This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3157197, IEEE Transactions on Pattern Analysis and Machine Intelligence

Incremental Ensemble Gaussian Processes

Qin Lu*, *Member, IEEE,* Georgios V. Karanikolas*, *Member, IEEE,*
and Georgios B. Giannakis, *Fellow, IEEE*

Abstract—Belonging to the family of Bayesian nonparametrics, Gaussian process (GP) based approaches have well-documented merits not only in learning over a rich class of nonlinear functions, but also in quantifying the associated uncertainty. However, most GP methods rely on a single preselected kernel function, which may fall short in characterizing data samples that arrive sequentially in time-critical applications. To enable *online* kernel adaptation, the present work advocates an incremental ensemble (IE-) GP framework, where an EGP assembler employs an *ensemble* of GP learners, each having a unique kernel belonging to a prescribed kernel dictionary. With each GP expert leveraging the random feature-based approximation to perform online prediction and model update with *scalability*, the EGP assembler capitalizes on data-adaptive weights to synthesize the per-expert predictions. Further, the novel IE-GP is generalized to accommodate time-varying functions by modeling structured dynamics at the EGP assembler and within each GP learner. To benchmark the performance of IE-GP and its dynamic variant in the adversarial setting where the modeling assumptions are violated, rigorous performance analysis has been conducted via the notion of regret, as the norm in online convex optimization. Last but not the least, online unsupervised learning for dimensionality reduction is explored under the novel IE-GP framework. Synthetic and real data tests demonstrate the effectiveness of the proposed schemes.

Index Terms—Gaussian processes, ensemble learning, online prediction, random features, regret analysis

1 Introduction

Gaussian processes (GPs) cross-fertilize merits of kernel methods and Bayesian models to benefit several learning tasks, including regression, classification, ranking, and dimensionality reduction [1]. In GP-based approaches, a Gaussian *prior* is assumed over a learning function $f(\cdot)$ with covariance (kernel) capturing similarities among $\{f(\mathbf{x}_t)\}$ dependent on inputs $\{\mathbf{x}_t\}$. Given observed outputs $\{y_t\}$ linked to the latent function $f(\cdot)$ via the conditionally independent per-datum likelihood $p(y_t|f(\mathbf{x}_t))$, Bayes rule produces the *posterior* distribution of $f(\cdot)$, based on which task-specific inference can be effected on the unseen data. Besides learning functions with rich expressiveness, the Bayesian framework of GP-based approaches further quantifies uncertainty of the function estimate, which is of utmost importance in safety-critical applications. For instance in medical diagnosis [2], human intervention would be called for when machine operated decisions are accompanied by high uncertainty.

In spite of the intriguing performance, applicability of plain-vanilla GPs in the big data regime is discouraged by the cubic computational complexity in the number of training samples [1]. To relieve the scalability issue, various attempts have been made, including efficient numerical operation [3], [4], and structured approximants of the kernel matrix [5], [6]. Of special interest to us is the random feature (RF) based approach, which, leveraging the spectral properties of stationary kernels, converts the nonparametric GP paradigm to a parametric one [7], [8]. Such a parametric approach readily accommodates online processing of data samples [9], which is necessitated in time-critical applications. For instance, the detection of spam emails is performed on an email-by-email basis in real time. Albeit accommodating online operation, the performance of existing RF-based GP methods hinges on the single *preselected* kernel, that may fall short in characterizing upcoming data. Henceforth, online kernel adaptation is essential to real-time decision making.

On the theoretical horizon, to benchmark performance of online approaches, analysis is usually conducted via the notion of regret, the norm in online convex optimization [10] and online learning with experts [11], to combat with the adversarial setting where the generative assumptions are violated. Although several scalable GP approaches have been developed for the online operation [12], [13], regret analysis has not been touch upon except for the plain-vanilla GP [14].

In accordance with the aforementioned desiderata, the goal of the current work is to pursue algorithmic developments of scalable GPs that could enable kernel adaptation to cope with function dynamics in the online scenario, as well as benchmark performance of the resultant approaches via regret-based analysis.

1.1 Related works

To contextualize the current contribution, the following existing works will be outlined.

Batch and online Scalable GPs. Approaches to effect scalability in GPs rely on advanced numerical methods [3], [4], [15], special kernel functions [16], [17], or low rank approximants of the kernel matrix [5], [6], [7], [18]. A well-known low-rank scheme summarizes the T training samples via $q(\ll T)$ pseudo data with inducing inputs that are employed for inference in the testing phase [5], [6], [18]. This global summary amounts to approximating the original GP prior with a kernel matrix having low rank q, thus reducing the complexity of batch computations to $\mathcal{O}(Tq^2)$. Rather than the spatial sampling, another less explored low-rank approach leverages spectral components of shift-invariant kernels to yield the random feature (RF) based kernel approximation [8]. Converting the nonparametric GP prior to a parametric one, the resultant RF-based GP approaches can afford complexity comparable to the inducing points-based approximants [7], [19]. To accommodate time-critical applications, online scalable GP ap-

- *The authors are with Dept. of Electrical and Computer Engineering and Digital Technology Center, University of Minnesota Minneapolis, MN 55455. E-mails: qlu@umn.edu; karan029@umn.edu; georgios@umn.edu*
- ∗ *The first two authors are equally contributed.*

0162-8828 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Minnesota. Downloaded on July 26,2022 at 19:20:57 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3157197, IEEE Transactions on Pattern Analysis and Machine Intelligence

proaches have also been developed when kernel hyperparameters are fixed [9], [20], [21] and are adapted incrementally [12], [13], [22], [23]. Albeit ensuring scalability, these approaches rely on a *single* GP kernel, which may limit expressiveness of the sought function. Also, theoretical analyses that quantify the robustness of the online solvers to the adversarial setting are largely unexplored.

Expert-based GPs. An *ensemble* of (*local* or distributed) GP experts, each relying on a unique kernel to summarize a subset of the training samples, has been leveraged to lower computational complexity, or/and account for nonstationarity of the learning function. Depending on how data samples are distributed and how predictions over experts are aggregated, well-known examples include the naive-local-experts [24], product-of-experts [25], [26], mixture-of-experts [27], [28], and most recently sum-product networks [29] based approaches. In spite of the advantages enjoyed in different lines of work, existing expert-based GP approaches operate in *batch* mode, thus falling short in dealing with time-critical applications that welcome online decision-making.

(Online) Multi-kernel learning. Parallel to the probabilistic GP paradigm, kernel-based learning has also been pursued in the *deterministic* reproducing kernel Hilbert space (RKHS). Faced with inscalability arising from abundance of training data, kernel-based approaches also resort to low-rank approximants of the kernel matrix, including the RF-based approximation [8]. Bypassing kernel selection via cross validation, data-driven multi-kernel learning enjoys well-documented performances; see, e.g., [30], [31]. To further accommodating online operation, scalable kernel-based learning has been investigated for a single [32] as well as for an ensemble of learners [33], [34]. Most recently, online RF-based approaches based on an ensemble of RKHS learners have been reported along with their regret-based performance for static and dynamic settings [34].

GP latent variable model (LVM). Leveraging GPs to model the mapping from the hidden low-dimensional input space to high-dimensional observations, GPLVMs are established probabilistic approaches to *nonlinear* dimensionality reduction [35]. Scalable GPLVMs have been devised by relying on inducing points-based approximations in the batch setting [36], [37], as well as the variational and online variants [37], [38], [39], [40]. However, to the best of our knowledge, RF-based counterparts, in spite of the application in kernel principle component analysis (PCA) (see, e.g. [41]), have not been touched upon in the realm of GPLVMs. Regarding ensemble learning, a GPLVM scheme which can be broadly categorized in this area is [39], where different from the proposed approach the goal is to track the latent state of a dynamical system. Ensemble methods are, nonetheless, more commonplace in the context of probabilistic PCA for *linear* dimensionality reduction; see [42] for the seminal work and [43] for an online variant.

1.2 Contributions

Relative to the aforementioned past works, the present paper aims at bringing together the fields of scalable GPs and online learning with expert advice [11]. The pursuit lies in algorithmic development, as well as performance analysis via the measure of regret to account for the violations of the generative models. The detailed contributions are highlighted as follows.

- **c1)** Towards online kernel adaptation, the present work advocates an incremental (I) approach based on a weighted ensemble (E) of GP learners with scalable RF-based kernel approximations. The novel IE-GP learns the unknown function and jointly adapts to the appropriate EGP kernel on-the-fly.
- **c2)** To cope with learning nonstationary functions, dynamic IE-GP variants have been devised to capture structured dynamics at the EGP assembler and individual GP learners via a hidden Markov model and the state-space models, respectively.
- **c3)** To account for data being adversarially chosen in the online setting, the performances of IE-GP and its dynamic variant are compared with some benchmark functions with data in hindsight via static and switching regret analyses. In both cases, the cumulative regrets over T slots are of order $\mathcal{O}(\log T)$, implying no regret on average.
- **c4)** Complementary to the supervised function learning task, online unsupervised learning for dimensionality reduction (a.k.a. latent variable model) is investigated under the proposed IE-GP paradigm.
- **c5)** Extensive experimental results are provided to validate the merits of the proposed methods in regression, classification and dimensionality reduction tasks.

Relative to the conference precursor [44], the novelty lies in the following four aspects: 1) A switching (S) IE-GP approach is devised by modeling dynamics at the EGP assembler via a first-order Markov chain; 2) The performance of the proposed SIE-GP is analysed via the notion of switching regret; 3) EGP-based online unsupervised learning with RFs for scalability is further explored; 4) Experimental section has been significantly expanded via the inclusion of classification and dimensionality reduction tests.

Notation. Scalars are denoted by lowercase, column vectors by bold lowercase, and matrices by bold uppercase fonts. Superscripts $^\top$ and $^{-1}$ denote transpose, and matrix inverse, respectively; while $\mathbf{0}_N$ stands for the $N \times 1$ all-zero vector; and $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{K})$ for the probability density function (pdf) of a Gaussian random vector $\mathbf{x}$ with mean $\boldsymbol{\mu}$, and covariance matrix $\mathbf{K}$. Subscript "$t+1|\mathbf{t}$" signifies that prediction for slot $t+1$ relies on the *batch* of samples up to and including t, while "$t+1|t$" stands for a single-step predictor. $I(x)$ represents the indicator function, which is 1 if x is true, and 0 otherwise.

2 Preliminaries and Background

As a prelude to our online EGP approach that will also introduce context and notation, this section deals with batch and scalable learning based on a single GP.

2.1 Non-scalable batch GP-based learning

Given data $\{\mathbf{x}_\tau, y_\tau\}$, the goal is to learn a function $f(\cdot)$ that links the $d \times 1$ input $\mathbf{x}_\tau$ with the scalar output y_τ as $\mathbf{x}_\tau \to f(\mathbf{x}_\tau) \to y_\tau$. Postulating f with a GP prior as $f \sim \mathcal{GP}(0, \kappa(\mathbf{x}, \mathbf{x}'))$, where $\kappa(\cdot,\cdot)$ is a kernel function measuring pairwise similarity of any two inputs, the joint prior pdf of function evaluations $\mathbf{f}_t := [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_t)]^\top$ at any inputs $\mathbf{X}_t := [\mathbf{x}_1, \ldots, \mathbf{x}_t]^\top$ is Gaussian distributed as [1]

$$p(\mathbf{f}_t|\mathbf{X}_t) = \mathcal{N}(\mathbf{f}_t; \mathbf{0}_t, \mathbf{K}_t) \quad \forall t \tag{1}$$

where $\mathbf{K}_t$ is a $t \times t$ covariance matrix with (τ, τ')th entry $[\mathbf{K}_t]_{\tau,\tau'} = \mathrm{cov}(f(\mathbf{x}_\tau), f(\mathbf{x}_{\tau'})) := \kappa(\mathbf{x}_\tau, \mathbf{x}_{\tau'})$.

0162-8828 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Minnesota. Downloaded on July 26,2022 at 19:20:57 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3157197, IEEE Transactions on Pattern Analysis and Machine Intelligence

To estimate f, we rely on the observed outputs $\mathbf{y}_t := [y_1, \ldots, y_t]^\top$ that are linked with $\mathbf{f}_t$ via the conditional likelihood $p(\mathbf{y}_t|\mathbf{f}_t, \mathbf{X}_t) = \prod_{\tau=1}^t p(y_\tau|f(\mathbf{x}_\tau))$ that is assumed known. Through Bayes' rule, the latter will yield the posterior $p(\mathbf{f}_t|\mathbf{y}_t, \mathbf{X}_t) \propto p(\mathbf{f}_t|\mathbf{X}_t)p(\mathbf{y}_t|\mathbf{f}_t, \mathbf{X}_t)$. For Gaussian process regression (GPR) the conditional likelihood is assumed normal with mean $\mathbf{f}_t$ and covariance matrix $\sigma_n^2\mathbf{I}_t$, that is, $p(\mathbf{y}_t|\mathbf{f}_t, \mathbf{X}_t) = \mathcal{N}(\mathbf{y}_t; \mathbf{f}_t, \sigma_n^2\mathbf{I}_t)$, which along with the GP prior in (1) yields the Gaussian posterior $p(\mathbf{f}_t|\mathbf{y}_t, \mathbf{X}_t)$. For non-Gaussian likelihoods, sampling or approximate inference techniques will be called for to carry out the analytically intractable posterior $p(\mathbf{f}_t|\mathbf{y}_t, \mathbf{X}_t)$ [1].

Prediction with a single GP. Given training data $\{\mathbf{X}_t, \mathbf{y}_t\}$ and a new test input $\mathbf{x}_{t+1}$, we have from (1) that $p(f(\mathbf{x}_{t+1})|\mathbf{f}_t, \mathbf{X}_t)$ is Gaussian with known mean and covariance. Together with the known posterior $p(\mathbf{f}_t|\mathbf{y}_t, \mathbf{X}_t)$, the so-termed predictive pdf of $f(\mathbf{x}_{t+1})$ can be obtained as [1]

$$p(f(\mathbf{x}_{t+1})|\mathbf{y}_t, \mathbf{X}_t) = \int p(f(\mathbf{x}_{t+1})|\mathbf{f}_t, \mathbf{X}_t)p(\mathbf{f}_t|\mathbf{y}_t, \mathbf{X}_t)d\mathbf{f}_t \quad (2)$$

which is generally non-Gaussian if $p(\mathbf{f}_t|\mathbf{y}_t, \mathbf{X}_t)$ is non-Gaussian, and thus necessitates Monte Carlo (MC) sampling to estimate it. Alternatively, $p(\mathbf{f}_t|\mathbf{y}_t, \mathbf{X}_t)$ can be approximated by a Gaussian, yielding a Gaussian approximation for (2) as well. Of course, $p(f(\mathbf{x}_{t+1})|\mathbf{y}_t, \mathbf{X}_t)$ is Gaussian for GPR, with its mean and covariance matrix available in closed form.

Using the pdf in (2) and the known $p(y_{t+1}|f(\mathbf{x}_{t+1}))$, it is also possible to find the predictive pdf of y_{t+1} as

$$p(y_{t+1}|\mathbf{y}_t, \mathbf{X}_{t+1}) = \int p(y_{t+1}|f(\mathbf{x}_{t+1}))p(f(\mathbf{x}_{t+1})|\mathbf{y}_t, \mathbf{X}_t)df(\mathbf{x}_{t+1}) \quad (3)$$

which generally requires MC sampling or $p(f_{t+1}|\mathbf{y}_t; \mathbf{X}_{t+1})$ to be (at least approximately) Gaussian. Either way, (3) yields the data predictive pdf that fully quantifies the uncertainty of y_{t+1}.

Specifically for GPR, we have

$$p(y_{t+1}|\mathbf{y}_t, \mathbf{X}_{t+1}) = \mathcal{N}(y_{t+1}; \hat{y}_{t+1|\mathbf{t}}, \sigma^2_{t+1|\mathbf{t}}) \quad (4)$$

where the mean and the variance of the predictor are given by [1]

$$\hat{y}_{t+1|\mathbf{t}} = \mathbf{k}_{t+1}^\top(\mathbf{K}_t + \sigma_n^2\mathbf{I}_t)^{-1}\mathbf{y}_t \quad (5a)$$

$$\sigma^2_{t+1|\mathbf{t}} = \kappa(\mathbf{x}_{t+1}, \mathbf{x}_{t+1}) - \mathbf{k}_{t+1}^\top(\mathbf{K}_t + \sigma_n^2\mathbf{I}_t)^{-1}\mathbf{k}_{t+1} + \sigma_n^2 \quad (5b)$$

with $\mathbf{k}_{t+1} := [\kappa(\mathbf{x}_1, \mathbf{x}_{t+1}), \ldots, \kappa(\mathbf{x}_t, \mathbf{x}_{t+1})]^\top$. Clearly, this GP predictor is not scalable, since the complexity $\mathcal{O}(t^3)$ for inverting the $t \times t$ matrix in (5) will become prohibitively high as t grows.

2.2 Scalable RF learning with a single GP

Various attempts have been made to effect scalability in GP-based learning; see, e.g., [6], [7], [18]. Most existing approaches amount to summarizing the training data via a much smaller number of pseudo data with inducing inputs, thereby obtaining a low-rank approximant of $\mathbf{K}_t$ [18]. However, finding the locations of these inducing inputs entails involved optimization procedure. Targeting a low-rank approximant that bypasses such intricate training practice, we rely here on a standardized shift-invariant $\bar{\kappa}(\mathbf{x}, \mathbf{x}') = \bar{\kappa}(\mathbf{x} - \mathbf{x}')$, whose inverse Fourier transform is

$$\bar{\kappa}(\mathbf{x} - \mathbf{x}') = \int \pi_{\bar{\kappa}}(\mathbf{v})e^{j\mathbf{v}^\top(\mathbf{x}-\mathbf{x}')}d\mathbf{v} := \mathbb{E}_{\pi_{\bar{\kappa}}}\left[e^{j\mathbf{v}^\top(\mathbf{x}-\mathbf{x}')}\right] \quad (6)$$

where $\pi_{\bar{\kappa}}$ is the power spectral density (PSD), and the last equality follows after normalizing so that $\pi_{\bar{\kappa}}(\mathbf{v})$ integrates to 1, thus allowing one to view it as a pdf.

