Analytical Stacked Gaussian Process Model

Kareem Abdelfatah*, Junshu Bao**, Gabriel Terejanu*

* Dept. of Computer Science & Engineering
University of South Carolina
{Krabea,Terejanu} @email.sc.edu

** Dept. of Statistics
University of South Carolina
bao3@email.sc.edu

Abstract

A probabilistic model is proposed by stacking a set of independently trained Gaussian processes to obtain prediction of quantities of interests that require composition of functions. Analytical derivations are provided for first and second-order moments of the stacked Gaussian process using RBF and polynomial kernels. The StackedGP model can be extended to any number of layers and nodes per layer, and it provides flexibility in kernel selection for each node. The proposed nonparametric stacked model is validated using different synthetic datasets and its performance is measured in two real-world applications.

1 Introduction

A stacked Gaussian process (StackedGP) is proposed to obtain analytical expectations of quantities of interests that require composition of functions. This type of problems are common in geostatistics and spatial predictions where spatial interpolations are required to determine external variables at the location of interest before mapping them into quantities of interest. Usually, the mapping is done using purely phenomenological models on data collected in the lab rather than in the field. As a result, there is a compound effect of uncertainties coming from interpolation errors and model errors that need to be quantified and exposed to the quantities of interest. One such example is estimating the geographical incidence of aflatoxin production without any spatial observations [10]. This require building models that map environmental factors such as temperature to aflatoxin production using data collected in wet-labs. These models are then driven by interpolation models developed using spatial observations of temperature to finally determine aflatoxin production at locations of interest. Both aflatoxin production and environmental data can be modeled using independently trained Gaussian processes (GP). The resulted probabilistic model is a stacked Gaussian process [10], where the environmental GP governs the input space of the aflatoxin GP. In Ref. [10], Monte Carlo sampling was used to propagate the uncertainty through the stack model and estimate the mean and variance of aflatoxin production.

In Ref. [4], the authors introduced the exact moments for the predictive mean and variance of a single GP using square exponential kernel under the assumption of a Gaussian input distribution. In this paper, the proposed StackedGP model extends and unifies the work in Refs. [4, 10]. StackedGP can be extended to any number of layers with any number of nodes within each layer, and for certain kernels as described in this paper, analytical moments can be derived for the output of any node in the network. Since the GP nodes are independently trained using different datasets, the running time of the StackedGP grows linearly with the number of nodes and can be speed up through embarrassing parallel training of GPs.

In geostatistics, the densely sampled secondary responses are usually used to enhance the prediction of the primary variables [9]. Several examples can illustrate the idea such as uranium spill accident [13] and predicting cadmium in Swiss Jura [9, 20]. In Ref. [20], the authors built a Gaussian Process Regression Network to model the correlations between multiple outputs such as primary and secondary re-
responses. The outputs are given by weighted linearly combinations of latent functions where GP priors are defined over the weights, unlike to similar studies [3,13] where the weights are considered fixed. In this case, while not designed to capture the correlations of response variables, StackedGP models can be constructed by stacking GPs for predicting intermediate secondary responses that govern the input space of GPs used to predict primary responses. This hierarchical framework outperforms other methods as described in the numerical results section.

A model carrying the same name was introduced in Ref. [11], where a stacked Gaussian process was proposed to model pedestrian and public transit flows in urban areas. The model proposed in Ref. [11] is capable of capturing shared common causes using a joint Bayesian inference for multiple tasks. In our work the inference is performed independent for each GP node and then the uncertainty is propagated analytically through the network. In the next section, we discuss the StackedGP model for radial basis function (RBF) and polynomial kernels. Then, in the experiment section, we validate the StackedGP model on different synthetic scenarios and two real-data applications.

2 Stacked Gaussian Process

The Gaussian process is a popular non-parametric model that has many uses in machine learning [12, 18, 19]. In supervised learning tasks, the output $z_i$ can be modeled as $z_i = g_i(x_i) + \epsilon_i$, where $g_i(x_i)$ represents an unknown function $g(x) \sim GP(0, k_z)$ of the input data $x_i$, where $\epsilon_i \sim N(0, \sigma^2)$. The kernel $k_z(x_i, x_j)$, e.g. RBF kernel $\exp\{-\theta_z(x_i-x_j)^2\}$, is used to measure the similarity between $x_i$ and $x_j$. Given a new input $x^*$, the predicted $z^*$ follows normal distribution $z^* \sim N(\mu_z, \sigma_z^2)$ with predicted mean $\mu_z = k_z(x^*, x^*) + \sigma^2 - k_z(x^*, x^*)k_z^{-1}$. We want to build a hierarchical GP to model an $M_l$ dimensional function $y(x)$ as shown in Figure 1. The model has $L$ stacking layers with each layer having $M_l$ GPs node ($l$ refers to the index of the layer and the value of $M_l$ can be different from layer to layer). We assume that we are given the following set of training datasets $D_{train} = \{D_1, D_2, ... D_Q\}$, where $Q = \sum_{l=1}^{L} M_l$ represents the total number of nodes in the model. Our motivation for this stacked model is to independently train each node using its own available data $D_q$ where $q = 1, Q$. Thus, each node acts as a standalone standard GP [12], where the hyper-parameter optimization/inference is conducted using node specific datasets.

