Sequential Gaussian Processes for Online Learning of Nonstationary Functions

Michael Minyi Zhang
University of Hong Kong

Bianca Dumitrascu
University of Cambridge

Sinead A. Williamson
UT Austin

Barbara E. Engelhardt
Stanford University

Abstract

We propose a sequential Monte Carlo algorithm to fit infinite mixtures of GPs that capture non-stationary behavior while allowing for online, distributed inference. Our approach empirically improves performance over state-of-the-art methods for online GP estimation in the presence of non-stationarity in time-series data. To demonstrate the utility of our proposed online Gaussian process mixture-of-experts approach in applied settings, we show that we can successfully implement an optimization algorithm using online Gaussian process bandits.

1 Introduction

Data are often observed as streaming observations that arrive sequentially across time. To model streaming data, it is more efficient to update model parameters as new observations arrive than to refit the model from scratch with the new observations appended onto existing data. Gaussian processes (GPs) are a convenient distribution on real-valued functions because, when evaluated at a fixed set of inputs, they have a multivariate normal distribution and hence allow closed-form posterior inference and prediction when used for regression.

From a statistical perspective, a typical GP regression model infers only stationary functions, meaning that properties of the function are constant across all input values. While there are covariance kernels that explicitly capture non-stationary effects in GP regression, they pose greater computational challenges than a stationary kernel as they often require calculating intractable integrals.

Mixture-of-experts GP models have been used to model non-stationary functions by fitting independent GPs to different segments of the input space. In particular, the IS-MOE approach fits mixtures of GP experts in a distributed manner using importance sampling (Zhang and Williamson, 2019); however, IS-MOE is not an online algorithm. Conversely, sparse online GPs are a state-of-the-art method for online GP estimation, but the estimated functions are constrained to be stationary.

We introduce a sequential Monte Carlo (SMC) algorithm to fit infinite mixtures of GPs. SMC samplers can be adapted to allow real-time updates to the model parameters, and are trivially parallelizable. We show a connection with online GPs to multi-armed bandits for optimization and demonstrate that our method can obtain superior performance compared to other GP-bandit optimization techniques.

2 Sequential Gaussian Processes for Online Learning

We assume that our data is generated from a Gaussian process mixture, similar to previous mixture-of-expert models for Gaussian processes. This hierarchical model allows for greater flexibility in modeling functions, at the cost of more difficult inference for which we propose a distributable

2022 NeurIPS Workshop on Gaussian Processes, Spatiotemporal Modeling, and Decision-making Systems.

solution in the next section. Our approach adopts the following generative model:

$$\begin{aligned} \mathbf{x}_i \sim \mathcal{T}(\boldsymbol{\mu}_{z_i}, \boldsymbol{\Psi}_{z_i}, \nu_{z_i}), \quad \alpha \sim \text{Gamma}(a_0, b_0), z_i|\alpha \sim \text{CRP}(\alpha), \\ (\theta_k, \sigma_k^2) \sim \log\mathcal{N}(\mathbf{m}_0, s_0^2\mathbf{I}), \mathbf{y}_k|\mathbf{X}_k, \theta_k, \sigma_k^2 \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\theta_k} + \sigma_k^2\mathbf{I}), \end{aligned} \tag{1}$$

where $(\mathbf{X}_k, \mathbf{y}_k) = (\mathbf{x}_i, y_i : z_i = k)$ represent the data associated with the mixture k. We assume that the inputs are distributed according to a Dirichlet process mixture of normal-inverse Wishart distributions, and we marginalize out the parameter locations from the inputs. The outputs are then assumed to be generated by independent GPs, given the mixture indicator. The GP parameters, $\boldsymbol{\theta}_k = [\theta_k, \sigma_k^2]$, are assumed to be log-normally distributed, and we update the state of $\boldsymbol{\theta}_k$ using the elliptical slice sampler (Murray et al., 2010). We assign the ith input sequentially to clusters according to the Chinese restaurant prior and the mixture locations marginalized out:

