Appendix

Anonymous Author(s)
Affiliation
Address
email

Note: In this supplementary material, we refer to equations and figures in the main paper.

1 Proofs

Proposition 1. *Let $f^*_{parent}|X_{parent}, Y_{parent} \sim \mathcal{GP}(\mu, k)$ be the full GP prior to splitting and let $\mathbf{x}^*$ be a test data point. The child GPs $f_i|X_i, Y_i \sim \mathcal{GP}(\mu_i, k)$, $i = 1, 2$ from the first split have the property that, prior to being updated with any new observations, $f^*|\mathbf{x}^* = f^*_{parent}|X_{parent}, Y_{parent}, \mathbf{x}^*$. That is, the predictive distribution is preserved by the splitting procedure.*

Proof. We consider the case of a single local model being split into two new local models. The posterior mean at an input $\mathbf{x}^*$ is given by

$$
\begin{aligned}
f^*|\mathbf{x}^* &= S^{-1}k(\mathbf{c}_1, \mathbf{x}^*)f^*_1|X_1, Y_1, \mathbf{x}^* + S^{-1}k(\mathbf{c}_2, \mathbf{x}^*)f^*_2|X_2, Y_2, \mathbf{x}^* && \text{by definition in (2)} \\
&= \alpha f^*_1|X_1, Y_1, \mathbf{x}^* + (1-\alpha)f^*_2|X_2, Y_2, \mathbf{x}^*, \quad \alpha = S^{-1}k(\mathbf{c}_1, \mathbf{x}^*) && \text{by (3.3)} \\
&= \alpha f^*_1|\mathbf{x}^* + (1-\alpha)f^*_2|\mathbf{x}^*, \ \alpha = S^{-1}k(\mathbf{c}_1, \mathbf{x}^*) \\
&\text{by assumption that no additional data has been observed since the split} \\
&= \alpha f^*_{parent}|X_{parent}, Y_{parent}, \mathbf{x}^* + (1-\alpha)f^*_{parent}|X_{parent}, Y_{parent}, \mathbf{x}^* && \text{by (1)} \\
&= f^*_{parent}|X_{parent}, Y_{parent}, \mathbf{x}^*.
\end{aligned}
$$

□

Proposition 2. *Suppose k is a kernel function and $f_i|X_i, Y_i \sim \mathcal{GP}(\mu_i, k)$, $i = 1, ..., C$. Then the random field given by*

$$f^*|\mathbf{x}^* = S^{-1}\sum_{i=1}^{C} k(\mathbf{c}_i, \mathbf{x}^*)\, f^*_i|X_i, Y_i, \mathbf{x}^*$$

is mean square continuous in the input space if and only if the kernel function, k, is continuous. Under the same condition, the mean prediction $\mathbb{E}[\, f^*|\mathbf{x}^*\,]$ is also continuous in the input space.

Proof. A common result on random fields gives that the random field $f^*|\mathbf{x}^*$ is mean square continuous if and only if its expectation and covariance functions are continuous; see Hristopulos [2020], Theorem 5.2, for example. It then suffices to show that $\mathbb{E}[\, f^*|\mathbf{x}^*\,]$ is continuous.

$$
\begin{aligned}
\mathbb{E}[\, f^*|\mathbf{x}^*\,] &= \mathbb{E}\left[\, S^{-1}\sum_{i=1}^{C} k(\mathbf{c}_i, \mathbf{x}^*)\, f^*_i \,\middle|\, X_i, Y_i, \mathbf{x}^* \,\right] && (1) \\
&= S^{-1}\sum_{i=1}^{C} k(\mathbf{c}_i, \mathbf{x}^*)\mathbb{E}[\, f^*_i|X_i, Y_i, \mathbf{x}^*\,] && (2)
\end{aligned}
$$

Recall that the predictive mean $\mathbb{E}[\, f^*_i|X_i, Y_i, \mathbf{x}^*\,]$ is a linear function of $\mathbf{x}^*$, and therefore continuous. Note from (3.3) that S^{-1} is continuous if and only if k is continuous. In this case, $\mathbb{E}[\, f^*|\mathbf{x}^*\,]$ is continuous, and hence $f^*|\mathbf{x}^*$ is mean square continuous. □

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

2 The splitting GP algorithm

Here we specify three algorithms which describe the main operations of the splitting GP model: splitting a GP, updating the model, and computing the predictive mean. We make use of the much of the same notation defined in the main paper.