The predictions using the StackedGP requires the propagation of uncertainty through the stacked model to determine the quantities of interest $y(x)$. The inputs of each node are outputs of other Gaussian processes, thus sampling methods such as Monte Carlo may be used to obtain estimates for QoIs with quantified uncertainties. In this section, we derive closed form solutions for the first and second moments for predictions of a StackedGP with RBF and polynomial kernels. The section starts with a simple 2 layer StackedGP and then an extension to arbitrary layers and nodes is presented.

2.1 RBF Kernel

Consider the following simple StackedGP: $L = 2$ and $M_l = 1$ for $l = 1, 2$. Here $x$ is the input to the first layer, $z$ is the output of the first layer and the input to the second one, and $y$ is the final output. The predicted mean of the StackedGP given an input $x^*$ can be obtained using the law of total expectation by integrating out the intermediate variable $z^*$:

$$E[y^*|y, (x^*)] = E_{z^*} \{E[y^*|y, (x^*), z^*]\}$$ (1)

Here, $E[y^*|y, (x^*), z^*] = k_y^T C_y^{-1} y$ is the expectation of the a standard GP with input $z$ and output $y$, and it can be expanded as follows:

$$E[y^*|y, (x^*), z^*] = y^T \sum_{i=1}^{F} C_y^{-1}(i) k_y(z^*, z_i)$$ (2)

where $C_y$ is the covariance matrix of the second layer GP and $k_y(z^*, z_i)$ is the kernel between the predicted
point \( z^* \) and the \( i^{th} \) training point \( z_i \). \( F \) is the number of training points for the target node. The final predicted analytical mean of \( y^* \) can be written as

\[
E[y^*|y,x^*] = y^T \sum_{i=1}^{F} C_y^{-1}(i,i)E_{z_i^*}\{k_y(z^*,z_i)\}
\]

(3)

where \( E_{z_i^*}\{k_y(z^*,z_i)\} \) is the key integration to obtain the analytical predicted mean in the stacked GP. Given that the predicted random variable \( z^* \) from the first layer node is normally distributed, with \( \mu_{z^*} \) and \( \sigma_{z^*}^2 \) as the predicted mean and variance of the GP node in the first layer, we can obtain analytical expectations for RBF and polynomial kernels.

In the case of RBF kernels, \( k_y(z_i,z_j) = \phi \exp \left\{ -\theta(z_i - z_j)^2 \right\} \), we obtain the following analytical mean of StackedGP.

\[
E[y^*|y,(x^*)] = \phi y^T \left\{ \frac{1}{\sigma_{z_i^*}^2 + (1/2\theta)} \right\}
\]

(4)

\[
\times \sum_{i=1}^{F} C_y^{-1}(i,i) \exp \left\{ -\frac{(z_i - \mu_{z_i^*})^2}{2(\sigma_{z_i^*}^2 + 1/2\theta)} \right\}
\]

where \( \theta \) is the corresponding length scale in the target node. Also, \( \phi \) is the variance kernel and \( \phi y^2 \) is the training points that have been used during training of the target GP node. The same method can be applied to polynomial kernels as will be demonstrated in the later sections.

We can be generalized the analytical mean in Eq. 4 to any number of input nodes in the first layer. By defining \( \mathbf{w} \in \mathbb{R}^{N \times 1} \) by \( \mathbf{w}_i = \prod_{j=1}^{M} \sqrt{\frac{1}{(1/2\theta) + \sigma_{z_i^*}^2}} \) for \( i = 1..N \), where \( M \) is the number of input dimension to the target GP node and \( N \) is the number of testing points. Also, we need to define \( \mathbf{Q} \in \mathbb{R}^{N \times F} \) by \( Q_{i,f} = \phi \exp \left\{ \sum_{j=1}^{M} -\frac{(z_{ij} - \mu_{z_i^*})^2}{2(1/(2\theta) + \sigma_{z_i^*}^2)} \right\} \) for \( i = 1..N \) and \( f = 1..F \).