Since $\bar{\kappa}$ is real, the expectation in (6) is given by $\mathbb{E}_{\pi_{\bar{\kappa}}}\left[\cos(\mathbf{v}^\top(\mathbf{x} - \mathbf{x}'))\right]$, which, upon drawing a sufficient number, say n_{RF}, of independent and identically distributed (i.i.d.) samples $\{\mathbf{v}_j\}_{j=1}^{n_{\text{RF}}}$ from $\pi_{\bar{\kappa}}(\mathbf{v})$, can be approximated by [1]

$$\check{\bar{\kappa}}(\mathbf{x}, \mathbf{x}') := \frac{1}{n_{\text{RF}}}\sum_{j=1}^{n_{\text{RF}}}\cos\left(\mathbf{v}_j^\top(\mathbf{x} - \mathbf{x}')\right). \quad (7)$$

Define the $2n_{\text{RF}} \times 1$ random feature (RF) vector as [7]

$$\boldsymbol{\phi}_{\mathbf{v}}(\mathbf{x}) := \frac{1}{\sqrt{n_{\text{RF}}}}\left[\sin(\mathbf{v}_1^\top\mathbf{x}), \cos(\mathbf{v}_1^\top\mathbf{x}), \ldots, \sin(\mathbf{v}_{n_{\text{RF}}}^\top\mathbf{x}), \cos(\mathbf{v}_{n_{\text{RF}}}^\top\mathbf{x})\right]^\top \quad (8)$$

which allows us to rewrite $\check{\bar{\kappa}}$ in (7) with $\check{\bar{\kappa}}(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}_{\mathbf{v}}^\top(\mathbf{x})\boldsymbol{\phi}_{\mathbf{v}}(\mathbf{x}')$; and thus, the parametric approximant

$$\check{f}(\mathbf{x}) = \boldsymbol{\phi}_{\mathbf{v}}^\top(\mathbf{x})\boldsymbol{\theta}, \quad \boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\theta}; \mathbf{0}_{2n_{\text{RF}}}, \sigma_\theta^2\mathbf{I}_{2n_{\text{RF}}}) \quad (9)$$

can be viewed as coming from a realization of the Gaussian $\boldsymbol{\theta}$ combined with $\boldsymbol{\phi}_{\mathbf{v}}$ to yield the GP prior in (1) with $\kappa = \sigma_\theta^2\bar{\kappa}$, where σ_θ^2 is the magnitude of κ. Clearly, for any $\mathbf{X}_t$, the prior pdf of $\check{\mathbf{f}}_t$ is then

$$p(\check{\mathbf{f}}_t|\mathbf{X}_t) = \mathcal{N}(\check{\mathbf{f}}_t; \mathbf{0}_t, \check{\mathbf{K}}_t), \quad \check{\mathbf{K}}_t = \sigma_\theta^2\boldsymbol{\Phi}_t\boldsymbol{\Phi}_t^\top \quad (10)$$

where $\boldsymbol{\Phi}_t := [\boldsymbol{\phi}_{\mathbf{v}}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}_{\mathbf{v}}(\mathbf{x}_t)]^\top$, and $\check{\mathbf{K}}_t$ is then a low rank ($2n_{\text{RF}}$) approximant of $\mathbf{K}_t$ in (1) for $t > 2n_{\text{RF}}$.

With the parametric form of $\check{f}(\mathbf{x})$ in (9), the likelihood $p(\mathbf{y}_t|\check{\mathbf{f}}_t, \mathbf{X}_t)$ is also parametrized by $\boldsymbol{\theta}$. This together with the Gaussian prior of $\boldsymbol{\theta}$ (cf. (9)), yields the posterior $p(\boldsymbol{\theta}|\mathbf{y}_t, \mathbf{X}_t)$, based on which we can predict f and y at new test input $\mathbf{x}$. Specifically, upon replacing $p(f(\mathbf{x}_{t+1})|\mathbf{f}_t, \mathbf{X}_t)$ and $p(\mathbf{f}_t|\mathbf{y}_t, \mathbf{X}_t)$ in (2) by $p(\check{f}(\mathbf{x}_{t+1})|\boldsymbol{\theta}) = \delta(\check{f}(\mathbf{x}_{t+1}) - \boldsymbol{\phi}_{\mathbf{v}}^\top(\mathbf{x}_{t+1})\boldsymbol{\theta})$ and $p(\boldsymbol{\theta}|\mathbf{y}_t, \mathbf{X}_t)$, respectively, we obtain the predictive pdf of the RF-based $\check{f}(\mathbf{x}_{t+1})$, which further leads to the predictive pdf of y_{t+1} in (3) after replacing $f(\mathbf{x}_{t+1})$ by $\check{f}(\mathbf{x}_{t+1})$. For GPR, the predictive pdf of y_{t+1} is

$$p(y_{t+1}|\mathbf{y}_t, \mathbf{X}_{t+1}) = \mathcal{N}(y_{t+1}; \check{\hat{y}}_{t+1|\mathbf{t}}, \check{\sigma}^2_{t+1|\mathbf{t}}) \quad (11)$$

where

$$\check{\hat{y}}_{t+1|\mathbf{t}} = \boldsymbol{\phi}_{\mathbf{v}}^\top(\mathbf{x}_{t+1})\left(\boldsymbol{\Phi}_t^\top\boldsymbol{\Phi}_t + \frac{\sigma_n^2}{\sigma_\theta^2}\mathbf{I}_{2n_{\text{RF}}}\right)^{-1}\boldsymbol{\Phi}_t^\top\mathbf{y}_t \quad (12a)$$

$$\check{\sigma}^2_{t+1|\mathbf{t}} = \boldsymbol{\phi}_{\mathbf{v}}^\top(\mathbf{x}_{t+1})\left(\frac{\boldsymbol{\Phi}_t^\top\boldsymbol{\Phi}_t}{\sigma_n^2} + \frac{\mathbf{I}_{2n_{\text{RF}}}}{\sigma_\theta^2}\right)^{-1}\boldsymbol{\phi}_{\mathbf{v}}(\mathbf{x}_{t+1}) + \sigma_n^2. \quad (12b)$$

This *batch* predictor incurs complexity $\mathcal{O}(t(2n_{\text{RF}})^2 + (2n_{\text{RF}})^3)$, which is dominated by $\mathcal{O}(t(2n_{\text{RF}})^2)$ for $t \gg 2n_{\text{RF}}$. This linear (in t) complexity is apparently much more affordable than the plain-vanilla GP predictor (5).

The RF-based function approximant $\check{f}$ easily accommodates *online* operation [9], which is called for in many time-critical applications, including time series prediction [45], and robot localization [46]. While the RF-based online approach for GPR is offered in [9], its performance hinges on a *preselected* kernel for the GP prior, which may fall short in characterizing upcoming data samples. Next, we will broaden the scope of a single GP prior by an ensemble (E) of GPs to enable real-time kernel adaptation. Besides serving the role of a non-Gaussian prior, EGP will turn out to be scalable too, after adopting once again the RF approximation.

1. Quantities with ˇ involve RF approximations.

0162-8828 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Minnesota. Downloaded on July 26,2022 at 19:20:57 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3157197, IEEE Transactions on Pattern Analysis and Machine Intelligence

3 Online scalable ensemble GPs

Towards data-driven kernel selection in the online setting, an EGP assembler employs an ensemble of M GP experts (a.k.a. models or learners), each of which places a unique GP prior on f as $f|m \sim \mathcal{GP}(0, \kappa^m(\mathbf{x}, \mathbf{x}'))$, where $m \in \mathcal{M} := \{1, \ldots, M\}$ is the expert index and κ^m is a shift-invariant kernel selected from a *known* kernel dictionary $\mathcal{K} := \{\kappa^1, \ldots, \kappa^M\}$. Here, $\mathcal{K}$ should be constructed as large as computational constraints allow, depending on resources and the learning task. Per expert m, the prior pdf of function values at $\mathbf{X}_t$ is

$$p(\mathbf{f}_t|i=m, \mathbf{X}_t) = \mathcal{N}(\mathbf{f}_t; \mathbf{0}_t, \mathbf{K}_t^m),\ [\mathbf{K}_t^m]_{\tau,\tau'} := \kappa^m(\mathbf{x}_\tau, \mathbf{x}_{\tau'})$$

where the hidden random variable i is introduced to denote the expert index. The ensemble prior pdf of $\mathbf{f}_t$ that accounts for all GP experts is given by the Gaussian mixture (GM)

$$p(\mathbf{f}_t|\mathbf{X}_t) = \sum_{m=1}^{M} w^m \mathcal{N}(\mathbf{f}_t; \mathbf{0}_t, \mathbf{K}_t^m)\ , \qquad \sum_{m=1}^{M} w^m = 1 \quad (13)$$

where the *unknown* weights $\{w^m\}_{m=1}^M$, viewed as probabilities of the GP experts to be present in the EGPs, are to be learned from data that arrive sequentially.

Seeking a scalable predictor, each expert m relies on the RF-based function approximant (9) with the per-expert parameter vector $\boldsymbol{\theta}^m$ and RF vector $\boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x})$ constructed as in (8) using $\{\mathbf{v}_j^m\}_{j=1}^{n_{\text{RF}}}$. Vectors $\{\mathbf{v}_j^m\}_{j=1}^{n_{\text{RF}}}$ here are drawn i.i.d. from $\pi_{\bar{\kappa}}^m(\mathbf{v})$, which is the PSD of the standardized kernel $\bar{\kappa}^m$, relating to κ^m through the magnitude $\sigma_{\theta^m}^2$ as $\kappa^m = \sigma_{\theta^m}^2 \bar{\kappa}^m$. The per expert m generative model for output y is then

$$p(\boldsymbol{\theta}^m) = \mathcal{N}(\boldsymbol{\theta}^m; \mathbf{0}_{2n_{\text{RF}}}, \sigma_{\theta^m}^2 \mathbf{I}_{2n_{\text{RF}}}) \quad (14a)$$

$$p(y|\boldsymbol{\theta}^m, \mathbf{x}) = p(y|\boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x})\boldsymbol{\theta}^m)\ . \quad (14b)$$

Focusing on the incremental (I) setting, each expert m interleaves prediction of y_{t+1} based on $p(\boldsymbol{\theta}^m|\mathbf{y}_t, \mathbf{X}_t)$, and update of the parameter posterior upon the arrival of y_{t+1} per slot. To assess the per-expert contribution, the EGP assembler relies on the posterior probability $w_t^m := \Pr(i = m|\mathbf{y}_t; \mathbf{X}_t)$. As we shall see next, the resultant IE-GP proceeds in two steps, namely prediction and correction, by propagating per-expert weights and posterior pdfs $\{w_t^m, p(\boldsymbol{\theta}^m|\mathbf{y}_t, \mathbf{X}_t)\}_{m=1}^M$ from slot to slot.

Prediction. Upon receiving $\mathbf{x}_{t+1}$, each expert m constructs the RF vector using $\boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_{t+1})$ as in (8). With $p(\boldsymbol{\theta}^m|\mathbf{y}_t, \mathbf{X}_t)$ available from slot t, the per-expert predictive pdf of y_{t+1} can be obtained by invoking the sum-product probability rule

$$p(y_{t+1}|\mathbf{y}_t, i=m, \mathbf{X}_{t+1}) = \int p(y_{t+1}|\boldsymbol{\theta}^m, \mathbf{x}_{t+1}) p(\boldsymbol{\theta}^m|\mathbf{y}_t, \mathbf{X}_t) d\boldsymbol{\theta}^m. \quad (15)$$

Leveraging again the sum-product rule, the EGP assembler seeks the ensemble predictive pdf as

$$\begin{aligned} p(y_{t+1}|\mathbf{y}_t, \mathbf{X}_{t+1}) &= \sum_{m=1}^{M} \Pr(i=m|\mathbf{y}_t, \mathbf{X}_t) p(y_{t+1}|\mathbf{y}_t, i=m, \mathbf{X}_{t+1}) \\ &= \sum_{m=1}^{M} w_t^m p(y_{t+1}|\mathbf{y}_t, i=m, \mathbf{X}_{t+1}) \end{aligned} \quad (16)$$

which takes an intuitive form as a weighted combination of predictions from the individual GP experts. Having available the predictive pdf, we are ready to update the posterior pdf of the RF model parameter vector.

Algorithm 1 IE-GP for GPR

Input: κ^m, $m = 1, \ldots, M$, and number of RFs n_{RF}.
Initialization:
for $m = 1, 2, \ldots, M$ **do**
  Draw n_{RF} random vectors $\{\mathbf{v}_i^m\}_{i=1}^{n_{\text{RF}}}$;
  $w_0^m = 1/M$; $\hat{\boldsymbol{\theta}}_0^m = \mathbf{0}_{2D}$; $\boldsymbol{\Sigma}_0^m = \sigma_{\theta^m}^2 \mathbf{I}_{2D}$;
end for

for $t = 1, 2, \ldots, T$ **do**
  Receive input datum $\mathbf{x}_t$;
  for $m = 1, 2, \ldots, M$ **do**
    Construct RF $\boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_t)$ via (8);
    Obtain per-expert pdf of y_t via (22);
    Update w_t^m via (25);
    Update per-expert pdf of $\boldsymbol{\theta}^m$ via (26);
  end for
end for

Correction. With the arrival of y_{t+1}, each expert m updates the posterior pdf of $\boldsymbol{\theta}^m$ via Bayes' rule as

$$p(\boldsymbol{\theta}^m|\mathbf{y}_{t+1}, \mathbf{X}_{t+1}) = \frac{p(\boldsymbol{\theta}^m|\mathbf{y}_t, \mathbf{X}_t) p(y_{t+1}|\boldsymbol{\theta}^m, \mathbf{x}_{t+1})}{p(y_{t+1}|\mathbf{y}_t, i=m, \mathbf{X}_{t+1})} \quad (17)$$

where $p(y_{t+1}|\boldsymbol{\theta}^m, \mathbf{x}_{t+1})$ is the known likelihood (cf. (14b)), and $p(\boldsymbol{\theta}^m|\mathbf{y}_t, \mathbf{X}_t)$ is available from slot t. For later use, the so-termed Bayesian loss incurred by expert m at slot $t+1$ is (cf. [47])

$$l_{t+1|t}^m := -\log p(y_{t+1}|\mathbf{y}_t, i=m, \mathbf{X}_{t+1}) \quad (18)$$

whose ensemble version is given by

$$\begin{aligned} \ell_{t+1|t} &:= -\log p(y_{t+1}|\mathbf{y}_t, \mathbf{X}_{t+1}) \\ &= -\log \sum_{m=1}^{M} w_t^m \exp\left(-l_{t+1|t}^m\right). \end{aligned} \quad (19)$$

Simultaneously, the EGP meta-leaner obtains the updated weight $w_{t+1}^m := \Pr(i = m|\mathbf{y}_{t+1}, \mathbf{X}_{t+1})$ as

$$\begin{aligned} w_{t+1}^m &= \frac{\Pr(i=m|\mathbf{y}_t, \mathbf{X}_t) p(y_{t+1}|\mathbf{y}_t, i=m, \mathbf{X}_{t+1})}{p(y_{t+1}|\mathbf{y}_t, \mathbf{X}_{t+1})} \\ &= w_t^m \exp(\ell_{t+1|t} - l_{t+1|t}^m) \end{aligned} \quad (20)$$

where w_t^m is available from slot t. Intuitively, large $l_{t+1|t}^m$ implies small $\ell_{t+1|t} - l_{t+1|t}^m$, and thus w_{t+1}^m relative to the rest will be smaller than that at slot t.

Summarizing, our scalable IE-GP algorithm for general likelihoods (and thus posteriors) relies on (15)-(17) to transition from slot t to slot $t+1$. Next, we specialize our novel IE-GP to GPR that enjoys closed-form pdf and weight updates, as well as non-Gaussian likelihoods that entail Laplace approximation [48] to evaluate the (possibly high-dimensional) integrals in the prediction and correction steps.

3.1 Closed-form updates for GPR

For GPR, the likelihood per expert is given by $p(y_t|\boldsymbol{\theta}^m, \mathbf{x}_t) = \mathcal{N}(y_t; \boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}_t)\boldsymbol{\theta}^m, \sigma_n^2)$, which together with the per-expert Gaussian prior $p(\boldsymbol{\theta}^m)$ (cf. (14a)), yields the Gaussian posterior at the end of slot t expressed as

$$p(\boldsymbol{\theta}^m|\mathbf{y}_t, \mathbf{X}_t) = \mathcal{N}(\boldsymbol{\theta}^m; \hat{\boldsymbol{\theta}}_t^m, \boldsymbol{\Sigma}_t^m) \quad (21)$$

0162-8828 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Minnesota. Downloaded on July 26,2022 at 19:20:57 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3157197, IEEE Transactions on Pattern Analysis and Machine Intelligence

with mean $\hat{\boldsymbol{\theta}}_t^m$ and covariance matrix $\boldsymbol{\Sigma}_t^m$ per expert m.

Building on (21) and (15), the predictive pdf of y_{t+1} from expert m is also Gaussian

$$p(y_{t+1}|\mathbf{y}_t, i=m, \mathbf{X}_{t+1}) = \mathcal{N}\left(y_{t+1}; \hat{y}_{t+1|t}^m, (\sigma_{t+1|t}^m)^2\right) \quad (22)$$

where the predicted mean and variance are

$$\hat{y}_{t+1|t}^m = \boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}_{t+1})\hat{\boldsymbol{\theta}}_t^m \quad (23a)$$

$$(\sigma_{t+1|t}^m)^2 = \boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}_{t+1})\boldsymbol{\Sigma}_t^m\boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_{t+1}) + \sigma_n^2 \,. \quad (23b)$$

Thus, the ensemble predictive pdf of y_{t+1} in (16) specialized to GPR is a GM, based on which the EGP assembler obtains the minimum mean-square error (MMSE) predictor of y_{t+1} together with the associated variance as

$$\hat{y}_{t+1|t} = \sum_{m=1}^M w_t^m \hat{y}_{t+1|t}^m \quad (24a)$$

$$\sigma_{t+1|t}^2 = \sum_{m=1}^M w_t^m[(\sigma_{t+1|t}^m)^2 + (\hat{y}_{t+1|t} - \hat{y}_{t+1|t}^m)^2]. \quad (24b)$$

When y_{t+1} becomes available, the EGP assembler updates the per-expert weight as (cf. (20) and (22))

$$w_{t+1}^m = \frac{w_t^m \mathcal{N}\left(y_{t+1}; \hat{y}_{t+1|t}^m, (\sigma_{t+1|t}^m)^2\right)}{\sum_{m'=1}^M w_t^{m'} \mathcal{N}\left(y_{t+1}; \hat{y}_{t+1|t}^{m'}, (\sigma_{t+1|t}^{m'})^2\right)} \,. \quad (25)$$

With the per-expert Gaussian likelihood, the arrival of y_{t+1} also propagates Gaussianity of the posterior pdf of $\boldsymbol{\theta}^m$ from slot t to $t+1$, expressed as

$$p(\boldsymbol{\theta}^m|\mathbf{y}_{t+1}, \mathbf{X}_{t+1}) = \mathcal{N}(\boldsymbol{\theta}^m; \hat{\boldsymbol{\theta}}_{t+1}^m, \boldsymbol{\Sigma}_{t+1}^m) \quad (26)$$

where the per-expert mean $\hat{\boldsymbol{\theta}}_{t+1}^m$ and covariance matrix $\boldsymbol{\Sigma}_{t+1}^m$ are

$$\hat{\boldsymbol{\theta}}_{t+1}^m = \hat{\boldsymbol{\theta}}_t^m + (\sigma_{t+1|t}^m)^{-2}\boldsymbol{\Sigma}_t^m\boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_{t+1})(y_{t+1} - \hat{y}_{t+1|t}^m) \quad (27a)$$

$$\boldsymbol{\Sigma}_{t+1}^m = \boldsymbol{\Sigma}_t^m - (\sigma_{t+1|t}^m)^{-2}\boldsymbol{\Sigma}_t^m\boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_{t+1})\boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}_{t+1})\boldsymbol{\Sigma}_t^m. \quad (27b)$$

Accounting for all M expert updates, our scalable IE-GP approach to GPR (see Algorithm 1) has per-iteration complexity of $\mathcal{O}(M(2n_{\text{RF}})^2)$; hence, scalability is not compromised by the ensemble approach that also offers a richer model for the learning function. The deterministic RKHS online approach (termed "Raker" in [34]) relies on first-order gradient descent to update $\boldsymbol{\theta}$ at per-iteration complexity of $\mathcal{O}(Mn_{\text{RF}}d)$, which is lower than our second-order update in (27). Our probabilistic IE-GP approach offers numerically improved performance that is also analytically quantifiable through the predictor variance (24b) in (24a).