$$P(z_i = k|\alpha, \mathbf{X}_k) \propto \begin{cases} N_k' \cdot \mathcal{T}(\boldsymbol{\mu}_k', \boldsymbol{\Psi}_k', \nu_k') & k \in K^+. \\ \alpha \cdot \mathcal{T}(\boldsymbol{\mu}_0, \boldsymbol{\Psi}_0, \nu_0) & \text{o.w.} \end{cases} \tag{2}$$

where $(\boldsymbol{\mu}_k', \boldsymbol{\Psi}_k', \nu_k')$ are the mean, covariance, and degrees-of-freedom parameters of the multivariate-t likelihood for observation i's sequential assignment to mixture k, where K^+ refers to the previously occupied clusters, $\{k : N_k' > 0\}$.

$$\begin{aligned} \boldsymbol{\mu}_k' = \frac{\lambda_0\boldsymbol{\mu}_0 + N_k'\bar{\mathbf{x}}_k}{\lambda_k'}, \bar{\mathbf{x}}_k' = \frac{\sum_{i':(z_{i'}=k, i'<i)} \mathbf{x}_{i'}}{N_k'}, N_k' = \sum_{i'=1}^{i-1} I(z_{i'} = k), \\ \lambda_k' = \lambda_0 + N_k', \nu_k' = \nu_0 + N_k' - D + 1, \boldsymbol{\Psi}_k' = \frac{\lambda_k' + 1}{\lambda_k'\nu_k'}\left(\boldsymbol{\Psi}_0 + \mathbf{S}_k' + \mathbf{S}_{\bar{\mathbf{x}}_k}'\right) \\ \mathbf{S}_k' = \sum_{i':(z_{i'}=k, i'<i)} \left(\mathbf{x}_{i'} - \bar{\mathbf{x}}_k'\right)\left(\mathbf{x}_{i'} - \bar{\mathbf{x}}_k'\right)^T, \mathbf{S}_{\bar{\mathbf{x}}_k}' = \frac{\lambda_0 N_k'}{\lambda_k'}\left(\bar{\mathbf{x}}_k' - \boldsymbol{\mu}_0\right)\left(\bar{\mathbf{x}}_k' - \boldsymbol{\mu}_0\right)^T. \end{aligned} \tag{3}$$

We use the $(\cdot)'$ notation to indicate that the summary statistics are conditioned only on observations $i' = 1, \ldots, i-1$. We also place a Gamma prior on the DP concentration parameter, α, which allows us to easily sample its full conditional up to observation i with a variable augmentation scheme (Escobar and West, 1995):

$$\begin{aligned} \rho|\alpha \sim \text{Beta}(\alpha + 1, i), K = |\{k : N_k > 0\}|, \frac{\pi_\alpha}{1 - \pi_\alpha} = \frac{a_0 + K - 1}{N(b_0 - \log\rho)} \\ \alpha|\mathbf{z}_{1:i}, \pi_\alpha, \rho = (1 - \pi_\alpha) \cdot \text{Gamma}(\alpha_0 + K - 1, b_0 - \log\rho) + \pi_\alpha \cdot \text{Gamma}(\alpha_0 + K, b_0 - \log\rho). \end{aligned} \tag{4}$$

2.1 SMC for Online GP-MOE

In an SMC setting with $j = 1, \ldots, J$ particles, we first propagate the particles $(\mathbf{z}^{(j)}, \boldsymbol{\theta}^{(j)}, \alpha^{(j)})$ from $i-1$ to i and fit a GP product-of-experts model. Then we calculate the particle weights. At the initial time, $i = 1$, on particle j is:

$$w_1^{(j)} \propto P(y_1|z_1^{(j)}, \mathbf{x}_1, \boldsymbol{\theta}^{(j)})P(\mathbf{x}_1|z_1^{(j)}, \alpha^{(j)}). \tag{5}$$

For $i > 1$, the particle weight in Equation 6 can be decomposed as the product of:

1. The previous weight, $w_{i-1}^{(j)}$,
2. The ratio of the model's likelihood up to observation i over the likelihood up to observation $i-1$ (Svensson et al., 2015),
3. The particle weight of $z_i^{(j)}$ for the Dirichlet process mixture model (Carvalho et al., 2010).