The predicted mean corresponding to arbitrary number of inputs can be written using the following equation where the symbol "\( \odot \)" is used to refer element by element product.

\[
E[y^*] = \mathbf{w} \odot (\mathbf{Q} C^{-1}) y
\]

(5)

Note that the predicted mean of the StackedGP has the same form as the standard GP \( k_y^T C_y^{-1} y \) but with two differences. First, \( \mathbf{w} \) acts as a weight to the predicted mean and it reflects the uncertainty of \( z_i \) based on \( \sigma_{z_i^*}^2 \). If we set \( \sigma_{z_i^*}^2 \) to zero, we obtain \( \mathbf{w} \) is equal to one. Second, \( \mathbf{Q} \) measures the similarity with the predicted mean \( \mu_{z_i^*} \) from the previous layer instead of the direct input. Its can be interpreted as RBF kernels with length scales that take into account the input uncertainty \( \sigma_{z_i^*}^2 \). The current length scale is \( 2((1/(2\theta)) + \sigma_{z_i^*}^2) \) instead of \( 1/(2\theta) \).

We assume that outputs of layers that propagate as inputs to the next layers’ nodes are independent. However, this is for convenience, but the methodology can accommodate correlated inputs. In our numerical results, we have used Independent Component Analysis (ICA) to obtain independent projections (this procedure does not include the deterministic input observations \( x \)).

The final predicted variance for the target node can be obtained in a similar way by using the law of total variance.

\[
\text{Var} (y^*|y,(x^*)) = \text{E}_{z_i^*} \left[ \text{Var} (y^*|y,(x^*)) \right] + \text{Var}_{z_i^*} (E [y^*|y,(x^*))
\]

\[
= \sigma_y^2 - \text{E}_{z_i^*} \left[ k_y^T C_y^{-1} k_y \right] + \text{Var}_{z_i^*} \left( k_y^T C_y^{-1} y \right) + \text{E}_{z_i^*} \left( k_y(z_i^*,z_i^*) \right)
\]

\[
= \sum_{i=1}^{F} \sum_{j=1}^{F} C_y^{-1}(i,j) E_{z_i^*} \left[ k_y(z_i^*,z_j^*) k_y(z_j^*,z_i^*) \right]
\]

\[
\sum_{i=1}^{F} \sum_{j=1}^{F} C_y^{-1}(i,j) \exp \left\{ -\frac{(z_i - z_j)^2}{2(1/(2\theta) + \sigma_{z_i^*}^2)} \right\}
\]

(6)

where \( \sigma_y^2 \) is the variance noise of the target GP. \( \Delta_1 \) can be expanded by a summation over expectations of elements \( E_{z_i^*} \left[ C_y^{-1}(i,j) k_y, k_y \right] \) as described in the following.

\[
\Delta_1 = \sum_{i=1}^{F} \sum_{j=1}^{F} C_y^{-1}(i,j) E_{z_i^*} \left[ k_y(z_i^*,z_j^*) k_y(z_j^*,z_i^*) \right]
\]

Therefore, following the same idea in equation 4 and in the case of RBF kernel, we can obtain analytical expectations.

\[
\Delta_1 = \phi^2 \sum_{i=1}^{F} \sum_{j=1}^{F} C_y^{-1}(i,j) \exp \left\{ -\frac{(z_i - z_j)^2}{2(1/(2\theta) + \sigma_{z_i^*}^2)} \right\}
\]

(7)

Regarding \( \Delta_2 \), the variance of \( (k_y^T C_y^{-1} y) \) over \( z^* \) can be written as \( y^T C_y^{-1} \text{Var}_{z_i^*z_j^*} (k_y) C_y^{-1} y \) where

\[
\text{Var}_{z_i^*} (k_y) = E_{z_i^*} [k_y k_y^T] - E_{z_i^*} [k_y] E_{z_i^*} [k_y^T] = \Sigma_k
\]

and

\[
\Sigma_k (i,j) = E_{z_i^*} \left[ k_y(z_i^*,z_i^*) k_y(z_j^*,z_i^*) \right] - E_{z_i^*} \left[ k_y(z_i^*,z_i^*) \right] E_{z_i^*} \left[ k_y(z_i^*,z_i^*) \right]
\]

\[
\sum_{i=1}^{F} \sum_{j=1}^{F} C_y^{-1}(i,j) \exp \left\{ -\frac{(z_i - z_j)^2}{2(1/(2\theta) + \sigma_{z_i^*}^2)} \right\}
\]

\[
\sum_{i=1}^{F} \sum_{j=1}^{F} C_y^{-1}(i,j) \exp \left\{ -\frac{(z_i - z_j)^2}{2(1/(2\theta) + \sigma_{z_i^*}^2)} \right\}
\]