3.2 Laplace approximation for non-Gaussian likelihood

When the likelihood is non-Gaussian as in classification, Poisson regression or ordinal regression, the per-expert parameter posterior is no longer available in closed form. Fortunately, the so-termed Laplace approximation [48] can be leveraged to carry out the prediction and correction steps with tractability in IE-GP. Specifically, each expert maintains a Gaussian approximant for the parameter vector as $p(\boldsymbol{\theta}^m|\mathbf{y}_t, \mathbf{X}_t) \approx \mathcal{N}(\boldsymbol{\theta}^m; \hat{\boldsymbol{\theta}}_t^m, \boldsymbol{\Sigma}_t^m)$, which readily allows the predicted pdf of y_{t+1} (15) to be calculated via Monte-Carlo sampling, or through probit approximation for binary classification with the logistic likelihood; see, e.g., [49, Chapter 8.4.4.2].

Given y_{t+1}, the EGP assembler updates the per-expert weight as in (20) with the per-expert loss at hand from the prediction step. Also, each expert relies on Laplace approximation to seek an updated Gaussian approximant as $p(\boldsymbol{\theta}^m|\mathbf{y}_{t+1}, \mathbf{X}_{t+1}) \approx \mathcal{N}(\boldsymbol{\theta}^m; \hat{\boldsymbol{\theta}}_{t+1}^m, \boldsymbol{\Sigma}_{t+1}^m)$, where $\hat{\boldsymbol{\theta}}_{t+1}^m$ and $\boldsymbol{\Sigma}_{t+1}^m$ are respectively the mode of the log posterior and the corresponding information matrix inverse, that can be obtained by solving the following optimization problem using Newton's iteration [49]

$$\hat{\boldsymbol{\theta}}_{t+1}^m = \arg\max_{\boldsymbol{\theta}^m} \log p(\boldsymbol{\theta}^m|\mathbf{y}_t, \mathbf{X}_t) + \log p(y_{t+1}|\boldsymbol{\theta}^m, \mathbf{x}_{t+1})$$

$$(\boldsymbol{\Sigma}_{t+1}^m)^{-1} = (\boldsymbol{\Sigma}_t^m)^{-1} - \nabla_{\theta^m}^2 \log p(y_{t+1}|\boldsymbol{\theta}^m, \mathbf{x}_{t+1})\Big|_{\boldsymbol{\theta}^m = \hat{\boldsymbol{\theta}}_{t+1}^m} \,.$$

Remark 1. Relying on a dictionary of kernels, the proposed IE-GP framework is related to the kernel selection works in the GP literature; see, e.g., [50], [51], [52]. Nevertheless, the latter not only requires batch data, but also falls short in scalability. Leveraging RF-based approximation, the online scalable IE-GP also welcomes regret analysis in the ensuing section.

4 Regret analysis

The pdf $p(y_{t+1}|\mathbf{y}_t, \mathbf{X}_{t+1})$ in (16) provide an online performance metric for $\hat{y}_{t+1|t}$, from which its mean and variance can be also obtained (even in closed form, cf. (24b)). These metrics however, rely on the assumption of knowing the prior pdf of f, and the conditional data likelihood. To guard against having *imperfect knowledge* of these pdfs (the norm in adversarial settings), regret analysis is well motivated along the lines of online convex optimization [10] and online learning with expert advice [11]. This is the subject of this section that aims to benchmark performance of our IE-GP predictor relative to the best function estimator with data in hindsight when the generative assumptions are violated.

To this end, let $\mathcal{L}(f(\mathbf{x}_\tau); y_\tau) := -\log p(y_\tau|f(\mathbf{x}_\tau))$ be the per-slot negative log-likelihood (NLL). For any fixed function estimator $\hat{f}^*(\cdot)$, the incurred loss over T slots is $\sum_{\tau=1}^T \mathcal{L}(\hat{f}^*(\mathbf{x}_\tau); y_\tau)$. With the EGP prior in (13), the best function estimate (benchmark) with data $\{\mathbf{X}_T, \mathbf{y}_T\}$ available in hindsight, are obtained with the optimal weights $\{w^m\}$ in the EGP prior by maximizing the batch function posterior, $p(\mathbf{f}_T|\mathbf{y}_T, \mathbf{X}_T) \propto p(\mathbf{f}_T|\mathbf{X}_T)p(\mathbf{y}_T|\mathbf{f}_T, \mathbf{X}_T)$, as

$$(\hat{\mathbf{f}}_T, \{\hat{w}^m\}) = \arg\max_{\substack{\mathbf{f}_T, \{w^m\} \\ \sum_m w^m = 1}} p(\mathbf{y}_T|\mathbf{f}_T, \mathbf{X}_T) \sum_{m=1}^M w^m p(\mathbf{f}_T|i=m, \mathbf{X}_T)$$

whose solution is $\hat{w}^{m^*} = 1$ and $\hat{w}^m = 0$ for $m \neq m^*$. This implies that only one GP expert m^* is active in the benchmark function estimate for $\tau = 1, \ldots, T$. The optimal estimate by expert m^* are then given by

$$\hat{\mathbf{f}}_T = \arg\max_{\mathbf{f}_T} \; p(\mathbf{f}_T|i=m^*, \mathbf{X}_T)p(\mathbf{y}_T|\mathbf{f}_T, \mathbf{X}_T) \,. \quad (29)$$

As every positive semidefinite kernel κ^m is associated with a unique RHKS $\mathcal{H}^m$ [14], the optimal function estimator $\hat{f}^{m^*}(\cdot)$ is extracted from (29) as

$$m^* \in \arg\min_{m\in\mathcal{M}} \sum_{\tau=1}^T \mathcal{L}(\hat{f}^m(\mathbf{x}_\tau); y_\tau) + \frac{1}{2}\|\hat{f}^m\|_{\mathcal{H}^m}^2 \quad (30)$$

where the optimal function estimator per expert $\hat{f}^m(\cdot)$, $m = 1, \ldots, M$, is obtained as

$$\hat{f}^m(\cdot) \in \arg\min_{f^m\in\mathcal{H}^m} \sum_{\tau=1}^T \mathcal{L}(f^m(\mathbf{x}_\tau); y_\tau) + \frac{1}{2}\|f^m\|_{\mathcal{H}^m}^2 \,.$$

0162-8828 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Minnesota. Downloaded on July 26,2022 at 19:20:57 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3157197, IEEE Transactions on Pattern Analysis and Machine Intelligence

With the best fixed function estimator $\hat{f}^{m^*}(\cdot)$ at hand, the static regret over T slots is then defined as [47]

$$\mathcal{R}^{\mathrm{ST}}(T) := \sum_{\tau=1}^{T} \ell_{\tau|\tau-1} - \sum_{\tau=1}^{T} \mathcal{L}(\hat{f}^{m^*}(\mathbf{x}_\tau); y_\tau) \tag{31}$$

where $\ell_{\tau|\tau-1}$, defined in (19), captures the ensemble online Bayesian loss incurred by IE-GP.

Although the cumulative online loss in the first sum of (31) has different form than that of the benchmark, they are comparable by the data likelihood, where the function is nonrandom. In other words, the online Bayesian loss is obtained by taking the expectation of the likelihood wrt the online predictive pdf of the function, thus eliminating the randomness of the function in the likelihood.

To proceed, we will need the following assumptions.

- **(as1)** The NLL $\mathcal{L}(z_\tau; y_\tau)$ is continuously twice differentiable with $|\frac{d^2}{dz_\tau^2}\mathcal{L}(z_\tau; y_\tau)| \leq c, \forall z_\tau$;
- **(as2)** The NLL $\mathcal{L}(z_t; y_t)$ is convex and has bounded derivative wrt z_t; that is, $|\frac{d}{dz_t}\mathcal{L}(z_t; y_t)| \leq L$;
- **(as3)** Kernels $\{\bar{\kappa}^m\}_{s=1}^M$ are shift-invariant, standardized and bounded, that is $\bar{\kappa}^m(\mathbf{x}_t, \mathbf{x}_{t'}) \leq 1, \forall \mathbf{x}_t, \mathbf{x}_{t'}$.

Differentiability and convexity of the NLL in (as1)-(as2) are satisfied by most forms of likelihood in GP-based learning, including the Gaussian likelihood in GPR, and the logistic one for classification. Conditions in (as3) hold for a wide class of kernels including Gaussian, Laplace and Cauchy ones [8]. As the derivations rely on the general form of IE-GP (cf. (15)-(17)) that corresponds to general likelihoods, the regret bound established here applies to general learning tasks.

To establish the static regret bound of IE-GP, we will need the following intermediate lemma.

Lemma 1. *Under (as1), and with prior of $\boldsymbol{\theta}^m$ given by* (14a)*, the following bound holds concerning the cumulative online Bayesian loss incurred by the IE-GP and the counterpart from a single RF-based GP expert with fixed $\boldsymbol{\theta}_*^m$*

$$\sum_{\tau=1}^{T} \ell_{\tau|\tau-1} - \sum_{\tau=1}^{T} \mathcal{L}(\boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}_\tau)\boldsymbol{\theta}_*^m; y_\tau) \leq \frac{\|\boldsymbol{\theta}_*^m\|^2}{2\sigma_{\theta^m}^2} + n_{RF} \log\left(1 + \frac{T\sigma_{\theta^m}^2}{2n_{RF}}\right) + \log M \ . \tag{32}$$

Proof: See Sec. 9.1.

Lemma 1 bounds the cumulative online Bayesian loss of IE-GP relative to any single RF-based GP learner with a fixed strategy. Next, we will work towards the ultimate static regret by further bounding the loss of RF-based function estimator relative to the best function estimator in the original RKHS for each expert.

Theorem 1. *Under as(1)-as(3) and with $\hat{f}^{m^*}$ belonging to the RHKS $\mathcal{H}^{m^*}$ induced by κ^{m^*}, for a fixed $\epsilon > 0$, the following bound holds with probability at least $1 - 2^8(\frac{\sigma_{m^*}}{\epsilon})^2 \exp\left(\frac{-n_{RF}\epsilon^2}{4d+8}\right)$*

$$\sum_{\tau=1}^{T} \ell_{\tau|\tau-1} - \sum_{\tau=1}^{T} \mathcal{L}(\hat{f}^{m^*}(\mathbf{x}_\tau); y_\tau) \leq \frac{(1+\epsilon)C^2}{2\sigma_{\theta^{m^*}}^2} + n_{RF} \log\left(1 + \frac{Tc\sigma_{\theta^{m^*}}^2}{2n_{RF}}\right) + \log M + \epsilon LTC \tag{33}$$

where C is a constant, and $\sigma_{m^*}^2 := \mathbb{E}_{\pi_{\tilde{\kappa}}^{m^*}}[\|\mathbf{v}^{m^*}\|^2]$ is the second-order moment of $\mathbf{v}^{m^*}$. Setting $\epsilon = \mathcal{O}(\log T/T)$, the static regret in (31) *boils down to*

$$\mathcal{R}^{\mathrm{ST}}(T) = \mathcal{O}(\log T) \ . \tag{34}$$

Proof. See Sec. 9.2.

Theorem 1 asserts that IE-GP incurs no regret on average with cumulative static regret $\mathcal{O}(\log T)$ over T slots, thereby demonstrating its robustness to the adversarial setting. It is also worth highlighting that this regret bound is tighter than that of the deterministic RKHS-based online multi-kernel counterpart [34] with regret $\mathcal{O}(\sqrt{T})$ in the static setting.

5 EGP FOR DYNAMIC LEARNING

In the proposed IE-GP, the EGP assembler relies on the posterior probability of a *static* random variable i to assess contributions of the GP experts, each of which models the learning function via the *time-invariant* parameter vector $\boldsymbol{\theta}^m$. Such a stationary setting implies that IE-GP handles no dynamics in the unknown function. This is also manifested in Sec. 4 that with batch data in hindsight the optimal function estimate is associated with one of the GP experts (cf. (29)). To further enable learning for dynamic functions, the rest of this section will explore extensions to accommodate *time-varying* i_t and $\boldsymbol{\theta}_t^m$ for the EGP assembler and individual GP learners, respectively.

5.1 Dynamics at EGP assembler

Capitalizing on time-dependent $i_t \in \mathcal{M}$ to denote the index of the contributing expert, the EGP assembler models the evolution of i_t via a Markov chain with prior transition probability

$$q_{mm'} := \Pr(i_{t+1} = m | i_t = m') \tag{35}$$

for $m, m' \in \mathcal{M}$. Such a dynamic model allows the learning function to jump among the candidate spaces associated with the GP experts, yielding the so-termed switching (S) IE-GP hereafter. The values of $\{q_{mm'}\}_{m,m'} \in [0,1]$ are user-defined parameters. It is worth pointing out that IE-GP can be regarded as a special case of SIE-GP with $q_{mm} = 1$.

The novel SIE-GP differs from IE-GP in the weight update. To illustrate this, the per-expert posterior weight in SIE-GP is first adapted as $w_{t|t}^m := \Pr(i_t = m | \mathbf{y}_t, \mathbf{X}_t)$ given time-varying i_t. Before propagating to $w_{t+1|t+1}^m$, the EGP assembler leverages the aforementioned Markov transition model to predict the weight for GP model m at slot $t+1$ via $w_{t+1|t}^m := \Pr(i_{t+1} = m | \mathbf{y}_t, \mathbf{X}_t)$, which, with $\{w_{t|t}^m\}$ available, can be obtained as

$$\begin{aligned} w_{t+1|t}^m &= \sum_{m'=1}^{M} \Pr(i_{t+1} = m, i_t = m' | \mathbf{y}_t, \mathbf{X}_t) \\ &= \sum_{m'=1}^{M} \Pr(i_{t+1} = m | i_t = m', \mathbf{y}_t, \mathbf{X}_t) \Pr(i_t = m' | \mathbf{y}_t, \mathbf{X}_t) \\ &= \sum_{m'=1}^{M} q_{m,m'} w_{t|t}^{m'} \end{aligned} \tag{36}$$

where the last equality holds since the evolution of i_t is independent from $\mathbf{y}_t$ and $\mathbf{X}_t$. With non-zero $\{q_{mm'}\}$, the prediction rule in (36) allows for activation of the previously inactive expert m

0162-8828 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Minnesota. Downloaded on July 26,2022 at 19:20:57 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3157197, IEEE Transactions on Pattern Analysis and Machine Intelligence

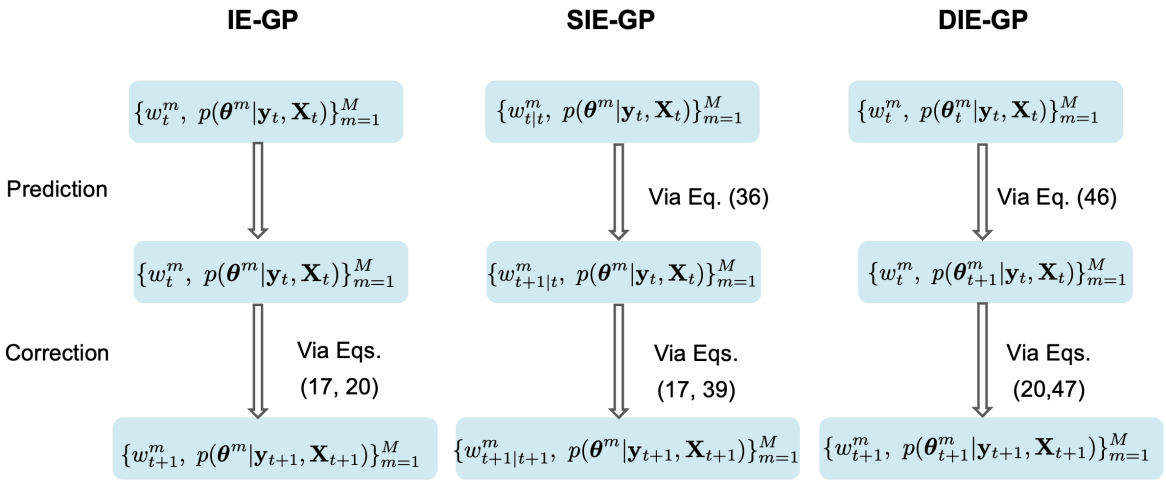


Fig. 1: Schematic diagrams for IE-GP, SIE-GP, and DIE-GP. Notice the difference in the subscripts of w and $\boldsymbol{\theta}$, and the implementation of the prediction and correction steps.

($w^m_{t|t} \approx 0$), thereby accommodating switching among candidate GP models or function spaces.

With the predicted weights (36) and the per-expert predictive pdf (15) available, the EGP assembler leverages the sum-product probability rule to predict the pdf of y_{t+1}

$$\begin{aligned} p(y_{t+1}|\mathbf{y}_t, \mathbf{X}_{t+1}) &= \sum_{m=1}^{M} p(y_{t+1}, i_{t+1}=m|\mathbf{y}_t, \mathbf{X}_{t+1}) \\ &= \sum_{m=1}^{M} \Pr(i_{t+1}=m|\mathbf{y}_t, \mathbf{X}_t) p(y_{t+1}|\mathbf{y}_t, i_{t+1}=m, \mathbf{X}_{t+1}) \\ &= \sum_{m=1}^{M} w^m_{t+1|t}\, p(y_{t+1}|\mathbf{y}_t, i_{t+1}=m, \mathbf{X}_{t+1}) \end{aligned} \tag{37}$$

where $w^m_{t+1|t}$ replaces w^m_t in (16) for IE-GP. To facilitate the upcoming regret analysis, the aggregated online loss for SIE-GP that accounts for the per-expert loss (18) is defined as

$$\ell^{\mathrm{SW}}_{t+1|t} := -\log \sum_{m=1}^{M} w^m_{t+1|t} \exp\left(-l^m_{t+1|t}\right) \tag{38}$$

where the superscript "SW", denoting the switching scenario, is used to distinguish from the loss from IE-GP in (19).