From a computational perspective, the necessary components to construct a GP model are the training inputs and training response, X_i and Y_i, respectively. For convenience of notation, the following algorithms are thus described in terms of matrix operations on these variables. We will use the triple $(X_i, Y_i, \mathbf{c}_i)$ to characterize the i^{th} child GP, where $\mathbf{c}_i$ is the center of the GP as defined in the main paper. The entirety of the splitting GP model is itself given as a triple $(\mathcal{A}, k_{\boldsymbol{\theta}}, m)$, where $\mathcal{A} = \{(X_i, Y_i, \mathbf{c}_i) : i = 1, \ldots, C\}$ is a set of the child GPs, $k_{\boldsymbol{\theta}}$ is the kernel function with hyperparameter $\boldsymbol{\theta}$, and m is the splitting limit of the splitting GP model. *The following pseudo-code makes use of explicit `for` loops for clarity, but please note that our implementation (see the supplementary material) makes use of vectorized versions of these algorithms for efficiency.*

We first specify the splitting algorithm, which is used to divide a GP into two smaller child GPs. To keep the pseudo-code as general as possible, we define the function `PrincipalDirection`(X) as one which computes the first principal component vector of a matrix X.

Algorithm 1: `Split`$((X_i, Y_i, \mathbf{c}_i))$

$\hat{X}_1, \hat{X}_2 \leftarrow [\quad], [\quad]$;
$\hat{Y}_1, \hat{Y}_2 \leftarrow [\quad], [\quad]$;
$\hat{n}_1, \hat{n}_2 \leftarrow 0, 0$;
$\mathbf{v} \leftarrow$ `PrincipalDirection`(X_i);
for $j \leftarrow 1$ **to** n_i **do**
  $I \leftarrow \mathbf{v}^T (\mathbf{x}_{ij} - \mathbf{c}_i)$;
  if $I > 0$ **then**
    $\hat{X}_1 \leftarrow \left[\hat{X}_1^T \quad \mathbf{x}_{ij}\right]^T$;
    $\hat{Y}_1 \leftarrow \left[\hat{Y}_1^T \quad y_{ij}\right]^T$;
    $\hat{n}_1 \leftarrow \hat{n}_1 + 1$;
  else
    $\hat{X}_2 \leftarrow \left[\hat{X}_2^T \quad \mathbf{x}_{ij}\right]^T$;
    $\hat{Y}_2 \leftarrow \left[\hat{Y}_2^T \quad y_{ij}\right]^T$;
    $\hat{n}_2 \leftarrow \hat{n}_2 + 1$;
  end
end
$\hat{\mathbf{c}}_1 \leftarrow \hat{n}_1^{-1} \sum_{j=1}^{\hat{n}_1} \hat{\mathbf{x}}_{1j}$;
$\hat{\mathbf{c}}_2 \leftarrow \hat{n}_2^{-1} \sum_{j=1}^{\hat{n}_2} \hat{\mathbf{x}}_{2j}$;
Result: $\left(\hat{X}_1, \hat{Y}_1, \hat{\mathbf{c}}_1\right), \left(\hat{X}_2, \hat{Y}_2, \hat{\mathbf{c}}_2\right)$

Using `Split`, we can now define the algorithm `Update` for updating the splitting GP model given a new data point $(\mathbf{x}, y)$. We use the shorthand `Train` to mean the standard fitting of a GP model with

the hyperparameter $\boldsymbol{\theta}$ to data by means of maximizing the log marginal likelihood [Rasmussen and Williams, 2005].