(8)
Here, the expectations of $\delta_3$ and $\delta_4$ have been obtained in Eq. 4 and the expectation $\delta_1$ is given by Eq. 7. Given that $z_i$ is the $i^{th}$ input training point for the target node, the analytical variance of the StackedGP becomes:

$$\text{Var}(y^*|y, x^*) = \phi + \sigma_y^2 + y^T C_y^{-1} \Sigma_f C_y^{-1} y$$

$$- \phi^2 \sum_{i=1}^{F} \sum_{j=1}^{F} C_y^{-1}(i, j) \{ \frac{1/(4\theta_i)}{1/(4\theta_i) + \sigma_{z_i}^2} \}$$

$$\times \exp \left\{ - \frac{\theta(z_i - z_j)^2}{2} - \frac{((z_i + z_j)/2 - \mu_{z_i} + \mu_{z_j})^2}{2(1/(4\theta_i) + \sigma_{z_i}^2)} \right\}$$

To generalize the Eq. 9 to an arbitrary number of inputs, we need to define several other terms: $u_i = \prod_{j=1}^{M} \left\{ \frac{1/(4\theta_j)}{(1/(4\theta_j) + \sigma_{z_i}^2)} \right\}$. $i = 1...N$ (10)

$$P_{i,f,f'} = \phi^2 \exp \left\{ - \sum_{j=1}^{M} \left\{ \frac{\theta_j (z_{ij} - z_{ij'})^2}{2} + \frac{((z_{ij} + z_{ij'})/2 - \mu_{z_{ij}} + \mu_{z_{ij'}})^2}{2(1/(4\theta_j) + \sigma_{z_{ij}}^2)} \right\} \right\}$$

(11)

$$T_{i,f,f'} = \exp \left\{ \sum_{j=1}^{M} \frac{(z_{ij} - \mu_{z_{ij}})^2 + (z_{ij'} - \mu_{z_{ij'}})^2}{2(1/(2\theta_j) + \sigma_{z_{ij}}^2)} \right\}$$

where $u_i$ and $w_i$ are scalers and $P_i$ and $T_i$ are $F \times F$, elements of which are defined above.

$$\zeta_i = y^T C_y^{-1} \psi_i C_y^{-1} y$$

Finally, the predicted variance of StackedGP $\text{var}[y^*]$ is given by:

$$\text{var}[y^*] = \phi + \varepsilon + \zeta_i - u_i (C^{-1} \circ P_i)$$

(15)

Similar as before, if we set the uncertainty that comes from the first layer $\sigma_{z_i}^2$ to zero, we obtain the same standard variance of a Gaussian process. In this case the vectors $u$ and $w$ will equal to one and $\zeta_i$ is equal to zero.

We can write $P_i = k_f(z_f, \mu_{z_f}) k_f(z_f', \mu_{z_f'})$, which results in a predicted variance of the stacked GP similar to the standard GP, $\phi + \varepsilon - k_f C_y^{-1} k_f$. Here, $k_f$ is the kernel evaluated at the training point and the predicted mean of the first layer. The expressions derived here are matching with those in Ref. [4]. In the next section we expand these derivations to polynomial kernels.

### 2.2 Linear and Polynomial Kernel

Following the same approach, we obtain analytical expressions for the predicted mean and variance of StackedGP when using linear and polynomial kernel. We start with the simple case, $L = 2$, $M_1 = 2$ and $M_2 = 1$, which leads to the following $d$-order polynomial kernel for the target node $k_y((z_1, \ldots , z_d), (z_1', \ldots , z_d')) = \left( [z_1 \ast z_1'] + (z_1' \ast z_2') \right)^d$.

From Eq. 3, the predicted mean can be written as

$$\text{E}[y^*|y, (x_1^*, x_2^*)] = y^T C_y^{-1} \psi_1 C_y^{-1} y$$

where this expectation follow the multinomial expansion.

$$\sum_{p_1+p_2+\ldots+p_M=d} \prod_{1 \leq t \leq M} \left[ \frac{d}{p_t} \right] [\sigma_{z_i}^{2u} \mu_i^{p_t-2u}]$$

(17)

where, $E_z \left( [z_1 \ast z_1'] + (z_1' \ast z_2') \right)^d$ (16)

with $\left( \begin{array}{c} p_1, p_2, \ldots, p_M \\ \vdots \\ p_t 
\end{array} \right) = \frac{d!}{p_t! (p_t)!}$ and the coefficient $a_{p_t}$, follows the non-central moment of normal distribution

$$\sum_{u=0}^{\frac{d}{2u}} \left( \begin{array}{c} pt \\ 2u 
\end{array} \right) (2u-1)!! \sigma_{z_i}^{2u} \mu_i^{p_t-2u}.$$ Thus, the analytical mean is given by

$$\text{E}[y^*|y, (x_1^*, x_2^*)] = v^T C_y^{-1} y$$

where the elements of $v \in \mathbb{R}^{N \times 1}$ are calculated from the multinomial expansion in Eq. 17.