Upon acquiring y_{t+1}, the EGP meta-leaner then updates the per-expert weight as

$$\begin{aligned} w^m_{t+1|t+1} &:= \frac{w^m_{t+1|t} p(y_{t+1}|\mathbf{y}_t, i_{t+1}=m, \mathbf{X}_{t+1})}{\sum_{m'=1}^{M} w^{m'}_{t+1|t} p(y_{t+1}|\mathbf{y}_t, i_{t+1}=m', \mathbf{X}_{t+1})} \\ &= w^m_{t+1|t} \exp(\ell^{\mathrm{SW}}_{t+1|t} - l^m_{t+1|t})\ . \end{aligned} \tag{39}$$

To sum up, relative to IE-GP, the meta-leaner in SIE-GP performs an additional weight prediction step (36), which contributes to the ensemble predictive pdf (37) and the weight update (39). Next, the regret analysis of SIE-GP will be conducted in line with Sec. 4 to account for the adversarial setting.

5.1.1 Switching regret analysis

With the underlying assumption that the active model per slot changes over time, the notion of switching or shifting regret [11, Chapter 5.2] is leveraged to analyse the performance of SIE-GP in the adversarial setting where the generative assumptions are violated. Specifically, SIE-GP is compared with an arbitrary sequence of benchmark functions $\{\hat{f}^{i_\tau} \in \mathcal{H}^{i_\tau}\}_{\tau=1}^T$ with data in hindsight, yielding the switching regret defined as

$$\mathcal{R}^{\mathrm{SW}}(T) := \sum_{\tau=1}^{T} \ell^{\mathrm{SW}}_{\tau|\tau-1} - \min_{i_1,\ldots,i_T} \sum_{\tau=1}^{T} \mathcal{L}(\hat{f}^{i_\tau}(\mathbf{x}_\tau); y_\tau) \tag{40}$$

where, as in static regret (31), the loss incurred by the comparator in the switching case is measured via the NLL.

Aiming at establishing an upper bound for (40), the following two additional assumptions will be entailed.

(as4) $q_{mm} = q_0$, $q_{mm'} = \frac{q_1}{M-1}$ for $m, m' \in \mathcal{M}$, $q_0 + q_1 = 1$, and $0 \leq q_1 < \frac{1}{2} < q_0 \leq 1$;

(as5) The number of switches for sequence $\{i_1, \ldots, i_T\}$ is upper bounded by S, i.e., $\sum_{\tau=1}^{T} I(i_\tau \neq i_{\tau+1}) \leq S$, and $S \ll T$.

In (as4), q_1 is the probability that the current GP model switches to a different model. With $0 \leq q_1 < \frac{1}{2} < q_0 \leq 1$, the learning function has larger probability of staying in the current space rather than switching to another space. This, together with (as5), bounds the variations of the benchmark function over slots. The transition probabilities dictated by (as4) yield a weight evolution strategy that is similar to the one given by the fixed-share forecaster in online expert-based learning [11, Chapter 5.2], where the conditions in (as5) are also leveraged in the regret analysis. Before obtaining the upper bound for (40) in Theorem 2, the following two intermediate lemmas will be established first.

Lemma 2. *Under as(4)-as(5), the following bound holds true concerning the cumulative ensemble switching loss and the single expert-based online counterpart for any sequence* $\{i_1, \ldots, i_T\}$

$$\sum_{\tau=1}^{T} \ell^{\mathrm{SW}}_{\tau|\tau-1} - \sum_{\tau=1}^{T} l^{i_\tau}_{\tau|\tau-1} \leq \log M + S\log T - S\log S + S\ . \tag{41}$$

Proof: See Sec. 9.3.

Lemma 3. *Under (as1) and for any sequence* $\{i_1, \ldots, i_T\}$*, the following bound holds regarding the difference of the cumulative*

0162-8828 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Minnesota. Downloaded on July 26,2022 at 19:20:57 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3157197, IEEE Transactions on Pattern Analysis and Machine Intelligence

single expert-based online loss and the counterpart incurred by RF-based benchmark functions with fixed parameters $\{\boldsymbol{\theta}_*^m\}_{m=1}^M$

$$\sum_{\tau=1}^{T} l_{\tau|\tau-1}^{i_\tau} - \sum_{\tau=1}^{T} \mathcal{L}(\boldsymbol{\phi}_{\mathbf{v}}^{i_\tau\top}(\mathbf{x}_\tau)\boldsymbol{\theta}_*^{i_\tau}; y_\tau) \leq \sum_{m=1}^{M} \frac{\|\boldsymbol{\theta}_*^m\|^2}{2\sigma_{\theta^m}^2} + n_{RF} M \log\left(1 + \frac{Tc\sigma_{\theta^*}^2}{2n_{RF}M}\right) \quad (42)$$

where $\sigma_{\theta^*}^2 := \max_{m\in\mathcal{M}} \sigma_{\theta^m}^2$.

Proof: See Sec. 9.4.

Theorem 2. *Under as(1)-as(5) and with* $\hat{f}^m$ *belonging to the RHKS* $\mathcal{H}^m$ *induced by* κ^m*, for a fixed* $\epsilon > 0$*, the following bound holds with probability at least* $1 - 2^8(\frac{\sigma_*}{\epsilon})^2 \exp\left(\frac{-n_{RF}\epsilon^2}{4d+8}\right)$

$$\sum_{\tau=1}^{T} \ell_{\tau|\tau-1}^{\mathrm{SW}} - \sum_{\tau=1}^{T} \mathcal{L}(\hat{f}^{i_\tau}(\mathbf{x}_\tau); y_\tau) \leq \log M + S\log T - S\log S + S + \epsilon LTC' + \sum_{m=1}^{M} \frac{(1+\epsilon)C'^2}{2\sigma_{\theta^{m*}}^2} + n_{RF} M \log\left(1 + \frac{Tc\sigma_{\theta^*}^2}{2n_{RF}M}\right) \quad (43)$$

where C' *is some constant, and* $\sigma_*^2 := \max_{m\in\mathcal{M}} \sigma_m^2 = \max_{m\in\mathcal{M}} \mathbb{E}_{\pi_{\bar{\kappa}^m}^*}[\|\mathbf{v}^{m^*}\|^2]$. *Setting* $\epsilon = \mathcal{O}(\log T/T)$*, the switching regret in* (40) *boils down to*

$$\mathcal{R}^{\mathrm{SW}}(T) = \mathcal{O}(\log T)\ . \quad (44)$$

Proof: See Sec. 9.5.

Even in the presence of model switching, the advocated SIE-GP suffers from diminishing average regret by explicitly accounting for such switching dynamics.

5.2 Dynamics within each GP expert

The aforementioned SIE-GP accounts for dynamics of expert switching at the EGP meta-leaner. To further handle a dynamic learning function within each expert m, a time-varying parameter vector $\boldsymbol{\theta}_t^m$ will be considered instead of time-invariant $\boldsymbol{\theta}^m$ in IE-GP, yielding the dynamic (D) IE-GP approach. Specifically, DIE-GP captures dynamics in $\boldsymbol{\theta}_t^m$ via the random walk model

$$\boldsymbol{\theta}_{t+1}^m = \boldsymbol{\theta}_t^m + \boldsymbol{\epsilon}_{t+1}^m \quad (45)$$

where the noise $\boldsymbol{\epsilon}_{t+1}^m$ is white and Gaussian distributed with mean zero and covariance matrix $\sigma_{\epsilon^m}^2 \mathbf{I}_{2D}$.

Rather than updating $p(\boldsymbol{\theta}^m|\mathbf{y}_t, \mathbf{X}_t)$ as in IE-GP, expert m in DIE-GP propagates $p(\boldsymbol{\theta}_t^m|\mathbf{y}_t, \mathbf{X}_t)$ across slots. Taking into account (45), expert m first predicts the pdf of $\boldsymbol{\theta}_{t+1}^m$ at the beginning of slot $t+1$ as

$$p(\boldsymbol{\theta}_{t+1}^m|\mathbf{y}_t, \mathbf{X}_{t+1}) = \int p(\boldsymbol{\theta}_{t+1}^m|\boldsymbol{\theta}_t^m) p(\boldsymbol{\theta}_t^m|\mathbf{y}_t, \mathbf{X}_t) d\boldsymbol{\theta}_t^m \quad (46)$$

which replaces $p(\boldsymbol{\theta}^m|\mathbf{y}_t, \mathbf{X}_t)$ in (15) and (17) to obtain the predictive pdf $p(y_{t+1}|\mathbf{y}_t, \mathbf{X}_{t+1})$, and the posterior $p(\boldsymbol{\theta}_{t+1}^m|\mathbf{y}_{t+1}, \mathbf{X}_{t+1})$ in the dynamic setting. Specifically for GPR with per-expert Gaussian posterior $p(\boldsymbol{\theta}_t^m|\mathbf{y}_t, \mathbf{X}_t) = \mathcal{N}(\boldsymbol{\theta}_t^m; \hat{\boldsymbol{\theta}}_t^m, \boldsymbol{\Sigma}_t^m)$, the predictive pdf in (46) is $p(\boldsymbol{\theta}_{t+1}^m|\mathbf{y}_t, \mathbf{X}_{t+1}) = \mathcal{N}(\boldsymbol{\theta}_{t+1}^m; \hat{\boldsymbol{\theta}}_t^m, \boldsymbol{\Sigma}_t^m + \sigma_{\epsilon^m}^2 \mathbf{I}_{2D})$. Upon receiving y_{t+1}, DIE-GP then updates the pdf (46) as

$$p(\boldsymbol{\theta}_{t+1}^m|\mathbf{y}_{t+1}, \mathbf{X}_{t+1}) \propto p(\boldsymbol{\theta}_{t+1}^m|\mathbf{y}_t, \mathbf{X}_{t+1}) p(y_{t+1}|\boldsymbol{\theta}_{t+1}^m, \mathbf{x}_{t+1}). \quad (47)$$

Algorithm 2 IE-GPLVM

Initialization:
for $m = 1, \ldots, M$ **do**
  Draw vectors $\{\mathbf{v}_j^m\}_{j=1}^{n_{\mathrm{RF}}} \sim \pi_{\bar{\kappa}^m}(\mathbf{v})$;
  Embed $\mathbf{Y}_{t_0} \to \hat{\mathbf{X}}_{t_0}^m$ and obtain hyperparameters (cf. (57));
  $\mathbf{B}_{t_0}^m = \hat{\boldsymbol{\Phi}}_{t_0}^{m\top} \mathbf{Y}_{t_0}$ with $\hat{\boldsymbol{\Phi}}_{t_0}^m = [\boldsymbol{\phi}_{\mathbf{v}}^m(\hat{\mathbf{x}}_1^m) \ldots \boldsymbol{\phi}_{\mathbf{v}}(\hat{\mathbf{x}}_{t_0}^m)]^\top$;
  $\mathbf{R}_{t_0}^m = \texttt{CholeskyFactor}(\hat{\boldsymbol{\Phi}}_{t_0}^{m\top} \hat{\boldsymbol{\Phi}}_{t_0}^m + \sigma_n^2 \mathbf{I}_{2n_{\mathrm{RF}}})$;
end for

for $t = t_0 + 1, t_0 + 2, \ldots$ **do**
  Receive datum $\mathbf{y}_{t:}$;
  for $m = 1, \ldots, M$ **do**
    Obtain embedding $\hat{\mathbf{x}}_t^m$ based on (54)
    $\mathbf{B}_t^m = \mathbf{B}_{t-1}^m + \boldsymbol{\phi}_{\mathbf{v}}^m(\hat{\mathbf{x}}_t^m)\mathbf{y}_{t:}^\top$
    $\mathbf{R}_t^m = \texttt{CholeskyUpdate}(\mathbf{R}_{t-1}^m, \boldsymbol{\phi}_{\mathbf{v}}^m(\hat{\mathbf{x}}_t^m))$
  end for
  Obtain $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^{m*}$ based on (55);
  Update w_{t+1}^m based on (56);
end for

Remark 2. The dynamics in EGP assembler and GP learners can be readily combined to yield the DSIE-GP generalization. See Fig. 1 for the schematic diagrams of (D,S)IE-GP.

6 EGPs FOR ONLINE UNSUPERVISED LEARNING

Rather than supervised learning, this section deals with EGP-based latent variable model (LVM) for unsupervised dimensionality reduction. Consider first the GPLVM context, where the $D \times 1$ observation $\mathbf{y}_{\tau:} := [y_{\tau,1} \ldots y_{\tau,D}]^\top$[2] is linked with the *unobserved* low-dimensional input $\mathbf{x}_\tau \in \mathbb{R}^d$ $(d < D)$ via [35]

$$y_{\tau j} = f_j(\mathbf{x}_\tau) + n_{\tau j}, \quad j = 1, \ldots, D \quad (48)$$

where $f_j(\cdot) \sim \mathcal{GP}(0, \kappa)$, and $\{n_{tj}\}$ are assumed to be drawn i.i.d. from $\mathcal{N}(0, \sigma_n^2)$.

Given t observations $\mathbf{Y}_t := [\mathbf{y}_{1:} \ldots \mathbf{y}_{t:}]^\top \equiv [\mathbf{y}_t^{(1)} \ldots \mathbf{y}_t^{(D)}]$, where $\mathbf{y}_t^{(j)} := [y_{1,j} \ldots y_{t,j}]^\top$, the estimate of the low-dimensional embedding $\mathbf{X}_t := [\mathbf{x}_1 \ldots \mathbf{x}_t]^\top$ is sought together with the kernel hyperparameters $\boldsymbol{\alpha}$ by solving [35]

$$(\hat{\mathbf{X}}_t, \hat{\boldsymbol{\alpha}}) = \arg\max_{\mathbf{X}_t, \boldsymbol{\alpha}} \log p(\mathbf{Y}_t|\mathbf{X}_t; \boldsymbol{\alpha}) + \log p(\mathbf{X}_t) \quad (49)$$

where $p(\mathbf{Y}_t|\mathbf{X}_t; \boldsymbol{\alpha}) = \prod_{j=1}^D \mathcal{N}(\mathbf{y}_t^{(j)}; \mathbf{0}, \mathbf{K}_t + \sigma_n^2 \mathbf{I}_t)$ is the so-termed marginal likelihood (ML), and a standard choice for the prior of $\mathbf{X}_t$ is $p(\mathbf{X}_t) = \prod_{\tau=1}^t \mathcal{N}(\mathbf{x}_\tau; \mathbf{0}, \sigma_x^2 \mathbf{I}_d)$.

The routine to solve (49) entails inverting the $t \times t$ kernel matrix, thus incurring unaffordable complexity as in the supervised setting [35]. To effect scalablility via the RF approximation and accommodate online kernel adaptation, an IE-GP based LVM is well motivated in accordance with the preceding discussion.

6.1 IE-GPLVM

In the novel IE-GPLVM, the EGP assembler employs an ensmeble of GP experts to independently seek low-dimensional embeddings

2. Notice the introduction of the colon in the subscript, indicating a multivariate observation at time τ; not to be confused with $\mathbf{y}_\tau$ in previous sections.

0162-8828 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Minnesota. Downloaded on July 26,2022 at 19:20:57 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3157197, IEEE Transactions on Pattern Analysis and Machine Intelligence

of the observations. As before, each expert m will leverage the κ^m-induced RF mapping $\boldsymbol{\phi}_{\mathbf{v}}^m(\cdot)$ (8) to yield the per-datum conditional likelihood

$$p(\mathbf{y}_{\tau:}|\boldsymbol{\Theta}^m, \mathbf{x}_\tau) = \prod_{j=1}^{D} \mathcal{N}(y_{\tau j}; \boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}_\tau)\boldsymbol{\theta}_j^m, \sigma_n^2) \tag{50}$$

where the RF-based parameter vectors over D output channels are collected in $\boldsymbol{\Theta}^m := [\boldsymbol{\theta}_1^m, \ldots, \boldsymbol{\theta}_D^m]^\top$, whose prior pdf is given by

$$p(\boldsymbol{\Theta}^m) = \prod_{j=1}^{D} \mathcal{N}(\boldsymbol{\theta}_j^m; \mathbf{0}, \sigma_{\theta^m}^2 \mathbf{I}_{2n_{\text{RF}}}) \ . \tag{51}$$

It is worth mentioning that such an RF-based GPLVM can be regarded as the nonlinear dual form of probabilistic PCA [35].

To incrementally project subsequent observations to the low-dimensional embeddings, each expert m in IE-GPLVM relies on the generative model (50)–(51) to summarize past outputs $\mathbf{Y}_t$ and the estimated inputs $\hat{\mathbf{X}}_t^m$ in the posterior pdf

$$p(\boldsymbol{\Theta}^m|\mathbf{Y}_t, \hat{\mathbf{X}}_t^m) = \prod_{j=1}^{D} \mathcal{N}(\boldsymbol{\theta}_j^m; \hat{\boldsymbol{\theta}}_{t,j}^m, \boldsymbol{\Sigma}_t^m) \tag{52}$$

where the parameter vectors associated with different output channels share the same covariance matrix $\boldsymbol{\Sigma}_t^m$. Further, the per-expert weight assessed by the EGP assembler in IE-GPLVM is adapted with estimated inputs as $w_t^m := \Pr(i = m|\mathbf{Y}_t, \{\hat{\mathbf{X}}_t^\nu\}_{\nu=1}^M))$. In the same spirit as IE-GP, each iteration in IE-GPLVM alternates between estimation of $\mathbf{x}_\tau$ from $\mathbf{y}_{\tau:}$, and correction of the per-expert weight w_t^m and posterior pdf (52). Note that instead of the moments of (52), matrices $\mathbf{A}_t^m := (\boldsymbol{\Sigma}_t^m)^{-1}$ and $\mathbf{B}_t^m := (\boldsymbol{\Sigma}_t^m)^{-1}\hat{\boldsymbol{\Theta}}_t^m$ will be equivalently updated with Chelosky decomposition performed via $\mathbf{A}_t^m = \mathbf{R}_t^{m\top}\mathbf{R}_t^m$ for numerical stability (cf. Alg. 2). However, for consistency with previous sections, the following discussion still uses the moments $\hat{\boldsymbol{\Theta}}_t^m$ and $\boldsymbol{\Sigma}_t^m$.