Algorithm 2: `Update`$((\mathcal{A}, k_{\boldsymbol{\theta}}, m), (\mathbf{x}, y))$

if $C = 0$ **then**
  $X_1 \leftarrow \mathbf{x}^T$;
  $Y_1 \leftarrow y^T$;
  $\mathbf{c}_1 \leftarrow \mathbf{x}$;
  $\mathcal{A} \leftarrow [(X_1, Y_1, \mathbf{c}_1)]$;
  $C \leftarrow 1$;
else
  $I \leftarrow \underset{i=1,\ldots,C}{argmax}\ k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{c}_i)$;
  $X_I \leftarrow \begin{bmatrix} X_I^T & \mathbf{x} \end{bmatrix}^T$;
  $Y_I \leftarrow \begin{bmatrix} Y_I^T & y \end{bmatrix}^T$;
  $n_I \leftarrow n_I + 1$;
  $\mathbf{c}_I \leftarrow n_I^{-1} \sum_{j=1}^{n_I} \mathbf{x}_{Ij}$;
  if $n_I > m$ **then**
    $(X_I, Y_I, \mathbf{c}_I), (X_{C+1}, Y_{C+1}, \mathbf{c}_{C+1}) \leftarrow$ `Split` $((X_I, Y_I, \mathbf{c}_I))$;
    $\mathcal{A} \leftarrow \{(X_i, Y_i, \mathbf{c}_i) : i = 1, \ldots, C+1\}$;
    $C \leftarrow C + 1$;
  end
end
`Train` $(\boldsymbol{\theta}, \mathcal{A})$;
Result: $(\mathcal{A}, k_{\boldsymbol{\theta}}, m)$

Finally, the algorithm for computing the mean prediction of response at the input $\mathbf{x}$ follows. Here we use the abbreviated `Mean` function to give the posterior mean for a child GP $(X_i, Y_i, \mathbf{c}_i)$. The posterior mean for each child GP is computed in closed form using linear algebra; see [Rasmussen and Williams, 2005].

Algorithm 3: `Predict`$((\mathcal{A}, k_{\boldsymbol{\theta}}, m), \mathbf{x})$

for $i \leftarrow 1$ **to** C **do**
  $s_i \leftarrow k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{c}_i)$;
  $E_i \leftarrow$ `Mean`$((X_i, Y_i, \mathbf{c}_i), \mathbf{x})$;
end
$S \leftarrow \sum_{i=1}^{C} s_i$;
$\hat{y} \leftarrow S^{-1} \sum_{i=1}^{C} s_i E_i$;
Result: $\hat{y}$

3 Hyperparameter tuning for the local GP model

In the experiment with the synthetic data set, we initially performed a grid search on the parameter w_{gen} of the local GP model [Nguyen-Tuong et al., 2009] from .9 to .1 at increments of .05, and obtained the best results for $w_{gen} = .1$. However, the resulting MSE was much higher than the other models considered. In the interest of fair comparison, we conducted a second grid search ranging from .1 to 10^{-3} by increments of 5×10^{-4}, and the best parameter was found to be 10^{-3}, for which we show the results in Fig. 3.

We chose not to continue lowering w_{gen} further, since the local GP model's MSE may be expected to decrease monotonically with w_{gen}. The parameter w_{gen} defines a *similarity threshold*, such that if a new observation is sufficiently dissimilar from existing local models (i.e. $k(\mathbf{x}^*, \mathbf{c}_i) \leq w_{gen}, \forall i = 1, ..., C$), a new local model will be created. If $w_{gen} = 0$, then only one local GP will be created, so that the model reduces to a full GP. This behavior can be seen in Fig. 5, where we performed a grid search on w_{gen} ranging as low as 10^{-10}. For values less than 10^{-8}, limitations to floating point

precision caused the local GP model's prediction procedure to become numerically unstable, so Fig. 5 does not include these parameter values.

References

Dionissios T. Hristopulos. Geometric Properties of Random Fields. In *Random Fields for Spatial Data Modeling: A Primer for Scientists and Engineers*, Advances in Geographic Information Science, pages 173–244. Springer Netherlands, 2020. ISBN 978-94-024-1918-4. doi: 10.1007/978-94-024-1918-4_5.

Duy Nguyen-Tuong, Jan Peters, and Matthias Seeger. Local Gaussian Process Regression for Real Time Online Model Learning. *Advances in Neural Information ProcessingSystems*, pages 1193–1200, 2009.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.