Regarding the predictive variance in Eq. 6, $\Delta_1$ can be calculated as $\sum_{f,f'} \left( C_y^{-1} \circ H \right)$ where the elements of $H \in \mathbb{R}^{P \times F}$ are obtained using the following multinomial expansion

$$\sum_{p_1+\ldots+p_M=d} \sum_{q_1+\ldots+q_M=d} \left( \begin{array}{c} d \\ q_1 \ldots q_M 
\end{array} \right) \prod_{1 \leq t \leq M} \left[ a_{p_t} q_t^{p_t-2u} \right]$$

where the elements of $v \in \mathbb{R}^{N \times 1}$ are calculated from the multinomial expansion in Eq. 17.
Table 1: Applying the stacked model shown in Fig 2 on different toy scenarios. These figures show the training set and the final predictions in different toy scenarios for the input $x_1$ and $x_2$. All figures follow this legend.

| Synthetic data | $z_1 = x_1^2$ | $z_2 = \ln(x_1^2)$ | $z_1 = \sin(x_1)$ | $z_2 = \sin(x_2)$ | $z_1 = x_1^2$ |
|----------------|---------------|-------------------|-----------------|-----------------|--------------|
| $y = z_1 + 2 \ast z_2$ | $y = \sin(\sqrt{z_1 + z_2})$ | $y = z_1 \ast z_2$ | $y = \sqrt{(z_1 + z_2)} + 3 \ast \cos(\sqrt{z_1 + z_2}) + 5$ |

Training

Testing

| RMSE | AvgRatio |
|------|----------|
| 0.0007 | 0.15 |
| 0.040 | 0.17 |
| 0.0067 | 0.10 |
| 0.33 | 0.41 |

Here, the coefficient $a_{p_1,q_1}$ follows the non-central moment of normal distribution which can be express as $\sum_{u=0}^{[p_1+q_1]} \frac{p_1 + q_1}{2u} (2u - 1)!! [\sigma_{z_i}^2 u^{p_1 + q_1 - 2u}]^2$.

The $\Sigma_K$ in Eq. 8 can be calculated as $H - \mathbf{vv}^T$. In additions the last term of Eq. 6 is a scaler that follows the multinomial expansion

$\sum_{p_1+p_2+\ldots+p_M=d}^{d} \mathbf{H}_{\mathbf{1},d} [a_{2p_1}]$ where the coefficient $a_{2p_1}$ follows the non-central moment of normal distribution,

$\sum_{u=0}^{[2p_1]} \frac{2 \ast p_1}{2u} (2u - 1)!! [\sigma_{z_i}^2 u^{2p_1 - 2u}]^2$. Therefore, we can generalize this formula as the follows:

$\text{Var}(y^* | y, (x_1^*, x_2^*)) = \sigma_y^2 + \Delta_3 + y^T C_y^{-1} \Sigma_K C_y^{-1} y - \sum_{f,f'} [C_y^{-1} \circ H]$

Note that for $d = 1$, we obtain the analytical expectations of the StackedGP with linear kernel. Also, it can be seen from the analytical derivation that the prediction mean and variance can be propagated from node to node allowing for flexible kernel selections at the node level. Analytical expressions for the first two moments can be obtain also for sum RBF and polynomials kernels using the provided derivations.

3 EXPERIMENTS

In this section, the StackedGP is used on both synthetic data and two real-world applications. The synthetic data is used to demonstrate that the StackedGP is able to capture the outputs of composite functions. In the first real-world application, we use the StackedGP to combine knowledge from two datasets to predict the burned area as part of a forest fire application. The second real-word application corresponds to the Jura dataset, where the StackedGP is used to enhance the prediction of primary responses using intermediate predictions of secondary responses.

3.1 Synthetic datasets

StackedGPs are build using synthetically generated data from four composite functions as shown in Table 1. One set of data is generated for the mappings between $(x_1, x_2)$ and $(z_1, z_2)$, and another data set is generated for the mappings between $(z_1, z_2)$ and $y$. The two datasets are used to build three independent GPs, which are then stacked to obtain a StackedGP, see Figure 2. Table 1 shows the training set and the testing set for each scenario, as well as the ability of the StackedGP to capture different non-linear hierarchal functions.

