectation propagation in gaussian process dynamical systems. In *Advances in Neural Information Processing Systems*, pp. 2609–2617.

Dowsland, K. A. and J. Thompson (2012). Simulated annealing. *Handbook of natural computing*, 1623–1655.

Frigola, R., F. Lindsten, T. B. Schön, and C. E. Rasmussen (2013). Bayesian inference and learning in gaussian process state-space models with particle mcmc. In *Advances in Neural Information Processing Systems*, pp. 3156–3164.

Furrer, R., M. G. Genton, and D. Nychka (2006). Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics 15*(3), 502–523.

Gardner, J., G. Pleiss, K. Q. Weinberger, D. Bindel, and A. G. Wilson (2018). Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, pp. 7576–7586.

Gramacy, R. B. (2020). *Surrogates: Gaussian process modeling, design, and optimization for the applied sciences*. CRC Press.

Gramacy, R. B. and D. W. Apley (2015). Local gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics 24*(2), 561–578.

Gramacy, R. B. and B. Haaland (2016). Speeding up neighborhood search in local gaussian process prediction. *Technometrics 58*(3), 294–303.

Gramacy, R. B. and H. Lian (2012). Gaussian process single-index models as emulators for computer experiments. *Technometrics 54*(1), 30–41.

Grünwald, P. (2012). The safe bayesian. In *International Conference on Algorithmic Learning Theory*, pp. 169–183. Springer.

Guinness, J. (2018). Permutation and grouping methods for sharpening gaussian process approximations. *Technometrics 60*(4), 415–429.

Havasi, M., J. M. Hernández-Lobato, and J. J. Murillo-Fuentes (2018). Inference in deep gaussian processes using stochastic gradient hamiltonian monte carlo. In *Advances in Neural Information Processing Systems*, pp. 7506–7516.

Hensman, J., N. Fusi, and N. D. Lawrence (2013). Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*.

Hensman, J., A. G. Matthews, M. Filippone, and Z. Ghahramani (2015). Mcmc for variationally sparse gaussian processes. In *Advances in Neural Information Processing Systems*, pp. 1648–1656.

Hoang, T. N., Q. M. Hoang, and B. K. H. Low (2015). A unifying framework of anytime sparse gaussian process regression models with stochastic variational inference for big data. In *ICML*, pp. 569–578.

Hoffman, M. D., D. M. Blei, C. Wang, and J. Paisley (2013). Stochastic variational inference. *The Journal of Machine Learning Research 14*(1), 1303–1347.

Jacot, A., F. Gabriel, and C. Hongler (2018). Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*.

Jones, B. and R. T. Johnson (2009). Design and analysis for the gaussian process model. *Quality and Reliability Engineering International 25*(5), 515–524.

Joseph, V. R., L. Gu, S. Ba, and W. R. Myers (2019). Space-filling designs for robustness experiments. *Technometrics 61*(1), 24–37.

Journel, A. G. and C. J. Huijbregts (1978). *Mining geostatistics*, Volume 600. Academic press London.

Kaufman, C. G., M. J. Schervish, and D. W. Nychka (2008). Covariance tapering for likelihood-based estimation in large spatial data sets. *Journal of the American Statistical Association 103*(484), 1545–1555.

Kennedy, M. C. and A. O'Hagan (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63*(3), 425–464.

Krishna, A., V. R. Joseph, S. Ba, W. A. Brenneman, and W. R. Myers (2020). Robust experimental designs for model calibration. *arXiv preprint arXiv:2008.00547*.

Lalchand, V. and C. E. Rasmussen (2019). Approximate inference for fully bayesian gaussian process regression. *arXiv preprint arXiv:1912.13440*.

Lindgren, F., H. Rue, and J. Lindström (2011). An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73*(4), 423–498.

Liu, H., Y.-S. Ong, X. Shen, and J. Cai (2018). When gaussian process meets big data: A review of scalable gps. *arXiv preprint arXiv:1807.01065*.

Martinez-Cantin, R. (2014). Bayesopt: a bayesian optimization library for nonlinear optimization, experimental design and bandits. *J. Mach. Learn. Res. 15*(1), 3735–3739.