Estimation. To estimate $\mathbf{x}_{t+1}$ based on $\mathbf{y}_{t+1:}$, expert m capitalizes on the parameter posterior (52) to obtain the conditional likelihood (similar to (22)) as $p(\mathbf{y}_{t+1:}|\mathbf{Y}_t, i=m, \hat{\mathbf{X}}_t^m, \mathbf{x}_{t+1}) = \mathcal{N}(\mathbf{y}_{t+1:}; \hat{\mathbf{y}}_{t+1:}^m(\mathbf{x}_{t+1}), (\sigma_{t+1}^m(\mathbf{x}_{t+1}))^2\mathbf{I}_D)$, where the first two moments are functions of $\mathbf{x}_{t+1}$ as

$$\hat{\mathbf{y}}_{t+1:}^m(\mathbf{x}_{t+1}) = \hat{\boldsymbol{\Theta}}_t^m \boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_{t+1}) \tag{53a}$$

$$(\sigma_{t+1}^m(\mathbf{x}_{t+1}))^2 = \boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}_{t+1})\boldsymbol{\Sigma}_t^m\boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_{t+1}) + \sigma_n^2 \ . \tag{53b}$$

Further imposing a prior on $\mathbf{x}_{t+1}$, the maximum-a-posteriori (MAP) estimate of $\mathbf{x}_{t+1}$ is given by

$$\hat{\mathbf{x}}_{t+1}^m = \underset{\mathbf{x}_{t+1}}{\arg\max}\ \log p(\mathbf{y}_{t+1:}|\mathbf{Y}_t, i=m, \hat{\mathbf{X}}_t^m, \mathbf{x}_{t+1}) + \log p(\mathbf{x}_{t+1}). \tag{54}$$

With the per-expert embeddings $\{\hat{\mathbf{x}}_{t+1}^m\}_{m=1}^M$ at hand, the EGP assembler seeks the final estimate as $\hat{\mathbf{x}}_{t+1} = \hat{\mathbf{x}}_{t+1}^{m^*}$, where

$$m^* = \underset{m\in\mathcal{M}}{\arg\max}\ w_t^m p(\mathbf{y}_{t+1:}|\mathbf{Y}_t, i=m, \hat{\mathbf{X}}_t^m, \hat{\mathbf{x}}_{t+1}^m)\ p(\hat{\mathbf{x}}_{t+1}^m) \ . \tag{55}$$

It can be readily verified that $(m^*, \hat{\mathbf{x}}_{t+1}^{m^*})$ corresponds to the MAP solution of $p(\mathbf{x}_{t+1}, i=m|\mathbf{y}_{t+1:}, \mathbf{Y}_t, \{\hat{\mathbf{X}}_t^\nu\}_{\nu=1}^M)$.

Correction. Upon obtaining the estimate $\hat{\mathbf{x}}_{t+1}^m$, the EGP assembler updates the per-expert weight

$$\begin{aligned} w_{t+1}^m &:= \Pr(i=m|\mathbf{Y}_{t+1}, \{\hat{\mathbf{X}}_{t+1}^\nu\}_{\nu=1}^M) \\ &\propto w_t^m\ p(\mathbf{y}_{t+1:}|\mathbf{Y}_t, i=m, \hat{\mathbf{X}}_t^m, \hat{\mathbf{x}}_{t+1}^m) \end{aligned} \tag{56}$$

which is quite intuitive in that it favors experts which assign higher likelihood to the observation $\mathbf{y}_{t+1:}$. In the long run, we expect the probability mass to concentrate at the expert(s) whose embeddings best describe the observed data. IE-GPLVM is thus effectively performing online kernel selection, adapting to the data as they become available.

Meanwhile, the pair $\{\hat{\mathbf{x}}_{t+1}^m, \mathbf{y}_{t+1:}\}$ allows expert m to update the posterior pdf of $\boldsymbol{\Theta}^m$ as $p(\boldsymbol{\Theta}^m|\mathbf{Y}_{t+1}, \hat{\mathbf{X}}_{t+1}^m) = \prod_{j=1}^D \mathcal{N}(\boldsymbol{\theta}_j^m; \hat{\boldsymbol{\theta}}_{t+1,j}^m, \boldsymbol{\Sigma}_{t+1}^m)$. As previously mentioned, the update of the moments is implemented by propagation of $\mathbf{B}_t^m$ and $\mathbf{R}_t^m$ as shown in Alg. 2, where `CholeskyFactor` computes the Cholesky factor (CF) of its argument and `CholeskyUpdate` performs a rank-one CF update.

A few remarks are in order.

Remark 3. To obtain the kernel hyperparameters, expert m will leverage the first t_0 samples $\mathbf{Y}_{t_0}$ to solve the optimisation problem

$$(\hat{\mathbf{X}}_{t_0}^m, \hat{\boldsymbol{\alpha}}^m) = \underset{\mathbf{X}_t, \boldsymbol{\alpha}}{\arg\max}\ \log p(\mathbf{Y}_{t_0}|\mathbf{X}_{t_0}, i=m; \boldsymbol{\alpha}) + \log p(\mathbf{X}_{t_0}) \tag{57}$$

where the RF-based likelihood for expert m is $p(\mathbf{Y}_{t_0}|i = m, \mathbf{X}_{t_0}; \boldsymbol{\alpha}) = \prod_{j=1}^D \mathcal{N}(\mathbf{y}_{t_0}^{(j)}; \mathbf{0}, \sigma_{\theta^m}^2\boldsymbol{\Phi}_{t_0}^m\boldsymbol{\Phi}_{t_0}^{m\top} + \sigma_n^2\mathbf{I}_{t_0})$ with $\boldsymbol{\Phi}_{t_0}^m := [\boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_1)\ldots\boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_{t_0})]^\top$. With $\hat{\mathbf{X}}_{t_0}^m$ at hand, expert m further relies on the generative model (50)–(51) with inputs replaced by the estimates, to obtain $p(\boldsymbol{\Theta}^m|\mathbf{Y}_{t_0}, \hat{\mathbf{X}}_{t_0}^m) = \prod_{j=1}^D \mathcal{N}(\boldsymbol{\theta}_j^m; \hat{\boldsymbol{\theta}}_{t_0,j}^m, \boldsymbol{\Sigma}_{t_0}^m)$, based on which the upcoming data will be processed incrementally (cf. Alg. 2).

Remark 4. In practice, when solving (54), $\mathbf{x}_{t+1}$ is initialized at the embedding corresponding to the point in $\mathbf{Y}_t$ that is the nearest neighbor (NN) of $\mathbf{y}_{t+1:}$ [38]. Note, however, that since the per incoming observation complexity for obtaining the NN, at slot t scales linearly with t, it can significantly surpass the (constant wrt t) likelihood evaluation complexity of $\mathcal{O}(n_{\text{RF}}^2)$. To obtain a scalable algorithm, we will rely on an approximate kNN search scheme that relies on a hierarchical graph construct [53]. This approach can be shown to (approximately, at the limit) achieve $\mathcal{O}(\log t)$ complexity; see [53] for a detailed description.

Remark 5. Dynamic variants of IE-GPLVM can also been pursued along the lines of Sec. 5.

7 Numerical Tests

To assess performance, real-data tests are presented here for regression, classification as well as dimensionality reduction tasks. The supplementary file also contains synthetic tests that validate the regret bounds in Theorems 1 and 2.

7.1 Regression

Regression tests were performed on the `SARCOS` dataset [1], widely used for evaluating GP-based approaches, as well as on the `Air quality` [54], `Tom's hardware` and `Twitter` datasets [60] from the UCI repository [61]. The statistics of the datasets are summarized in Table 1. We compared the proposed (D)IE-GP approaches with AdaRaker [34], Incremental Sparse Spectrum Gaussian Process Regression (I-SSGPR) [9], and the

0162-8828 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Minnesota. Downloaded on July 26,2022 at 19:20:57 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3157197, IEEE Transactions on Pattern Analysis and Machine Intelligence

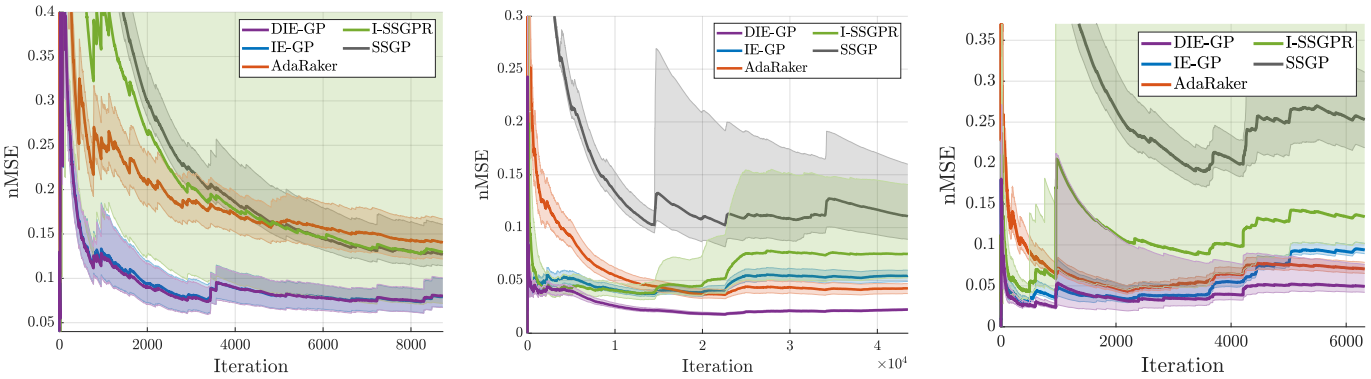


Fig. 2: nMSE plots on (a) `Tom's hardware`; (b) `SARCOS`; and, (c) `Air quality` datasets.

TABLE 1: Statistics of the datasets

Task	**Datasets**	T	d	D
Regression	Tom's hardware	9725	96	1
Regression	SARCOS [1]	44484	21	1
Regression	Air Quality [54]	7322	12	1
Classification	Banana [55]	5300	2	1
Classification	Musk [56]	6598	166	1
Classification	Ionosphere [57]	351	34	1
GPLVMs	USPS [58]	5474	NA	256
GPLVMs	Oil flow [59]	1000	NA	12
GPLVMs	MNIST	70000	NA	784

Streaming Sparse Gaussian Process (SSGP) approach [13], in terms of normalized mean-square error (nMSE) and running time. SIE-GP is not included for comparison since the datasets exhibit no switching behavior among the candidate GP models. With s_y^2 denoting the sample variance of $\mathbf{y}_T$, the nMSE is defined as $\text{nMSE}_t := t^{-1}\sum_{t'=1}^{t}(y_{t'} - \hat{y}_{t'|t'-1})^2/s_y^2$.

For all RF-based approaches (namely (D)IE-GP, AdaRaker and I-SSGPR) we used $2n_{\text{RF}} = 100$ and the reported results correspond to the run which resulted in the median nMSE among 101 runs for the corresponding method. Finally, all reported runtimes include hyperparameter learning/model initialization computations performed on the first $1,000$ samples. If for some expert m and time instance t we have that $w_t^m = 0$, it follows that $w_{t'}^m = 0$ for all $t' > t$ (cf. (20)). Experts with $w_t^m < 10^{-16}$ were deemed inactive for $t' > t$; thus, we set $w_{t'}^m = 0$ for $t' > t$, and avoided unnecessary prediction/correction steps.

The kernel dictionary for (D)IE-GP and AdaRaker comprised radial basis functions (RBFs) with variances from the set $\{10^k\}_{k=-4}^{6}$. The automatic relevance determination (ARD) kernel was used for I-SSGPR, as in [9]. The per kernel noise and prior variances (as well as ARD length scales for I-SSGPR), were estimated by maximizing the marginal likelihood of the first $1,000$ samples using the `minimize` function from the GPML toolbox [62]. The aforementioned samples were not used in the deployment phase. In DIE-GP, $\sigma_{\epsilon^m}^2 = 0.001$ was used for all m and in all experiments. Regarding SSGP, the ARD kernel was used, the batch size was set to 300, the number of inducing points was 100 and the first $1,000$ samples were used for obtaining an initial model, all as per the original work [13].

The nMSE performance of the tested approaches on the `Tom's hardware` dataset is plotted in Fig. 2(a). The proposed (D)IE-GP approaches outperform the competing alternatives in terms of nMSE while also featuring the lowest running time, which corresponds to less than 0.3% of that of the most closely competing (in terms of nMSE) alternative (cf. Fig. 3(a)). Here, the nearly identical performance of IE-GP and DIE-GP can be explained by that `Tom's hardware` dataset exhibits no dynamics. The results on the `SARCOS` dataset are depicted in Fig. 2(b). Our IE-GP remains competitive whereas the proposed dynamic variant (DIE-GP) features the lowest nMSE, while also achieving both faster convergence as well as a runtime that is an order of magnitude lower than that of the second best (in terms of nMSE) approach (cf. Fig. 3(b)). These results further highlight the computational efficiency of the proposed approaches. Similar observations can be made on the `Air quality` (cf. Figs. 2,3(c)).

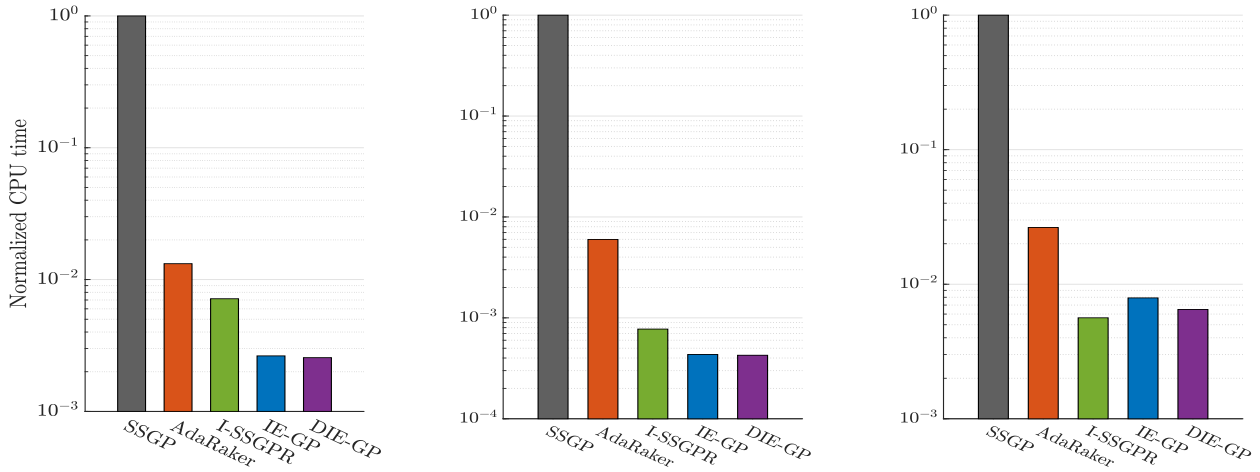
To further demonstrate (D)IE-GP's uncertainty quantification performance, tests were conducted among GP-based approaches regarding the predictive negative log-likelihood (pnLL) as $\text{pnLL}_t := -\log p(y_t|\mathbf{y}_{t-1}, \mathbf{X}_t)$, which is computable from (16). As illustrated in Fig. **??** in the Appendix, (D)IE-GP always outperform I-SSGPR; while they outperform SSGP in `SARCOS` and `air-quality` datasets; they are comparable in the `Twitter` dataset; and perform inferior to SSGP on the `Tom's hardware` dataset, even though SSGP is two orders of magnitude slower than (D)IE-GP. The supplementary file contains additional results concerning the comparison of (D)IE-GP with local GPs [63] (cf. Fig. **??**) and the effect of number n_{RF} of spectral features on nMSE performance (cf. Fig. **??**).

7.2 Classification

Coupled with the logistic likelihood, our IE-GP and the switching variant were tested for binary classification using Laplace approximation (cf. Sec. 3.2). The performance of (S)IE-GP were also compared with AdaRaker [34] and SSGP [13] in terms of classification error and running time on the `Banana`, `Musk`, and `Ionosphere` datasets whose statistics are provided in Table 1. DIE-GP are not included for comparison since it achieves similar performance relative to IE-GP. For (S)IE-GP and AdaRaker, the value of n_{RF} was set to 15, and the kernel dictionary is the same as in the regression test. Regarding SSGP, the number of inducing points was 30, the batch size was chosen to be 40, and the first 20% of the samples were used for model initialization. For (S)IE-GP, the kernel magnitude $\sigma_{\theta^m}^2$, the only hyperparameter per expert, was obtained by maximizing the marginal likelihood using Laplace apprximation [1, Chapter 5.5.1].

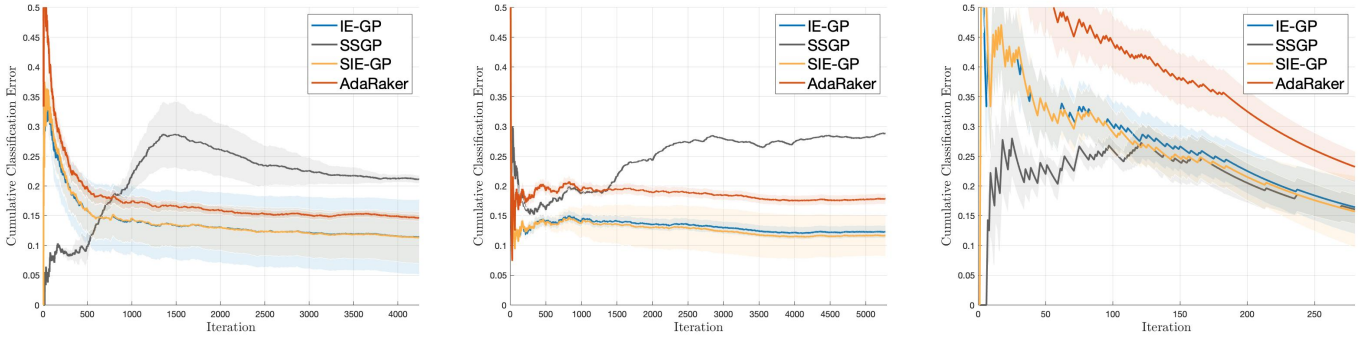
0162-8828 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Minnesota. Downloaded on July 26,2022 at 19:20:57 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3157197, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) (b) (c)

Fig. 3: Normalized running times for regression on (a) `Tom's hardware`; (b) `SARCOS`; and, (c) `Air quality` datasets.



(a) (b) (c)

Fig. 4: Cumulative classification errors on (a) `Banana`, (b) `Musk`, and (c) `Ionosphere` datasets.