The root mean square error (RMSE) is used to measure the performance of the stacked model by comparing the prediction of the StackedGP at various
inputs \((x_1, x_2)\) and the true value \(y\) of the composite function at those inputs. In addition, the following average ratio \(\frac{|\hat{y} - y^*|}{\sigma}\) is reported to verify that the true value falls within the 95% credible interval as predicted by the model. This corresponds to a departure of less than 2.0 standard deviations from the mean. Predictions from StackedGP are well inside the credible interval with a maximum average ratio of 0.41. Here, \(\hat{y}\) is the predicted mean, \(y^*\) is the actual true value, and \(\sigma\) is the analytical predicted standard deviation.

![Figure 2: StackedGP for predicting the output of a composite function.](image)

Figure 2: StackedGP for predicting the output of a composite function. The input to this model are \(x_1\) and \(x_2\), and the final output is \(y\) with \(z_1\) and \(z_2\) as outputs from the middle layer.

### 3.2 Forest Fire Dataset

The prediction of the burned area from forest fires has been discussed in different studies [5, 6]. The burned area of forest fires has been predicted using meteorological conditions (e.g. temperature, wind) and/or several Canadian forest Fire Weather Indices [15] for rating fire danger, namely Fine Fuel Moisture Code (FFMC), Duff Moisture Code (DMC), Drought Code (DC), Initial Spread Index (ISI), and Buildup Index (BUI), as shown in Figure 3.

In this application we are interested in developing a StackedGP by first modeling the fire indices using meteorological variables \(T\) from one dataset presented in Ref. [16] and then model the burned area based on fire indices using another dataset presented in Ref. [6]. The proposed StackedGP is depicted in Figure 4. The GP nodes corresponding to the four fire indices (FFMC, DMC, DC, and ISI) are trained from data published in Ref. [16] according to the hierarchical structure shown in Figure 3. While the second dataset [6] contains meteorological conditions along with the fire indices and burned area, we assume that the meteorological conditions are missing in the training phase from this dataset and use only the fire indices and burned area data to train the GP node in the last layer of the StackedGP.

A 10-fold cross validation is applied to the dataset published in Ref. [6] to train the burned area node and test the whole StackedGP model. Because of the skewed distribution of the burned area values, instead of directly modeling the burned area using StackedGP, we have modeled the log of the burned area. The final mean and variance of the burned area \(B[T]\) as a function of the meteorological conditions \(T\) is given by Eqs. 19 and 20 respectively. In additions, we have found that scaling the target variable to have zero mean and unit variance to be a beneficial preprocessing step.

![Figure 3: Structure of the Fire Weather Index (FWI) system module of the Canadian Forest Fire Danger Rating System [15].](image)

\[
E[B[T]] = (e^{\sigma_{lnB}^2} - 1)e^{\mu_{lnB} + \sigma_{lnB}^2} \tag{19}
\]

\[
Var[B] = e^{2\mu_{lnB} + 0.5\sigma_{lnB}^2} - e^{2\mu_{lnB} + \sigma_{lnB}^2} \tag{20}
\]

Here, \(\mu_{lnB}\) and \(\sigma_{lnB}\) can be found from the output of the probabilistic analytical StackedGP in Eqs. 5, 15 respectively.

The result of modeling the burned area using the StackedGP is shown in Table 2. The StackedGP model is compared with the results of other 5 regression models reported in Ref. [6]. Because these regression models have been tested using different input spaces, Table 2 tabulates the best results achieved by each model as described in Ref. [6]. Even though the StackedGP predicts the burned area based on estimated indices from the first dataset and not the actual values as presented in the second dataset, it is still able to give comparable results with the other models that make use of meteorological conditions and/or fire indices available in the second dataset. This experiment emphasizes that the StackedGP is able to combine knowledge from different datasets with noticeable performance.
Figure 4: StackedGP for predicting burned area based on estimated FWI indices. Letters P,T,RH,W stands for Precipitation, Temperature, Relative Humidity and Wind respectively. Also, the first two layers are trained using $D_1$ dataset, while the $D_2$ dataset are used to train the last layer.