Matheron, G. (1973). The intrinsic random functions and their applications. *Advances in applied probability 5*(3), 439–468.

Matthews, A. G. d. G., M. Rowland, J. Hron, R. E. Turner, and Z. Ghahramani (2018). Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*.

Miller, J. W. and D. B. Dunson (2018). Robust bayesian inference via coarsening. *Journal of the American Statistical Association*.

Plumlee, M. (2019). Computer model calibration with confidence and consistency. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 81*(3), 519–545.

Plumlee, M., C. Erickson, B. Ankenman, and E. Lawrence (2020). Composite grid designs for adaptive computer experiments with fast inference. *Biometrika*.

Rana, S., C. Li, S. Gupta, V. Nguyen, and S. Venkatesh (2017). High dimensional bayesian optimization with elastic gaussian process. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2883–2891. JMLR. org.

Rényi, A. et al. (1961). On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California.

Ripley, B. D. (1981). *Spatial statistics*, Volume 575. John Wiley & Sons.

Rose, K., E. Gurewitz, and G. Fox (1990). A deterministic annealing approach to clustering. *Pattern Recognition Letters 11*(9), 589–594.

Sacks, J., W. J. Welch, T. J. Mitchell, and H. P. Wynn (1989). Design and analysis of computer experiments. *Statistical science*, 409–423.

Schölkopf, B., A. J. Smola, F. Bach, et al. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Snelson, E. and Z. Ghahramani (2006). Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pp. 1257–1264.

Snoek, J., H. Larochelle, and R. P. Adams (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pp. 2951–2959.

Srinivas, N., A. Krause, S. M. Kakade, and M. Seeger (2009). Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*.

Stein, M. L. (2014). Limitations on low rank approximations for covariance matrices of spatial data. *Spatial Statistics 8*, 1–19.

Sung, C.-L., Y. Hung, W. Rittase, C. Zhu, and C. Jeff Wu (2020). A generalized gaussian process model for computer experiments with binary time series. *Journal of the American Statistical Association 115*(530), 945–956.

Takapoui, R. and H. Javadi (2016). Preconditioning via diagonal scaling. *arXiv preprint arXiv:1610.03871*.

Thompson, P. D. (1956). Optimum smoothing of two-dimensional fields 1. *Tellus 8*(3), 384–393.

Titsias, M. and N. D. Lawrence (2010). Bayesian gaussian process latent variable model. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 844–851.

Titsias, M. K. (2014). Variational inference for gaussian and determinantal point processes.

Tran, D., R. Ranganath, and D. M. Blei (2015). The variational gaussian process. *arXiv preprint arXiv:1511.06499*.

Tuo, R. and W. Wang (2020). Kriging prediction with isotropic matern correlations: robustness and experimental designs. *Journal of Machine Learning Research 21*(187), 1–38.

Wang, K. A., G. Pleiss, J. R. Gardner, S. Tyree, K. Q. Weinberger, and A. G. Wilson (2019). Exact gaussian processes on a million data points. *arXiv preprint arXiv:1903.08114*.

Wang, W., R. Tuo, and C. Jeff Wu (2019). On prediction properties of kriging: Uniform error bounds and robustness. *Journal of the American Statistical Association*, 1–27.

Wei, P., F. Liu, and C. Tang (2018). Reliability and reliability-based importance analysis of structural systems using multiple response gaussian process model. *Reliability Engineering & System Safety 175*, 183–195.

Yang, G. (2019). Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *arXiv preprint arXiv:1902.04760*.

Yue, X. and R. A. Kontar (2020a). Joint models for event prediction from time series and survival data. *Technometrics*, 1–10.

Yue, X. and R. A. Kontar (2020b). Why non-myopic bayesian optimization is promising and how far should we look-ahead? a study via rollout. In *International Conference on Artificial Intelligence and Statistics*, pp. 2808–2818. PMLR.

Zhang, Q., P. Chien, Q. Liu, L. Xu, and Y. Hong (2021). Mixed-input gaussian process emulators for computer experiments with a large number of categorical levels. *Journal of Quality Technology 53*(4), 410–420.