The GP term of the particle weight from Svensson et al. (2015) in this setting simplifies to the ratio of the new likelihood (including observation i) and the old likelihood (excluding observation i) of the mixture z_i. We can store the old likelihood in memory from the last time we updated the particle weights, so the only computationally intensive step is computing the new likelihood for mixture z_i.

After calculating the particle weights, we calculate the effective sample size, $N_{\text{eff}} = 1/\sum_{j=1}^J \left(w_i^{(j)}\right)^2$. If the effective number of samples drops below a certain threshold (typically

$J/2$), then we resample the particles with probability $w_i^{(1)}, \ldots, w_i^{(J)}$ to avoid the particle degeneracy problem. The details for updating the particles are in Algorithm 1.

To calculate the predictive posterior distribution of the GP-MOE for test data $\mathbf{x}_*$, we calculate the predictive mean and variance on each individual particle, averaged over the mixture assignment for the test data. Then, we average the predictive distribution on each particle, weighted by $w_i^{(1)}, \ldots, w_i^{(J)}$.

Algorithm 1: Online GP-MOE

Input: New observation, $(\mathbf{x}_i, y_i)$.

`/* Particle propagation. */`

for $j = 1, \ldots, J$ *in parallel* **do**

Sample $z_i^{(j)}$ from $P(z_i^{(j)}|\alpha^{(j)}, \mathbf{X}_{1:i-1})$, in Eq. 2

Sample $\boldsymbol{\theta}_{z_i}^{(j)}$ using the elliptical slice sampler.

Sample $\alpha^{(j)}$ from the full conditional $P(\alpha^{(j)}|\mathbf{z}_{1:i})$ in Eq. 4.

Update particle weight:

$$w_i^{(j)} = w_{i-1}^{(j)} \cdot \frac{P\left(\mathbf{y}_{z_i^{(j)}} \middle| \mathbf{X}_{z_i^{(j)}}, , \boldsymbol{\theta}_{z_i^{(j)}}^{(j)}\right)}{P\left(\mathbf{y}'_{z_i^{(j)}} \middle| \mathbf{X}'_{z_i^{(j)}}, \boldsymbol{\theta}_{z_i^{(j)}}^{(j)}\right)} \cdot P(\mathbf{x}_i|z_i^{(j)}, \alpha^{(j)}) \tag{6}$$

where

$$(\mathbf{X}'_k, \mathbf{y}'_k) = (\mathbf{x}_{i'}, y_{i'}, i' : (z_{i'} = k, i' < i)), (\mathbf{X}_k, \mathbf{y}_k) = (\mathbf{x}_i, y_i, i : z_i = k) \tag{7}$$

Normalize weights, $w_i^{(j)} := w_i^{(j)} / \sum_{j=1}^J w_i^{(j)}$.

`/* Particle resampling. */`

if $N_{\mathit{eff}} < J/2$ **then**

Resample particles $\left(\mathbf{z}_{1:i}^{(\mathbf{j}^*)}, \boldsymbol{\theta}^{(\mathbf{j}^*)}, \alpha^{(\mathbf{j}^*)}\right)$ from $\mathbf{j}^* \sim$ Multinomial $\left(J, w_i^{(1)}, \ldots, w_i^{(J)}\right)$.

Set $w_i^{(j)} := 1/J$ for $j = 1, \ldots, J$.

Output: Particle weights $\left(w_i^{(1)}, \ldots, w_i^{(J)}\right)$ and particles $\left(\mathbf{z}_{1:t}^{(1:J)}, \boldsymbol{\theta}^{(1:J)}, \alpha^{(1:J)}\right)$.

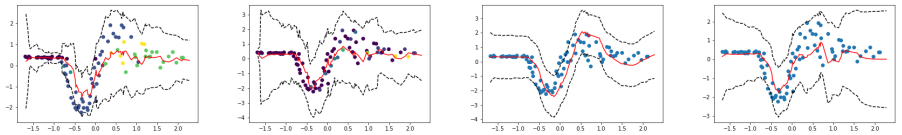
3 Empirical Analyses for Online GP-MOE

To demonstrate the ability of our algorithm to fit streaming non-stationary GPs, we apply our online GP-MOE to a collection of empirical time-series datasets that exhibit non-stationary behavior. In our comparisons, we look at the following datasets: 1.) An accelerometer measurement of a motorcycle crash. 2.) The price of Brent crude oil. 3.) The annual water level of the Nile river data. 4.) The exchange rate between the Euro and the US Dollar. 5.) The annual carbon dioxide output in Canada.