Table 2: Predictive results using different models. The input for each model is T for meteorological features and FWI for fire indices. Multiple Regression (MR), Decision trees (DT), Random Forests (RF), And Neural Networks (NN).

| Model     | Input | MAE  | RMSE |
|-----------|-------|------|------|
| StackedGP | T     | 12.80| 46.0 |
| MR        | FWI   | 13   | 64.5 |
| DT        | T     | 13.18| 64.5 |
| RF        | T     | 12.98| 64.4 |
| NN        | T     | 13.08| 64.6 |
| SVM       | T     | 12.71| 64.7 |

3.3 Jura Dataset

In this subsection we use Jura dataset collected by the Swiss Federal Institute of Technology at Lausanne [2,17]. Data contains concentration samples of several heavy metals at 359 different locations. Similar to the experiments setup that have been contacted in [1,9,20], we are interesting in predicting cadmium concentrations, the primary response at 100 locations given 259 training measurement points. The training data contains location information and concentrations of various metals (Cd, Zn, Ni, Cr, Co, Pb and Cu) at the sampled sites. The primary response is the concentration of Cd, and the other metals are considered secondary responses.

Note that standard Gaussian process such as multi-kriging [19] models each response variable independently and thus knowledge of secondary responses cannot help in predicting the primary one [13]. Thus in this case a standard Gaussian process (StandardGP) will use a training dataset with only locations as inputs and Cd measurements as target [1,20]. Multi-output regression models such as co-kriging [7] can use the correlation between secondary and primary response to improve the prediction of Cd. The StackedGP, while it does not model the correlation between primary and secondary responses, it can be used to enhance the prediction of the primary response using intermediate predictions of the secondary responses.

In the special case, the heterotopic case [9], when the primary target is undersampled relative to the secondary variables leading to have access to the secondary information such as Ni and Zn at the 100 locations being estimated, then standard Gaussian processes can be built to have Ni and Zn directly as inputs. Here we will denote it as StandardGP(Zn,Ni). This is also the case for comparing our results with other six multi-task regression models as reported in Ref. [20] and tabulated in Table 3.

The first proposed StackedGP uses the first layer to model Zn and Ni based on locations and the second layer to model Cd based on the locations and the estimated output of the first layer, see Figure 5. In the heterotopic case the StackedGP can use directly the available measurements of Ni and Zn instead of predictions by setting the uncertainty associated with these measurements to zero. In this case the StackedGP acts as the StandardGP(Zn,Ni).

Three other structures are proposed by using intermediate predictions of Co, Cr, and Co and Cr together. In this case, we have a three layer StackedGP to model Cd, see Figure 6. The first layer is the same as in the previous setup. The second layer models i-
intermediate responses (Co, Cr, and Co and Cr). The third layer is used to model Cd based on the second layer predictions in additions to the input/output of the first layer, namely location and Zn and Ni.

![Figure 6: StackedGP for predicting Cadmium Cd based on estimated Zn, Ni, Co, and Cr at location of interest X and Y.](image)

Table 3 shows the results of these stacked structures, StackedGP(Co), StackedGP(Cr) and StackedGP(Co,Cr). While measurements of Ni and Zn are available in the testing scenarios, there are no measurements for Co and Cr during testing. Thus, Cd predictions of these three StackedGPs rely on predictions of Co and Cr using locations and Ni and Zn measurements at these locations.

Table 3: Performance on modeling Cd using different two/three layers StackedGP structures with mean absolute error (MAE) as performance metric.

| Model                     | MAE   |
|---------------------------|-------|
| StackedGP                 | 0.3833|
| StackedGP(Co)             | 0.3617|
| StackedGP(Cr)             | 0.3884|
| StackedGP(Co,Cr)          | 0.3602|
| GPRN(VB) [20]             | 0.4040|
| SLFM(VB) [13]             | 0.4247|
| SLFM [13]                 | 0.4578|
| ICM [9]                   | 0.4608|
| CMOGP [1]                 | 0.4552|
| Co-Kriging                | 0.51  |
| StandardGP(Zn,Ni)         | 0.3833|
| StandardGP                | 0.5714|

The mean absolute error (MAE) between the true and estimated cadmium is calculated at the 100 target locations. Overall StackedGP gives a better results as compared with the other models. Also note that in the case when Zn and Ni measurements are available as assumed by the other multi-output regression models [1, 20], then a StandardGP(Ni,Zn) can provide a lower MAE than the other six multi-output regression models. However, the stacked Gaussian process can provide a better performance over the Standard(Zn,Ni) by making use of intermediate predictions of secondary responses.

In additions, the complexity of most of multi-task models is $O(F^3 p^3)$, where $p$ is the number of output responses [20]. However, StackedGP scales linearly with the number of nodes in the structure because of the independent training of the nodes, which can be easily parallelized. While the standard GP requires $O(F^3)$, several other sparse approximation techniques can be used to reduce this complexity to $O(FG^2)$, where $G$ is much smaller than $G$ [8, 14].

For all these experiments we found that the log transformation and normalization can lead to better results. For multi-responses in the middle layer, we used Independent Component Analysis (ICA) to obtain independent projections of secondary responses. This is required as the current derivation assumes that inputs to a GP node are independent.