The cumulative classification error and running time of the four competing approaches are plotted in Figs. 4–5. Clearly, (S)IE-GP outperforms SSGP and AdaRaker in both classification accuracy and computational efficiency on the `Banana` and `Musk` datasets. Although achieving lower classification error than (S)IE-GP on the `Ionosphere` dataset, SSGP runs more than two orders of magnitude slower. Also, it is worth mentioning that the performance of SSGP depends on the batch size. Decreasing its value from 40 to 20 yields SSGP performing inferior to (S)IE-GP in classification accuracy. Regarding the two proposed approaches, SIE-GP achieves comparable classification accuracy relative to the static IE-GP since the tested datasets exhibit negligible switching dynamics among the candidate models. In addition, SIE-GP's higher running time is explained by the fact that all the GP experts update at all slots to detect possible model switching, whereas IE-GP implements expert shutdown for computational efficiency.

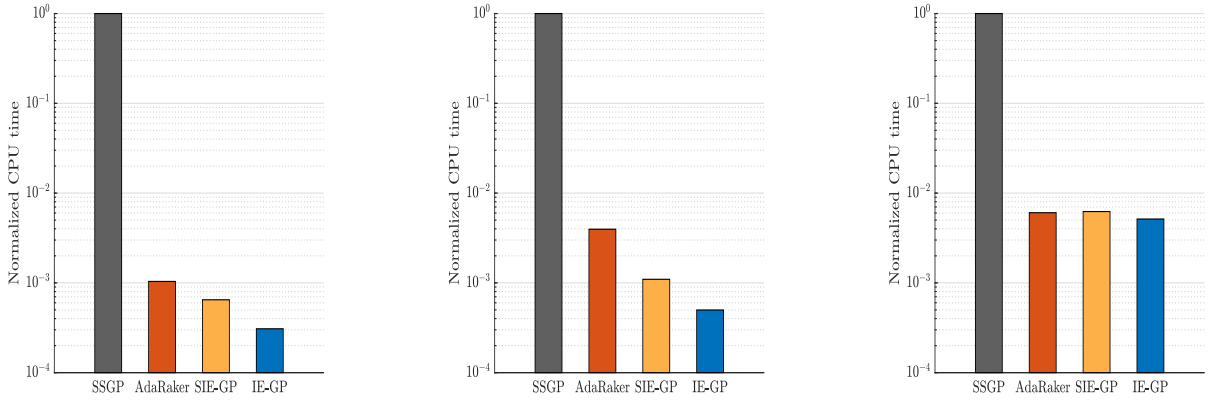
7.3 Dimensionality reduction

Tests for dimensionality reduction were performed on several benchmark datasets, including `MNIST` ($D = 784$) as well as the `oil` flow data ($D = 12$) [59], and the subset of the `USPS` handwritten digits set ($D = 256$) comprising digits $0 - 4$. . The latter two datasets were also used in the original GPLVM paper [35]. Several competing alternatives were considered. GPLVM based methods comprise the original GPLVM [35], [36], a variational inference based scheme (varGPLVM) [37], as well as an online GPLVM variant (onGPLVM); see Alg. 2 in [38]. PCA based alternatives encompass online PCA [64], [65], and (batch) kernel PCA [66]. The embedding dimensionality was set to $d = 2$, and the results presented correspond to the median across 11 trials. Regarding the proposed IE-GPLVM scheme, $n_{\text{RF}} = 50$ random features were used, each expert relied on a RBF kernel with variance taken from the set $\{2^k\}_{k=-3}^{3}$, t_0 was set to 5% of the number of samples for `MNIST` and to 10% for the remaining (smaller) datasets, and (57) was additionally optimized over σ_n^2. For the GPLVM based methods, the RBF kernel was used, 100 inducing points were utilized, and the maximum number of iterations was set to $1,000$. Initializations were provided by means of PPCA embeddings. Finally, for kernel PCA (kPCA), a grid search was performed over RBF kernels with variances in $\{2^k\}_{k=-10}^{10}$ and the lowest error rate achieved is reported. Note that for kPCA the reported runtime does not include the computational time required for the grid search.

The error rate of the nearest neighbor classification rule was used as the performance metric, when applied to the resultant embeddings; see e.g. [35]. The results for the three tested datasets are summarized in Fig. 6, in the form of error rate versus runtime plots. In the `MNIST` dataset, the proposed `IE-GPLVM` approach achieves both the lowest overall error rate, as well as runtime among GPLVM schemes. The only method achieving a somewhat

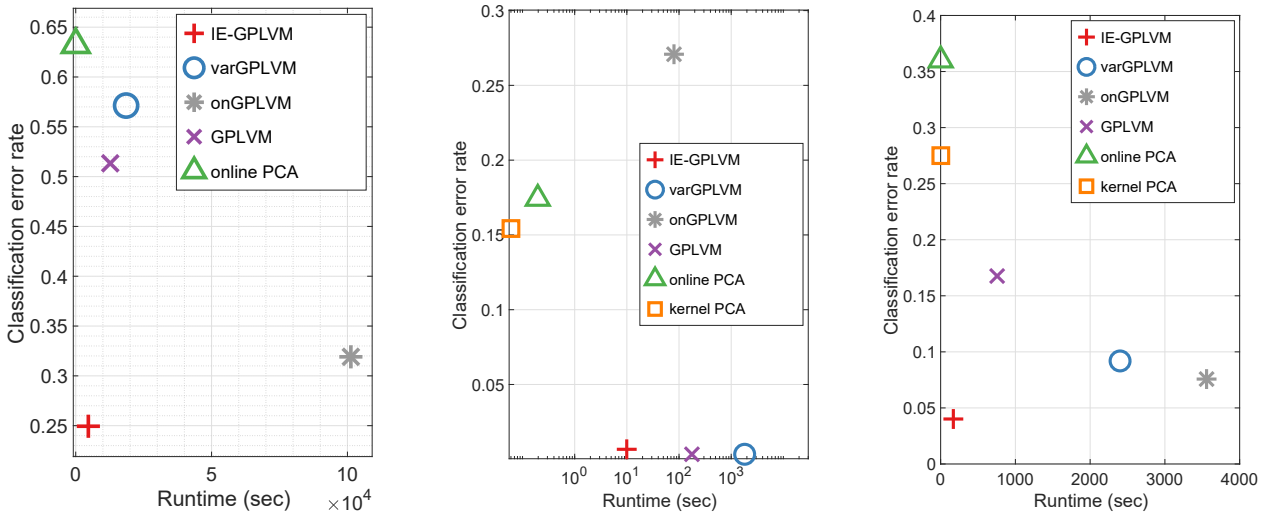
0162-8828 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Minnesota. Downloaded on July 26,2022 at 19:20:57 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3157197, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) (b) (c)

Fig. 5: Normalized running times for classification on (a) `Banana`, (b) `Musk`, and (c) `Ionosphere` datasets.



(a) (b) (c)

Fig. 6: Classification error versus runtime plots for dimensionality reduction on the (a) `MNIST`, (b) `oil`, and (c) `USPS` datasets.

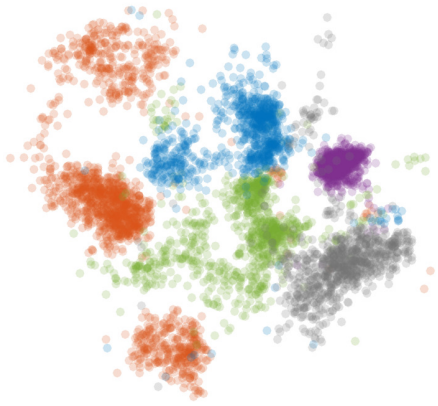


Fig. 7: Visualization of the embedding attained by IE-GPLVM on the `USPS` dataset. Colors represent different digits.

similar error rate, namely `onGPLVM`, has a runtime that is more than 20 times higher relative to our approach. In the `oil` dataset, our IE-GPLVM achieves similar error rate (less than 1%) to `GPLVM` and `varGPLVM`, while being more than one and two orders of magnitude faster, respectively. In the `USPS` set, our approach achieves the lowest overall error rate. The only schemes that achieve error rates in the same order of magnitude, namely `varGPLVM` and `onGPLVM`, have runtimes that are 14 and 21 times higher, respectively. Finally, in both experiments, PCA based schemes, although computationally efficient, yield high error rates. A visualization of the embedding attained is provided for the proposed IE-GPLVM on the `USPS` dataset (Fig. 7). We can observe that good separation between clusters of different digits is achieved, in line with the low classification error rate in Fig. 6.

8 Conclusions

This paper put forth an incremental scheme that leverages an ensemble of scalable RF-based parametric GP learners to jointly infer the unknown function along with its performance, and a data-driven kernel combination. Dynamic function learning was enabled through modeling structured dynamics for the EGP assembler and individual GP learners. On the theoretical aspect, regret analysis was conducted to benchmark even in adversarial settings the novel IE-GP and its dynamic variant relative to

0162-8828 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Minnesota. Downloaded on July 26,2022 at 19:20:57 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3157197, IEEE Transactions on Pattern Analysis and Machine Intelligence

benchmark strategies with data in hindsight. Further, EGP-based latent variable model is devised for online kernel-adaptive dimensionality reduction. Extensive experimental results are provided to illustrate the superior performance of the novel IE-GP schemes.

9 PROOFS

9.1 Proof of Lemma 1

To prove Lemma 1, we will first upper bound the cumulative online Bayesian loss associated with IE-GP, $\sum_{\tau=1}^{T}\ell_{\tau|\tau-1}$, relative to that incurred by any RF-based GP expert m, namely $\sum_{\tau=1}^{T}l^m_{\tau|\tau-1}$. Reorganizing (20), we have $\exp(-\ell_{\tau|\tau-1})/\exp(-l^m_{\tau|\tau-1})=w^m_{\tau-1}/w^m_{\tau}$, multiplying which (20) from $\tau=1$ to T, it follows that $\exp(-\sum_{\tau=1}^{T}\ell_{\tau|\tau-1}+\sum_{\tau=1}^{T}l^m_{\tau|\tau-1})=1/(Mw^m_T)$, yielding

$$\sum_{\tau=1}^{T}\ell_{\tau|\tau-1}-\sum_{\tau=1}^{T}l^m_{\tau|\tau-1}=\log M+\log w^m_T\overset{(a)}{\leq}\log M \tag{58}$$

where (a) holds because $w^m_T\in[0,1]$.

Next, we will bound the difference between $\sum_{\tau=1}^{T}l^m_{\tau|\tau-1}$ and the cumulative loss incurred by a fixed strategy $\boldsymbol{\theta}^m_*$, for any expert $m\in\mathcal{M}$. For notational brevity, we will drop expert index m in the remaining of the proof.

Upon defining the cumulative loss over T slots with a time-invariant $\boldsymbol{\theta}$ as $\mathcal{L}_\theta:=\sum_{\tau=1}^{T}\mathcal{L}(\boldsymbol{\phi}^\top_{\mathbf{v}}(\mathbf{x}_\tau)\boldsymbol{\theta};y_\tau)=-\log p(\mathbf{y}_T|\boldsymbol{\theta},\mathbf{X}_T)$, the expected cumulative loss over any pdf $q(\boldsymbol{\theta})$ is [47]

$$\bar{\mathcal{L}}_{q_\theta}:=\mathbb{E}_q[\mathcal{L}_\theta]=\int_{\boldsymbol{\theta}}q(\boldsymbol{\theta})\mathcal{L}_\theta d\boldsymbol{\theta}\ .$$

On the other hand, the following equality holds for the cumulative online Bayesian loss based on Bayes' rule

$$\sum_{\tau=1}^{T}\ell_{\tau|\tau-1}=\sum_{\tau=1}^{T}-\log p(y_\tau|\mathbf{y}_{\tau-1},\mathbf{X}_\tau)=-\log p(\mathbf{y}_T|\mathbf{X}_T)\ .$$

Let $q(\boldsymbol{\theta})=\mathcal{N}(\boldsymbol{\theta};\boldsymbol{\theta}_*,\xi^2\mathbf{I}_{2n_{\text{RF}}})$ with the variational parameter ξ to be tuned later, and $p(\boldsymbol{\theta})=\mathcal{N}(\boldsymbol{\theta};\mathbf{0},\sigma^2_\theta\mathbf{I}_{2n_{\text{RF}}})$. It then follows that

$$\begin{aligned}
&\sum_{\tau=1}^{T}l_{\tau|\tau-1}-\bar{\mathcal{L}}_{q_\theta}=\int q(\boldsymbol{\theta})\log\frac{p(\mathbf{y}_T|\boldsymbol{\theta},\mathbf{X}_T)}{p(\mathbf{y}_T|\mathbf{X}_T)}d\boldsymbol{\theta}\\
&\overset{(a)}{=}\int q(\boldsymbol{\theta})\log\frac{p(\boldsymbol{\theta}|\mathbf{y}_T,\mathbf{X}_T)}{p(\boldsymbol{\theta})}d\boldsymbol{\theta}\\
&=\int q(\boldsymbol{\theta})\log\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})}d\boldsymbol{\theta}-\int q(\boldsymbol{\theta})\log\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathbf{y}_T,\mathbf{X}_T)}d\boldsymbol{\theta}\\
&=\text{KL}(q(\boldsymbol{\theta})\|p(\boldsymbol{\theta}))-\text{KL}(q(\boldsymbol{\theta})\|p(\boldsymbol{\theta}|\mathbf{y}_T,\mathbf{X}_T))\overset{(b)}{\leq}\text{KL}(q(\boldsymbol{\theta})\|p(\boldsymbol{\theta}))\\
&=2n_{\text{RF}}\log\sigma_\theta+\frac{\|\boldsymbol{\theta}_*\|^2+2n_{\text{RF}}\xi^2}{2\sigma^2_\theta}-n_{\text{RF}}-2n_{\text{RF}}\log\xi
\end{aligned}\tag{59}$$

where (a) holds since Bayes' rule yields $p(\mathbf{y}_T|\boldsymbol{\theta},\mathbf{X}_T)p(\boldsymbol{\theta})=p(\mathbf{y}_T|\mathbf{X}_T)p(\boldsymbol{\theta}|\mathbf{y}_T,\mathbf{X}_T)$; and, (b) comes from the fact that $\text{KL}(q(\boldsymbol{\theta})\|p(\boldsymbol{\theta}|\mathbf{y}_T,\mathbf{X}_T))\geq 0$.

The last step towards bounding $\sum_{\tau=1}^{T}l_{\tau|\tau-1}-\mathcal{L}_{\theta_*}$ is to establish an upper bound for $\bar{\mathcal{L}}_{q_\theta}-\mathcal{L}_{\theta_*}$. To this end, let $z_\tau=\boldsymbol{\phi}^\top_{\mathbf{v}}(\mathbf{x}_\tau)\boldsymbol{\theta}$ and $z^*_\tau=\boldsymbol{\phi}^\top_{\mathbf{v}}(\mathbf{x}_\tau)\boldsymbol{\theta}_*$. Taking the Taylor's expansion of $\mathcal{L}(z_\tau;y_\tau)$ around z^*_τ, yields

$$\begin{aligned}
\mathcal{L}(z_\tau;y_\tau)=\mathcal{L}(z^*_\tau;y_\tau)+\frac{d\mathcal{L}(z^*_\tau;y_\tau)}{dz_\tau}(z_\tau-z^*_\tau)\\
+\frac{d^2}{dz^2_\tau}\mathcal{L}(h(z_\tau);y_\tau)\frac{(z_\tau-z^*_\tau)^2}{2}
\end{aligned}\tag{60}$$

where $h(z_\tau)$ is some function lying between z_τ and z^*_τ. Taking the expectation of (60) wrt $q(\boldsymbol{\theta})$, leads to

$$\begin{aligned}
\mathbb{E}_q[\mathcal{L}(z_\tau;y_\tau)]-\mathcal{L}(z^*_\tau;y_\tau)&=\mathbb{E}_q\left[\frac{d^2}{dz^2_\tau}\mathcal{L}(h(z_\tau);y_\tau)\frac{(z_\tau-z^*_\tau)^2}{2}\right]\\
&\overset{(a)}{\leq}c\mathbb{E}\left[\frac{(z_\tau-z^*_\tau)^2}{2}\right]\overset{(b)}{\leq}\frac{c\xi^2}{2}
\end{aligned}\tag{61}$$

where (a) makes use of (as1) that $\left|\frac{d^2}{dz^2}\mathcal{L}(z;y)\right|\leq c\ \forall z$, and (b) relies on the equality $\|\boldsymbol{\phi}_{\mathbf{v}}(\mathbf{x}_\tau)\|^2=1$.

Summing (61) from $\tau=1$ to T, we have

$$\bar{\mathcal{L}}_{q_\theta}\leq\mathcal{L}_{\theta_*}+\frac{Tc\xi^2}{2}\tag{62}$$

which, in conjunction with (59), yields the inequality

$$\begin{aligned}
&\sum_{\tau=1}^{T}l_{\tau|\tau-1}-\mathcal{L}_{\theta_*}\\
&\leq\frac{Tc\xi^2}{2}+2n_{\text{RF}}\log\sigma_\theta+\frac{\|\boldsymbol{\theta}_*\|^2+2n_{\text{RF}}\xi^2}{2\sigma^2_\theta}-n_{\text{RF}}-2n_{\text{RF}}\log\xi\ .
\end{aligned}\tag{63}$$

Replacing the RHS of (63), a convex function of ξ, with the minimal value taken at $\xi^2=\frac{2n_{\text{RF}}\sigma^2_\theta}{2n_{\text{RF}}+Tc\sigma^2_\theta}$, simplifies (63) for any expert $m\in\mathcal{M}$ to

$$\sum_{\tau=1}^{T}l^m_\tau-\sum_{\tau=1}^{T}\mathcal{L}\left(\boldsymbol{\phi}^{m\top}_{\mathbf{v}}(\mathbf{x}_\tau)\boldsymbol{\theta}^m_*;y_\tau\right)\leq\frac{\|\boldsymbol{\theta}^m_*\|^2}{2\sigma^2_{\theta^m}}+n_{\text{RF}}\log\left(1+\frac{Tc\sigma^2_{\theta^m}}{2n_{\text{RF}}}\right)$$

which, together with (58), readily prove Lemma 1.

9.2 Proof of Theorem 1

For a given shift-invariant standardized kernel $\bar{\kappa}^m$, the maximum point-wise error of the RF kernel approximant is uniformly bounded with probability at least $1-2^8(\frac{\sigma_m}{\epsilon})^2\exp\left(\frac{-n_{\text{RF}}\epsilon^2}{4d+8}\right)$ [8]

$$\sup_{\mathbf{x}_i,\mathbf{x}_j\in\mathcal{X}}\left|\boldsymbol{\phi}^{m\top}_{\mathbf{v}}(\mathbf{x}_i)\boldsymbol{\phi}^m_{\mathbf{v}}(\mathbf{x}_j)-\bar{\kappa}^m(\mathbf{x}_i,\mathbf{x}_j)\right|<\epsilon\tag{64}$$

where ϵ is a given constant, n_{RF} is the number of spectral feature vectors, d is the dimension of $\mathbf{x}$, and $\sigma^2_m:=\mathbb{E}_{\pi^m_{\bar{\kappa}}}[\|\mathbf{v}^m\|^2]$ is the second-order moment of the RF vector $\mathbf{v}^m$.