We compare our method against three alternative approaches: a product-of-experts model (POE), which is a special case of our algorithm with only one particle, a sparse online GP method using the Woodbury identity and structured kernel interpolation (Stanton et al., 2021, WISKI), and an online sparse variational GP method (Bui et al., 2017, OSVGP)[1].

In these experiments, we set the number of inducing points to be 50 for each of the sparse methods. We evaluate our method when J is equal to 1 (equivalent to a POE), 100, and 500 particles. Our choice of kernel for each of these methods is the radial basis function (RBF). The OSVGP method requires a fixed number of optimization iterations to estimate the variational parameters, and the results are highly sensitive to this setting. To make the model settings comparable to the GP-MOE settings, we also set the optimizer iterations to 1, 100, and 500. In the GP-MOE model, we distribute the inference of each particle over 16 cores. In this setting, we run an online prediction experiment where we initialize the model using the first observation in the time-series data set. Then, we sequentially predict the subsequent observation and update the model with the next data point.

[1] Our code is available at `https://github.com/michaelzhang01/GPMOE`.



(a) GP-MOE (b) POE (c) WISKI (d) OSVGP

Figure 1: Online posterior predictive mean (plotted with solid red lines) and 95% credible intervals (plotted with dashed black lines) for the motorcycle dataset.The color of the data points in these figures for GP-MOE and POE represent the mixture assignment of that observation for the particle with the highest weight. $N = 94$.

	Motorcycle	Nile	Brent	Canada	EUR-USD
GP-MOE 500	**-63.686 (2.370)**	-144.397 (2.447)	**266.563 (45.322)**	**305.852 (14.763)**	-4585.758 (20.097)
GP-MOE 100	-72.157 (3.841)	-147.531 (4.369)	120.428 (73.548)	261.257 (37.422)	-4539.573 (42.731)
POE	-114.665 (11.187)	-142.832 (1.513)	-456.628 (17.927)	-114.958 (12.374)	-4538.173 (25.029)
WISKI	-112.467 (0.000)	-127.998 (0.000)	-800.852 (0.000)	-152.242 (0.000)	**-4482.185 (0.000)**
OSVGP 500	-99.523 (3.019)	**-127.289 (0.157)**	-1250.734 (142.780)	43.021 (5.292)	-4766.513 (43.271)
OSVGP 100	-125.862 (0.306)	-138.537 (0.057)	-731.435 (156.499)	-230.525 (0.893)	-4494.476 (0.244)
OSVGP 1	-135.776 (0.100)	-145.048 (0.116)	-1438.428 (3.493)	-313.022 (0.296)	-4676.530 (5.742)

Table 1: Online predictive log likelihood over five trials. One standard error reported in parentheses.

Our method generally performs better than the competing online GP methods. We broadly observe that the GP-MOE performs better than POE because we can integrate over the space of partitions and, thus, we will better capture the predictive uncertainty. However, WISKI is undoubtedly the fastest method as the computational complexity is constant with respect to the number of observations. But because these data sets exhibit non-stationarity, WISKI is not able to handle changes in the kernel behavior (like heteroscedasticity, for example) and therefore performs poorly in terms of MSE and log likelihood in these experiments.

For the motorcycle, Brent, and Canadian carbon dioxide datasets, the GP-MOE performs the best in terms of predictive mean squared error and log likelihood. In the Nile river data set, the OSVGP performs the best in terms of online predictive log likelihood. This could be because the only non-stationary component of the Nile river data set is the mean value, which we assume to be constant at all values of $\mathbf{x}$ in GP-MOE. The OSVGP and WISKI obtain the best MSE and log likelihood results on the EUR-USD data sets as well, which exhibits only time varying noise, zero mean, and stationary length-scale for the entire duration of the time-series data. Here, OSVGP and WISKI produce wider noise estimates than GP-MOE.