4 Conclusion

A stacked Gaussian process is proposed as a supervised non-parametric hierarchical regression framework that provides analytical first and second order moments of quantities of interests that require composition of functions. Analytical expression for the mean and variance of the StackedGP can be obtained for both RBF and polynomial as well as kernels obtained via their sum. The StackedGP can be extended to any number of nodes and layers and has no restriction in selecting a suitable kernel per node.

The numerical results show the utility of using StackedGP to learn from different datasets and propagate the uncertainty to quantity of interests. While it is not specifically designed to model correlations between secondary and primary responses, it can be used to enhance the prediction of primary responses by creating an intermediate layer for predictions of secondary responses. This comes with a lower computational complexity as compared with multi-output methods - and can make use of off-the-shelves Gaussian processes.

While in the current paper we assume that outputs of intermediate layers are independent and resolve this using ICA preprocessing, we plan to extend
our derivation to account for these correlations in the next study. This will allow multi-output models, such as GPRN, to act as nodes in the proposed StackedGP.

References

[1] Mauricio A Alvarez and Neil D Lawrence. Computationally efficient convolved multiple output gaussian processes. *Journal of Machine Learning Research*, 12(May):1459–1500, 2011.

[2] O Atteia, J-P Dubois, and R Webster. Geostatistical analysis of soil contamination in the swiss jura. *Environmental Pollution*, 86(3):315–327, 1994.

[3] Phillip Boyle and Marcus Frean. Dependent gaussian processes. In *Advances in neural information processing systems*, pages 217–224, 2004.

[4] Joaquin Quinonero Candela, Agathe Girard, Jan Larsen, and Carl Edward Rasmussen. Propagation of uncertainty in bayesian kernel models-application to multiple-step ahead forecasting. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP’03)*. 2003 IEEE International Conference on, volume 2, pages II–701. IEEE, 2003.

[5] M Castelli, L Vanneschi, and A Popović. Predicting burned areas of forest fires: an artificial intelligence approach. *Fire Ecology*, 11(1):106–118, 2015.

[6] Paulo Cortez and Aníbal de Jesus Raimundo Morais. A data mining approach to predict forest fires using meteorological data. 2007.

[7] Noel Cressie. Statistics for spatial data. *Terra Nova*, 4(5):613–617, 1992.

[8] Andreas Damianou, Michalis K Titsias, and Neil D Lawrence. Variational gaussian process dynamical systems. In *Advances in Neural Information Processing Systems*, pages 2510–2518, 2011.

[9] Pierre Goovaerts. *Geostatistics for natural resources evaluation*. Oxford University Press on Demand, 1997.

[10] Hui Li, Asif Chowdhury, Gabriel Terejanu, Anindya Chanda, and Sourav Banerjee. A Stacked Gaussian Process for Predicting Geographical Incidence of Aflatoxin with Quantified Uncertainties . In *International Conference on Advances in Geographic Information Systems ACM SIGSPATIAL*, Seattle, Washington, November 2015.

[11] Marion Neumann, Kristian Kersting, Zhao Xu, and Daniel Schulz. Stacked gaussian process learning. In *2009 Ninth IEEE International Conference on Data Mining*, pages 387–396. IEEE, 2009.

[12] Carl Edward Rasmussen. Gaussian processes for machine learning. 2006.

[13] Matthias Seeger, Yee-Whye Teh, and Michael Jordan. Semiparametric latent factor models. Technical report, 2005.

[14] Edward Lloyd Snelson. *Flexible and efficient Gaussian process models for machine learning*. PhD thesis, Citeseer, 2007.

[15] Stephen W Taylor and Martin E Alexander. Science, technology, and human factors in fire danger rating: the canadian experience. *International Journal of Wildland Fire*, 15(1):121–135, 2006.

[16] CE Van Wagner, TL Pickett, et al. *Equations and FORTRAN program for the Canadian forest fire weather index system*, volume 33. 1985.

[17] R Webster, O Atteia, and J-P DUBOIS. Coregionalization of trace metals in the soil in the swiss jura. European Journal of Soil Science, 45(2):205–218, 1994.

[18] Christopher KI Williams. Prediction with gaussian processes: From linear regression to linear prediction and beyond. In *Learning in graphical models*, pages 599–621. Springer, 1998.

[19] Christopher KI Williams and Carl Edward Rasmussen. Gaussian processes for regression. 1996.

[20] Andrew G. Wilson, David A. Knowles, and Zoubin Ghahramani. Gaussian process regression networks. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML)*, Edinburgh, June 2012. Omnipress.