The optimal function estimator in $\mathcal{H}^m$ incurred by κ^m is

$$\hat{f}^m(\mathbf{x}):=\sum_{\tau=1}^{T}\hat{\alpha}^m_\tau\kappa^m(\mathbf{x},\mathbf{x}_\tau)=\sigma^2_{\theta^m}\sum_{\tau=1}^{T}\hat{\alpha}^m_\tau\bar{\kappa}^m(\mathbf{x},\mathbf{x}_\tau)\tag{65}$$

and its RF-based approximant is $\check{f}^m_*(\mathbf{x}):=\boldsymbol{\phi}^{m\top}_{\mathbf{v}}(\mathbf{x})\boldsymbol{\theta}^m_*$ with

0162-8828 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Minnesota. Downloaded on July 26,2022 at 19:20:57 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3157197, IEEE Transactions on Pattern Analysis and Machine Intelligence

$\boldsymbol{\theta}_*^m := \sigma_{\theta^m}^2 \sum_{\tau=1}^T \hat{\alpha}_\tau^m \boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_\tau)$. We then have that

$$\begin{aligned}
&\left|\sum_{\tau=1}^T \mathcal{L}\left(\check{f}_*^m(\mathbf{x}_\tau); y_\tau\right) - \sum_{\tau=1}^T \mathcal{L}\left(\hat{f}^m(\mathbf{x}_\tau); y_\tau\right)\right| \qquad (66)\\
&\overset{(a)}{\leq} \sum_{\tau=1}^T \left|\mathcal{L}\left(\check{f}_*^m(\mathbf{x}_\tau); y_\tau\right) - \mathcal{L}\left(\hat{f}^m(\mathbf{x}_\tau); y_\tau\right)\right|\\
&\overset{(b)}{\leq} \sum_{\tau=1}^T L\sigma_{\theta^m}^2 \left|\sum_{\tau'=1}^T \hat{\alpha}_{\tau'}^m \boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}_\tau)\boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_{\tau'}) - \sum_{\tau'=1}^T \hat{\alpha}_{\tau'}^m \bar{\kappa}^m(\mathbf{x}_\tau, \mathbf{x}_{\tau'})\right|\\
&\overset{(c)}{\leq} \sum_{\tau=1}^T L\sigma_{\theta^m}^2 \sum_{\tau'=1}^T |\hat{\alpha}_{\tau'}^m| \left|\boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}_\tau)\boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_{\tau'}) - \bar{\kappa}^m(\mathbf{x}_\tau, \mathbf{x}_{\tau'})\right|
\end{aligned}$$

where (a) follows from the triangle inequality; (b) makes use of (as2), which states the convexity and bounded derivative of $\mathcal{L}(z; y)$ wrt z, and (c) results from the Cauchy-Schwarz inequality. Leveraging (64) to upper bound the RHS of (66), we find

$$\begin{aligned}
&\left|\sum_{\tau=1}^T \mathcal{L}\left(\check{f}_*^m(\mathbf{x}_\tau); y_\tau\right) - \sum_{\tau=1}^T \mathcal{L}\left(\hat{f}^m(\mathbf{x}_\tau); y_\tau\right)\right|\\
&\leq \sum_{\tau=1}^T L\sigma_{\theta^m}^2 \epsilon \sum_{\tau=1}^T |\hat{\alpha}_\tau^m| \leq \epsilon LTC, \text{ w.h.p.} \qquad (67)
\end{aligned}$$

where $C := \max_{m\in\mathcal{M}} \sum_{\tau=1}^T \sigma_{\theta^m}^2 |\hat{\alpha}_\tau^m|$. It thus holds w.h.p. that

$$\sum_{\tau=1}^T \mathcal{L}\Big(\boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}_\tau)\boldsymbol{\theta}_*^m; y_\tau\Big) - \sum_{\tau=1}^T \mathcal{L}\Big(\hat{f}^m(\mathbf{x}_\tau); y_\tau\Big) \leq \epsilon LTC. \qquad (68)$$

On the other hand, the uniform convergence bound in (64) and (as3) imply w.h.p. that

$$\sup_{\mathbf{x}_\tau, \mathbf{x}_{\tau'} \in \mathcal{X}} \boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}_\tau)\boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_{\tau'}) \leq 1+\epsilon \qquad (69)$$

based on which

$$\begin{aligned}
\|\boldsymbol{\theta}_*^m\|^2 &:= \left\|\sigma_{\theta^m}^2 \sum_{\tau=1}^T \hat{\alpha}_\tau^m \boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_\tau)\right\|^2 \qquad (70)\\
&= \sigma_{\theta^m}^4 \sum_{\tau=1}^T \sum_{\tau'=1}^T \hat{\alpha}_\tau^m \hat{\alpha}_{\tau'}^m \boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}_\tau)\boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_{\tau'}) \leq (1+\epsilon)C^2 .
\end{aligned}$$

Hence, in conjunction with (70), (68) and Lemma 1, it follows for any $m \in \mathcal{M}$ that

$$\begin{aligned}
&\sum_{\tau=1}^T \ell_{\tau|\tau-1} - \sum_{\tau=1}^T \mathcal{L}(\hat{f}^m(\mathbf{x}_\tau); y_\tau) \qquad (71)\\
&\leq n_{\text{RF}} \log\left(1 + \frac{Tc\sigma_{\theta^m}^2}{2n_{\text{RF}}}\right) + \log M + \frac{(1+\epsilon)C^2}{2\sigma_{\theta^m}^2} + \epsilon LTC
\end{aligned}$$

thus completing the proof of Theorem 1 with $m = m^*$.

9.3 Proof of Lemma 2

Inspired by [11], the proof of Lemma 2 will be conducted relying on the notion of compound experts, each associated with a sequence of contributing models over T slots, denoted as $\mathbf{i}_T = [i_1, \ldots, i_T]^\top$. Let $\bar{w}_t(\mathbf{i}_T)$ represent the posterior weight of compound expert $\mathbf{i}_T$ at slot t as $\bar{w}_t(\mathbf{i}_T) := \Pr(\mathbf{i}_T|\mathbf{y}_t, \mathbf{X}_t)$, which is updated at slot $t+1$ as

$$\begin{aligned}
\bar{w}_{t+1}(\mathbf{i}_T) &= \frac{\bar{w}_t(\mathbf{i}_T) p(y_{t+1}|\mathbf{y}_t, \mathbf{i}_T, \mathbf{X}_{t+1})}{p(y_{t+1}|\mathbf{y}_t, \mathbf{X}_{t+1})}\\
&\overset{(a)}{=} \bar{w}_t(\mathbf{i}_T) \exp\left(\bar{\ell}_{t+1|t} - l_{t+1|t}^{i_{t+1}}\right) \qquad (72)
\end{aligned}$$

where equality (a) holds since $p(y_{t+1}|\mathbf{y}_t, \mathbf{i}_T, \mathbf{X}_{t+1}) = p(y_{t+1}|\mathbf{y}_t, i_{t+1}, \mathbf{X}_{t+1})$, and $\bar{\ell}_{t+1|t}$ signifies the ensemble online loss accounting for all the compound experts as

$$\begin{aligned}
\bar{\ell}_{t+1|t} &:= -\log \sum_{i_1,\ldots,i_T} \bar{w}_t(\mathbf{i}_T) p(y_{t+1}|\mathbf{y}_t, \mathbf{i}_T, \mathbf{X}_{t+1})\\
&= -\log \sum_{i_{t+1}} \bar{w}_{t+1}^{i_{t+1}} \exp(-l_{t+1|t}^{i_{t+1}}) \qquad (73)
\end{aligned}$$

with marginal weight at slot $t+1$ given by

$$\bar{w}_{t+1}^{i_{t+1}} := \sum_{i_1,\ldots,i_t,i_{t+2},\ldots,i_T} \bar{w}_t(\mathbf{i}_T) . \qquad (74)$$

Next, the equivalence of $\bar{\ell}_{t+1|t}$ and $\ell_{t+1|t}^{\text{SW}}$ (38) will be established via proving the equality $\bar{w}_{t+1}^{i_{t+1}} = w_{t+1|t}^{i_{t+1}}$ by induction. Towards this, we will first leverage (72) to obtain

$$\bar{w}_t(\mathbf{i}_T) \propto \bar{w}_0(\mathbf{i}_T) \exp\left(\sum_{\tau=1}^t -l_{\tau|\tau-1}^{i_\tau}\right)$$

based on which, (74) can be rewritten as

$$\begin{aligned}
\bar{w}_{t+1}^{i_{t+1}} &\propto \sum_{i_1,\ldots,i_t} \bar{w}_0(\mathbf{i}_{t+1}) \exp\left(\sum_{\tau=1}^t -l_{\tau|\tau-1}^{i_\tau}\right)\\
&= \sum_{i_t} \frac{\bar{w}_0(\mathbf{i}_{t+1})}{\bar{w}_0(\mathbf{i}_t)} e^{-l_{t|t-1}^{i_t}} \sum_{i_1,\ldots,i_{t-1}} \bar{w}_0(\mathbf{i}_t) \exp\left(\sum_{\tau=1}^{t-1} -l_{\tau|\tau-1}^{i_\tau}\right)\\
&\propto \sum_{i_t} \Pr(i_{t+1}|i_t) \exp\left(-l_{t|t-1}^{i_t}\right) \bar{w}_t^{i_t}\\
&\overset{(a)}{=} \sum_{i_t} \Pr(i_{t+1}|i_t) \exp\left(-l_{t|t-1}^{i_t}\right) w_{t|t-1}^{i_t}\\
&\overset{(b)}{\propto} \sum_{i_t} \Pr(i_{t+1}|i_t) w_{t|t}^{i_t} \overset{(c)}{=} w_{t+1|t}^{i_{t+1}} \qquad (75)
\end{aligned}$$

where (a) follows due to the induction assumption that $\bar{w}_t^{i_t} = w_{t|t-1}^{i_t}$; and, (b) and (c) are based on (39) and (36), respectively.

With $\bar{\ell}_{t+1|t} = \ell_{t+1|t}^{\text{SW}}$ being established according to (75), multiplying (72) from $t=1$ to T yields

$$\begin{aligned}
\sum_{\tau=1}^T \ell_{\tau+1|\tau}^{\text{SW}} - \sum_{\tau=1}^T l_{\tau+1|\tau}^{i_{\tau+1}} &= \log \bar{w}_T(\mathbf{i}_T) - \log \bar{w}_0(\mathbf{i}_T)\\
&\leq -\log \bar{w}_0(\mathbf{i}_T) . \qquad (76)
\end{aligned}$$

Since $\bar{w}_0(\mathbf{i}_T) = \Pr(i_1) \prod_{\tau=2}^T \Pr(i_\tau|i_{\tau-1}) = \frac{1}{M} q_0^{T-s} q_1^s$ with s denoting the number of switches in $\mathbf{i}_T$, (76) can be rewritten as

$$\begin{aligned}
\sum_{\tau=1}^T \ell_{\tau+1|\tau}^{\text{SW}} - \sum_{\tau=1}^T l_{\tau+1|\tau}^{i_{\tau+1}} &\leq \log M - (T-s)\log q_0 - s\log q_1\\
&\overset{(b)}{\leq} \log M - T\log q_0 + S\log\frac{q_0}{1-q_0} \qquad (77)
\end{aligned}$$

where the inequality in (a) results from $s \leq S$ and $q_0 \geq q_1 = 1 - q_0$ based on (as4)-(as5). As the RHS of (77) is a convex function of q_0, the following holds true upon setting it to its minimal value

0162-8828 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Minnesota. Downloaded on July 26,2022 at 19:20:57 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3157197, IEEE Transactions on Pattern Analysis and Machine Intelligence

taken at $q_0^* = (T-S)/T$

$$\sum_{\tau=1}^{T} \ell^{\text{SW}}_{\tau+1|\tau} - \sum_{\tau=1}^{T} l^{i_{\tau+1}}_{\tau+1|\tau} \leq \log M - S\log\frac{S}{T} + (T-S)\log\frac{T}{T-S}$$

which, upon leveraging $(T-S)\log\frac{T}{T-S} \leq (T-S)\frac{S}{T-S} = S$ for $S \ll T$, yields Lemma 2.

9.4 Proof of Lemma 3

For any sequence $\mathbf{i}_T$, the cumulative loss over T slots measured by negative log-likelihood for fixed parameter set $\boldsymbol{\Theta} := \{\boldsymbol{\Theta}^m\}_{m=1}^M$ is given by $\mathcal{L}_{\Theta}^{\mathbf{i}_T} := -\log p(\mathbf{y}_T|\boldsymbol{\Theta}, \mathbf{i}_T, \mathbf{X}_T) = \sum_{\tau=1}^{T} \mathcal{L}(\boldsymbol{\phi}_{\mathbf{v}}^{i_\tau\top}(\mathbf{x}_\tau)\boldsymbol{\theta}^{i_\tau}; y_\tau)$, whose expected value over factorized pdf $q(\boldsymbol{\Theta}) = \prod_{m=1}^M q(\boldsymbol{\theta}^m) = \prod_{m=1}^M \mathcal{N}(\boldsymbol{\theta}^m; \boldsymbol{\theta}_*^m, \xi_m^2\mathbf{I}_{2D},)$ is

$$\bar{\mathcal{L}}_{q_\Theta}^{\mathbf{i}_T} := \int \mathcal{L}_{\Theta}^{\mathbf{i}_T} q(\boldsymbol{\Theta}) d\boldsymbol{\Theta} = \sum_{\tau=1}^{T} \int \mathcal{L}(\boldsymbol{\phi}_{\mathbf{v}}^{i_\tau\top}(\mathbf{x}_\tau)\boldsymbol{\theta}^{i_\tau}; y_\tau) q(\boldsymbol{\theta}^{i_\tau}) d\boldsymbol{\theta}^{i_\tau}$$

which can be re-expressed as $\bar{\mathcal{L}}_{q_\Theta}^{\mathbf{i}_T} = \sum_{m=1}^M \bar{\mathcal{L}}_{q_\Theta}^m$, where

$$\bar{\mathcal{L}}_{q_\Theta}^m := \sum_{\tau\in\mathcal{T}_m} \int \mathcal{L}(\boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}_\tau)\boldsymbol{\theta}^m; y_\tau) q(\boldsymbol{\theta}^m) d\boldsymbol{\theta}^m$$

with $\mathcal{T}_m$ collecting the T_m slot indices when GP model from expert m is in action, that is, $\mathcal{T}_m := \{\tau | i_\tau = m, 1 \leq \tau \leq T\}$.

On the other hand, the cumulative online loss for any $\mathbf{i}_T$ is

$$\sum_{\tau=1}^{T} l^{i_\tau}_{\tau|\tau-1} = \sum_{\tau=1}^{T} -\log p(y_\tau|\mathbf{y}_{\tau-1}, i_\tau, \mathbf{X}_\tau) = -\log p(\mathbf{y}_T|\mathbf{i}_T, \mathbf{X}_T).$$

With $p(\boldsymbol{\Theta}) = \prod_{m=1}^M p(\boldsymbol{\theta}^m) = \prod_{m=1}^M \mathcal{N}(\boldsymbol{\theta}^m; \mathbf{0}_{2n_{\text{RF}}}, \sigma_{\theta^m}^2\mathbf{I}_{2n_{\text{RF}}})$, the following inequality can be proved in accordance with (59)

$$\begin{aligned}
&\sum_{\tau=1}^{T} l^{i_\tau}_{\tau|\tau-1} - \bar{\mathcal{L}}_{q_\Theta}^{\mathbf{i}_T} = \int q(\boldsymbol{\Theta}) \log\frac{p(\mathbf{y}_T|\boldsymbol{\Theta}, \mathbf{i}_T, \mathbf{X}_T)}{p(\mathbf{y}_T|\mathbf{i}_T, \mathbf{X}_T)} d\boldsymbol{\Theta} \\
&= \int q(\boldsymbol{\Theta}) \log\frac{p(\boldsymbol{\Theta}|\mathbf{y}_T, \mathbf{i}_T, \mathbf{X}_T)}{p(\boldsymbol{\Theta})} d\boldsymbol{\Theta} \\
&= \sum_{m=1}^{M} \int q(\boldsymbol{\theta}^m) \left(\log\frac{q(\boldsymbol{\theta}^m)}{p(\boldsymbol{\theta}^m)} - \log\frac{q(\boldsymbol{\theta}^m)}{p(\boldsymbol{\theta}^m|\mathbf{y}^m, \mathbf{X}^m)}\right) d\boldsymbol{\theta}^m \\
&\leq \sum_{m=1}^{M} KL(q(\boldsymbol{\theta}^m) \| p(\boldsymbol{\theta}^m)) \\
&= \sum_{m=1}^{M} \left(2n_{\text{RF}} \log\frac{\sigma_{\theta^m}}{\xi_m} + \frac{1}{2\sigma_\theta^2}\left(\|\boldsymbol{\theta}_*^m\|^2 + 2n_{\text{RF}}\xi_m^2\right) - n_{\text{RF}}\right). \quad (78)
\end{aligned}$$

Next, following the derivations in (60)–(62) yields for any m

$$\bar{\mathcal{L}}_{q_\Theta}^m \leq \sum_{t\in\mathcal{T}_m} \mathcal{L}(\boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}_t)\boldsymbol{\theta}_*^m; y_t) + \frac{T_m c\xi_m^2}{2}$$

the sum of which over m in conjunction with (78) results in

$$\begin{aligned}
&\sum_{\tau=1}^{T} l^{i_\tau}_{\tau|\tau-1} - \sum_{\tau=1}^{T} \mathcal{L}(\boldsymbol{\phi}_{\mathbf{v}}^{i_\tau\top}(\mathbf{x}_\tau)\boldsymbol{\theta}_*^{i_\tau}; y_\tau) \leq \quad (79) \\
&\sum_{m=1}^{M} \left(2n_{\text{RF}} \log\frac{\sigma_{\theta^m}}{\xi_m} + \frac{1}{2\sigma_\theta^2}\left(\|\boldsymbol{\theta}_*^m\|^2 + 2n_{\text{RF}}\xi_m^2\right) - n_{\text{RF}} + \frac{T_m c\xi_m^2}{2}\right).
\end{aligned}$$

It is evident that the RHS of (79) is a sum of M convex functions of ξ_m, each of which takes minimal value at $\xi_m^* = \sqrt{(2n_{\text{RF}}\sigma_{\theta^m}^2)/(2n_{\text{RF}} + T_m c\sigma_{\theta^m}^2)}$. Setting the RHS of (79) to its minimal value yields

$$\begin{aligned}
&\sum_{\tau=1}^{T} l^{i_\tau}_{\tau|\tau-1} - \sum_{\tau=1}^{T} \mathcal{L}(\boldsymbol{\phi}_{\mathbf{v}}^{i_\tau\top}(\mathbf{x}_\tau)\boldsymbol{\theta}_*^{i_\tau}; y_\tau) \\
&\leq \sum_{m=1}^{M} \left(\frac{\|\boldsymbol{\theta}_*^m\|^2}{2\sigma_{\theta^m}^2} + n_{\text{RF}} \log\left(1 + \frac{T_m c\sigma_{\theta^m}^2}{2n_{\text{RF}}}\right)\right) \\
&\leq \sum_{m=1}^{M} \left(\frac{\|\boldsymbol{\theta}_*^m\|^2}{2\sigma_{\theta^m}^2} + n_{\text{RF}} \log\left(1 + \frac{T_m c\sigma_{\theta^*}^2}{2n_{\text{RF}}}\right)\right) \quad (80)
\end{aligned}$$

where $\sigma_{\theta^*}^2 = \max_{m\in\mathcal{M}} \sigma_{\theta^m}^2$. Since $\log(\cdot)$ is a concave function, it follows with $T = \sum_{m=1}^M T_m$ that

$$\sum_{m=1}^{M} \log\left(1 + \frac{T_m c\sigma_{\theta^*}^2}{2n_{\text{RF}}}\right) \leq M \log\left(1 + \frac{Tc\sigma_{\theta^*}^2}{2n_{\text{RF}}}\right)$$

which, upon plugging into (80), finalizes the proof of Lemma 3.