4 Conclusion and Future Directions

In this paper, we introduced an online inference algorithm for fitting mixtures of Gaussian processes that can perform online estimation of non-stationary functions. For future work, we are interested in applying our online mixture of expert approach for modeling patient vital signs in a hospital setting. We will to extend our approach to multi-output GP models, and implement kernel functions customized for health care scenarios. By combining fast inference with flexible modeling, our approach will have a profound impact in real-time monitoring and decision-making in patient health.

	Motorcycle	Nile	Brent	Canada	EUR-USD
GP-MOE 500	**0.389 (0.007)**	**0.722 (0.003)**	**0.049 (0.006)**	0.019 (0.002)	1.010 (0.002)
GP-MOE 100	0.417 (0.019)	0.740 (0.005)	0.061 (0.008)	**0.018 (0.002)**	1.006 (0.002)
POE	0.479 (0.038)	0.807 (0.017)	0.123 (0.007)	0.102 (0.019)	1.016 (0.002)
WISKI	0.631 (0.000)	0.767 (0.000)	0.177 (0.000)	0.048 (0.000)	1.007 (0.000)
OSVGP 500	0.413 (0.028)	0.765 (0.003)	0.922 (0.047)	0.028 (0.001)	1.036 (0.013)
OSVGP 100	0.802 (0.004)	0.852 (0.001)	0.444 (0.050)	0.366 (0.004)	**1.003 (0.000)**
OSVGP 1	0.986 (0.001)	0.934 (0.001)	0.916 (0.003)	0.929 (0.005)	1.017 (0.002)

Table 2: Online predictive mean squared error over five trials. One standard error reported in parentheses.

References

Bui, T. D., Nguyen, C., and Turner, R. E. (2017). Streaming sparse Gaussian process approximations. In *Advances in Neural Information Processing Systems*, pages 3299–3307.

Carvalho, C. M., Lopes, H. F., Polson, N. G., and Taddy, M. A. (2010). Particle learning for general mixtures. *Bayesian Analysis*, 5(4):709–740.

Cheng, L.-F., Dumitrascu, B., Darnell, G., Chivers, C., Draugelis, M., Li, K., and Engelhardt, B. E. (2020). Sparse multi-output Gaussian processes for online medical time series prediction. *BMC Medical Informatics and Decision Making*, 20(1):1–23.

Escobar, M. D. and West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588.

Futoma, J., Hariharan, S., and Heller, K. (2017a). Learning to detect sepsis with a multitask Gaussian process RNN classifier. In *International Conference on Machine Learning*, pages 1174–1182. JMLR.

Futoma, J., Hariharan, S., Sendak, M., Brajer, N., Clement, M., Bedoya, A., O'Brien, C., and Heller, K. (2017b). An improved multi-output Gaussian process RNN with real-time validation for early sepsis detection. *arXiv preprint arXiv:1708.05894*.

Minsker, S., Srivastava, S., Lin, L., and Dunson, D. B. (2014). Scalable and robust Bayesian inference via the median posterior. In *International Conference on Machine Learning*, pages 1656–1664.

Murray, I., Adams, R., and MacKay, D. (2010). Elliptical slice sampling. In *International Conference on Artificial Intelligence and Statistics*, pages 541–548. JMLR.

Srivastava, S., Cevher, V., Dinh, Q., and Dunson, D. B. (2015). WASP: Scalable Bayes via barycenters of subset posteriors. In *Artificial Intelligence and Statistics*, pages 912–920.

Stanton, S., Maddox, W., Delbridge, I., and Wilson, A. G. (2021). Kernel interpolation for scalable online Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 3133–3141.

Svensson, A., Dahlin, J., and Schön, T. B. (2015). Marginalizing Gaussian process hyperparameters using sequential Monte Carlo. In *International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, pages 477–480. IEEE.

Zhang, M. M. and Williamson, S. A. (2019). Embarrassingly parallel inference for Gaussian processes. *Journal of Machine Learning Research*, 20(169):1–26.