9.5 Proof of Theorem 2.

For a given $\mathbf{i}_T$, expert m relies on batch data $\{\mathbf{x}_\tau, y_\tau, \tau \in \mathcal{T}_m\}$ in hindsight to learn the benchmark function $\hat{f}^m(\mathbf{x})$ in the RKHS and the RF-based one $\check{f}_*^m(\mathbf{x})$ with parameter vector $\boldsymbol{\theta}_*^m$. Following (64)–(68) in Sec. 9.2, it holds true with probability at least $1 - 2^8(\frac{\sigma_m}{\epsilon})^2 \exp\left(\frac{-n_{\text{RF}}\epsilon^2}{4d+8}\right)$ that

$$\sum_{\tau\in\mathcal{T}_m} \mathcal{L}\Big(\boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}_\tau)\boldsymbol{\theta}_*^m; y_\tau\Big) - \sum_{\tau\in\mathcal{T}_m} \mathcal{L}\Big(\hat{f}^m(\mathbf{x}_\tau); y_\tau\Big) \leq \epsilon L T_m C'$$

where $C' := \max_{m\in\mathcal{M}} \sum_{\tau\in\mathcal{T}_m} \sigma_{\theta^m}^2 |\hat{\alpha}_\tau^m|$. Summing the above inequality over $m \in \mathcal{M}$ leads to

$$\sum_{\tau=1}^{T} \mathcal{L}\Big(\boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}_\tau)\boldsymbol{\theta}_*^m; y_\tau\Big) - \sum_{\tau=1}^{T} \mathcal{L}\Big(\hat{f}^m(\mathbf{x}_\tau); y_\tau\Big) \leq \epsilon L T C' \quad (81)$$

holding true with probability at least $1 - 2^8(\frac{\sigma_*}{\epsilon})^2 \exp\left(\frac{-n_{\text{RF}}\epsilon^2}{4d+8}\right)$, where $\sigma_*^2 := \max_{m\in\mathcal{M}} \sigma_m^2$.

Meanwhile, adapting the result in (70) to expert m possessing data $\mathcal{D}_m$, yields the ensuing inequality concerning $\boldsymbol{\theta}_*^m$

$$\begin{aligned}
\|\boldsymbol{\theta}_*^m\|^2 &:= \left\|\sigma_{\theta^m}^2 \sum_{\tau\in\mathcal{T}_m} \hat{\alpha}_\tau^m \boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_\tau)\right\|^2 \quad (82) \\
&= \sigma_{\theta^m}^4 \sum_{\tau\in\mathcal{T}_m} \sum_{\tau'\in\mathcal{T}_m} \hat{\alpha}_\tau^m \hat{\alpha}_{\tau'}^m \boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}_\tau)\boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_{\tau'}) \leq (1+\epsilon)C'^2
\end{aligned}$$

which, in conjunction with Lemmas 2-3 and (81), readily proves Theorem 2.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their constructive feedback. We also gratefully acknowledge the support from NSF grants 1901134 and 2126052.

REFERENCES

[1] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.

[2] I. Kononenko, "Machine learning for medical diagnosis: history, state of the art and perspective," *Artif. Intel. in Medicine*, vol. 23, no. 1, pp. 89–109, 2001.

0162-8828 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Minnesota. Downloaded on July 26,2022 at 19:20:57 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3157197, IEEE Transactions on Pattern Analysis and Machine Intelligence

[3] K. Cutajar, M. Osborne, J. Cunningham, and M. Filippone, "Preconditioning kernel matrices," *Proc. Int. Conf. Mach. Learn.*, pp. 2529–2538, 2016.

[4] K. Wang, G. Pleiss, J. Gardner, S. Tyree, K. Q. Weinberger, and A. G. Wilson, "Exact Gaussian processes on a million data points," *Proc. Adv. Neural Inf. Process. Syst.*, pp. 14 648–14 659, 2019.

[5] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," *Proc. Adv. Neural Inf. Process. Syst.*, pp. 1257–1264, 2006.

[6] M. Titsias, "Variational learning of inducing variables in sparse Gaussian processes," *Proc. Int. Conf. Artif. Intel. and Stats.*, pp. 567–574, 2009.

[7] M. Lázaro-Gredilla, J. Quiñonero Candela, C. E. Rasmussen, and A. Figueiras-Vidal, "Sparse spectrum Gaussian process regression," *J. Mach. Learn. Res.*, vol. 11, no. Jun, pp. 1865–1881, 2010.

[8] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," *Proc. Adv. Neural Inf. Process. Syst.*, pp. 1177–1184, 2008.

[9] A. Gijsberts and G. Metta, "Real-time model learning using incremental sparse spectrum Gaussian process regression," *Neural Networks*, vol. 41, pp. 59–69, 2013.

[10] E. Hazan, "Introduction to online convex optimization," *Foundations and Trends® in Optimization*, vol. 2, no. 3-4, pp. 157–325, 2016.

[11] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning, and games*. Cambridge University press, 2006.

[12] C.-A. Cheng and B. Boots, "Incremental variational sparse Gaussian process regression," *Proc. Adv. Neural Inf. Process. Syst.*, pp. 4410–4418, 2016.

[13] T. D. Bui, C. Nguyen, and R. E. Turner, "Streaming sparse Gaussian process approximations," *Proc. Adv. Neural Inf. Process. Syst.*, 2017.

[14] S. M. Kakade, M. W. Seeger, and D. P. Foster, "Worst-case bounds for Gaussian process models," *Proc. Adv. Neural Inf. Process. Syst.*, pp. 619–626, 2006.

[15] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson, "Gpytorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration," *Proc. Adv. Neural Inf. Process. Syst.*, 2018.

[16] E. Gilboa, Y. Saatçi, and J. P. Cunningham, "Scaling multidimensional inference for structured Gaussian processes," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 37, no. 2, pp. 424–436, 2013.

[17] J. P. Cunningham, K. V. Shenoy, and M. Sahani, "Fast Gaussian process methods for point process intensity estimation," *Proc. Int. Conf. Mach. Learn.*, pp. 192–199, 2008.

[18] J. Quiñonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *J. Mach. Learn. Res.*, vol. 6, no. Dec, pp. 1939–1959, 2005.

[19] Y. Gal and R. Turner, "Improving the Gaussian process sparse spectrum approximation by representing uncertainty in frequency inputs," *Proc. Int. Conf. Mach. Learn.*, 2015.

[20] L. Csató and M. Opper, "Sparse on-line Gaussian processes," *Neural computation*, vol. 14, no. 3, pp. 641–668, 2002.

[21] M. A. Osborne, "Bayesian Gaussian processes for sequential prediction, optimisation and quadrature," Ph.D. dissertation, Oxford University, UK, 2010.

[22] T. N. Hoang, Q. M. Hoang, and B. K. H. Low, "A unifying framework of anytime sparse Gaussian process regression models with stochastic variational inference for big data." *Proc. Int. Conf. Mach. Learn.*, pp. 569–578, 2015.

[23] S. Stanton, W. Maddox, I. Delbridge, and A. G. Wilson, "Kernel interpolation for scalable online Gaussian processes," *Proc. Int. Conf. Artif. Intel. and Stats.*, pp. 3133–3141, 2021.

[24] H.-M. Kim, B. K. Mallick, and C. Holmes, "Analyzing nonstationary spatial data using piecewise Gaussian processes," *J. of the Amer. Stat. Assoc.*, vol. 100, no. 470, pp. 653–668, 2005.

[25] V. Tresp, "A Bayesian committee machine," *Neural Computation*, vol. 12, no. 11, pp. 2719–2741, 2000.

[26] M. P. Deisenroth and J. W. Ng, "Distributed Gaussian processes," in *Proc. of Intl. Conf. on Machine Learning*, 2015.

[27] C. E. Rasmussen and Z. Ghahramani, "Infinite mixtures of Gaussian process experts," *Proc. Adv. Neural Inf. Process. Syst.*, pp. 881–888, 2002.

[28] E. Meeds and S. Osindero, "An alternative infinite mixture of Gaussian process experts," *Proc. Adv. Neural Inf. Process. Syst.*, pp. 883–890, 2006.

[29] M. Trapp, R. Peharz, F. Pernkopf, and C. E. Rasmussen, "Deep structured mixtures of Gaussian processes," *Proc. Int. Conf. Artif. Intel. and Stats.*, pp. 2251–2261, 2020.

[30] C. A. Micchelli and M. Pontil, "Learning the kernel function via regularization," *J. Mach. Learn. Res.*, vol. 6, no. Jul, pp. 1099–1125, 2005.

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

[32] J. Lu, S. C. Hoi, J. Wang, P. Zhao, and Z.-Y. Liu, "Large scale online kernel learning," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1613–1655, 2016.

[33] R. Jin, S. C. Hoi, and T. Yang, "Online multiple kernel learning: Algorithms and mistake bounds," in *Proc. of Intl. Conf. on Algorithmic Learning Theory*, 2010, pp. 390–404.

[34] Y. Shen, T. Chen, and G. B. Giannakis, "Random feature-based online multi-kernel learning in environments with unknown dynamics," *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 773–808, 2019.

[35] N. Lawrence, "Probabilistic non-linear principal component analysis with Gaussian process latent variable models," *J. Mach. Learn. Res.*, vol. 6, no. Nov, pp. 1783–1816, 2005.

[36] N. D. Lawrence, "Learning for larger datasets with the Gaussian process latent variable model," *Proc. Int. Conf. Artif. Intel. and Stats.*, pp. 243–250, 2007.

[37] A. C. Damianou, M. K. Titsias, and N. D. Lawrence, "Variational inference for latent variables and uncertain inputs in Gaussian processes," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1425–1486, 2016.

[38] A. Yao, J. Gall, L. V. Gool, and R. Urtasun, "Learning probabilistic non-linear latent variable models for tracking complex activities," *Proc. Adv. Neural Inf. Process. Syst.*, pp. 1359–1367, 2011.

[39] Y. Wang, M. A. Brubaker, B. Chaib-draa, and R. Urtasun, "Bayesian filtering with online Gaussian process latent variable models." in *Uncertainty in Artif. Intel.*, 2014, pp. 849–857.

[40] X. Qin, P. Blomstedt, and S. Kaski, "Scalable Bayesian non-linear matrix completion," *arXiv preprint arXiv:1908.01009*, 2019.

[41] M. Ghashami, D. J. Perry, and J. Phillips, "Streaming kernel principal component analysis," *Proc. Int. Conf. Artif. Intel. and Stats.*, pp. 1365–1374, 2016.

[42] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol. 11, no. 2, pp. 443–482, 1999.

[43] A. Bellas, C. Bouveyron, M. Cottrell, and J. Lacaille, "Model-based clustering of high-dimensional data streams with online mixture of probabilistic PCA," *Advances in Data Analysis and Classification*, vol. 7, no. 3, pp. 281–300, 2013.

[44] Q. Lu, G. Karanikolas, Y. Shen, and G. B. Giannakis, "Ensemble Gaussian processes with spectral features for online interactive learning with scalability," *Proc. Int. Conf. Artif. Intel. and Stats.*, pp. 1910–1920, 2020.

[45] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Trans. Sig. Process.*, vol. 57, no. 3, pp. 1058–1067, 2008.

[46] N. Xu, K. H. Low, J. Chen, K. K. Lim, and E. B. Ozgul, "GP-localize: Persistent mobile robot localization using online sparse Gaussian process observation model," in *AAAI Conf. on Artif. Intel.*, 2014.

[47] S. M. Kakade and A. Y. Ng, "Online bounds for Bayesian algorithms," *Proc. Adv. Neural Inf. Process. Syst.*, pp. 641–648, 2005.

[48] C. K. Williams and D. Barber, "Bayesian classification with Gaussian processes," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 20, no. 12, pp. 1342–1351, 1998.

[49] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

[50] D. Duvenaud, J. Lloyd, R. Grosse, J. Tenenbaum, and G. Zoubin, "Structure discovery in nonparametric regression through compositional kernel search," *Proc. Int. Conf. Mach. Learn.*, pp. 1166–1174, 2013.

[51] H. Kim and Y. W. Teh, "Scaling up the automatic statistician: Scalable structure discovery using Gaussian processes," *Proc. Int. Conf. Artif. Intel. and Stats.*, pp. 575–584, 2018.

[52] G. Malkomes, C. Schaff, and R. Garnett, "Bayesian optimization for automated model selection," *Proc. Adv. Neural Inf. Process. Syst.*, 2016.

[53] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 42, no. 4, pp. 824–836, 2020.

[54] S. De Vito, E. Massera, M. Piga, L. Martinotto, and G. Di Francia, "On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario," *Sensors and Actuators B: Chemical*, vol. 129, no. 2, pp. 750–757, 2008.

[55] G. Rätsch, "Ensemble learning methods for classification," April 1998, diploma thesis (in german). [Online]. Available: http://www.first.gmd.de/ raetsch/diplom.ps.gz

[56] T. G. Dietterich, A. N. Jain, R. H. Lathrop, and T. Lozano-Perez, "A comparison of dynamic reposing and tangent distance for drug activity prediction," *Proc. Adv. Neural Inf. Process. Syst.*, pp. 216–223, 1994.

0162-8828 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Minnesota. Downloaded on July 26,2022 at 19:20:57 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3157197, IEEE Transactions on Pattern Analysis and Machine Intelligence

[57] V. G. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker, "Classification of radar returns from the ionosphere using neural networks," *Johns Hopkins APL Technical Digest*, vol. 10, no. 3, pp. 262–266, 1989.

[58] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 16, no. 5, pp. 550–554, 1994.

[59] C. M. Bishop and G. D. James, "Analysis of multiphase flows using dual-energy gamma densitometry and neural networks," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 327, no. 2-3, pp. 580–593, 1993.

[60] F. Kawala, A. Douzal-Chouakria, E. Gaussier, and E. Dimert, "Prédictions d'activité dans les réseaux sociaux en ligne," in *4ième conférence sur les modèles et l'analyse des réseaux : Approches mathématiques et informatiques*, France, Oct. 2013, p. 16.

[61] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[62] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (gpml) toolbox," *J. Mach. Learn. Res.*, vol. 11, no. Nov, pp. 3011–3015, 2010.

[63] D. Nguyen-Tuong, J. Peters, and M. Seeger, "Local Gaussian process regression for real time online model learning," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 21, 2008.

[64] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Intl. J. of Comput. Vis.*, vol. 77, no. 1-3, pp. 125–141, 2008.

[65] A. Levey and M. Lindenbaum, "Sequential Karhunen-Loeve basis extraction and its application to images," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1371–1374, 2000.

[66] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

Qin Lu (M'18) received her Ph.D. degree in electrical engineering from University of Connecticut (UConn) in 2018. Currently, she is a postdoctoral research associate at University of Minnesota, Twin Cities. Her research interests span the areas of machine learning, data science, and network science, with special focus on Bayesian inference, Bayesian optimization, and spatio-temporal inference over graphs. In the past, she has worked on statistical signal processing, multiple target tracking, and underwater acoustic communications. She was awarded Summer Fellowship and Doctoral Dissertation Fellowship at UConn. She was also a recipient of the Women of Innovation Award in Collegian Innovation and Leadership by Connecticut Technology Council in March, 2018.

Georgios V. Karanikolas (M'21) received his Diploma (valedictorian) in Electrical and Computer Engineering from the University of Patras, Greece, in 2014, his M.Sc. in Electrical Engineering in 2016 and his Ph.D. in Electrical Engineering in 2021, both from the University of Minnesota (UMN). He is currently a postdoctoral associate with the Dept. of Electrical and Computer Engineering at UMN. His research interests lie in the broad areas of machine learning, signal processing and network science.

Georgios B. Giannakis (F'97) received his Diploma in Electrical Engr. from the Ntl. Tech. Univ. of Athens, Greece, 1981. From 1982 to 1986 he was with the Univ. of Southern California (USC), where he received his MSc. in Electrical Engineering, 1983, MSc. in Mathematics, 1986, and Ph.D. in Electrical Engr., 1986. He was a faculty member with the University of Virginia from 1987 to 1998, and since 1999 he has been a professor with the Univ. of Minnesota, where he holds an ADC Endowed Chair, a University of Minnesota McKnight Presidential Chair in ECE, and serves as director of the Digital Technology Center. His general interests span the areas of statistical learning, signal processing, communications, and networking - subjects on which he has published more than 480 journal papers, 780 conference papers, 25 book chapters, two edited books and two research monographs. Current research focuses on Data Science, and Network Science with applications to the Internet of Things, and power networks with renewables. He is the (co-) inventor of 34 issued patents, and the (co-) recipient of 10 best journal paper awards from the IEEE Signal Processing (SP) and Communications Societies, including the G. Marconi Prize Paper Award in Wireless Communications. He also received the IEEE-SPS Norbert Wiener Society Award (2019); EURASIP's A. Papoulis Society Award (2020); Technical Achievement Awards from the IEEE-SPS (2000) and from EURASIP (2005); the IEEE ComSoc Education Award (2019); and the IEEE Fourier Technical Field Award (2015). He is a member of the Academia Europaea, and Fellow of the National Academy of Inventors, the European Academy of Sciences, IEEE and EURASIP. He has served the IEEE in a number of posts, including that of a Distinguished Lecturer for the IEEE-SPS.

0162-8828 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Minnesota. Downloaded on July 26,2022 at 19:20:57 UTC from IEEE Xplore. Restrictions apply.