Statistica Sinica Preprint No: SS-2020-0456	
Title	A Clustered Gaussian Process Model for Computer Experiments
Manuscript ID	SS-2020-0456
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202020.0456
Complete List of Authors	Chih-Li Sung, Benjamin Haaland, Youngdeok Hwang and Siyuan Lu
Corresponding Author	Chih-Li Sung
E-mail	sungchih@msu.edu
Notice: Accepted version subject to English editing.	

A Clustered Gaussian Process Model for Computer Experiments

Chih-Li Sung[1], Benjamin Haaland[2], Youngdeok Hwang[3], Siyuan Lu[4]

[1]*Michigan State University,* [2]*University of Utah*

[3]*City University of New York,* [4]*IBM Thomas J. Watson Research Center*

Abstract: The Gaussian process has been one of the most important approaches for emulating computer simulations. However, the stationarity assumption that is common to Gaussian process emulation and computational intractability for large-scale datasets limit accuracy and feasibility in practice. In this article, we propose a clustered Gaussian process model which *simultaneously* segments the input data into multiple clusters and fits a Gaussian process model in each. The model parameters and the clusters are learned through the efficient stochastic expectation-maximization, which allows for emulation for large-scale computer simulations. Importantly, the proposed method provides valuable model interpretability by identifying clusters, which reveal hidden patterns in the input-output relationship. The number of clusters, which controls the bias-variance trade-off, is efficiently selected via cross-validation to ensure accurate predictions. In our simulations as well as a real application to solar irradiance emulation, our proposed method has smaller mean squared errors than its main competitors, with competitive computation time, and provides valuable insights from data by discovering the clusters. An `R` package for the proposed methodology is provided in an open repository.

Key words and phrases: Non-stationarity, large-scale data, uncertainty quantification, mixture models, solar irradiance emulation

1. Introduction

Gaussian processes (GPs) have been one of the most popular modeling tools in various research topics, such as spatial statistics (Stein, 2012), computer experiments (Fang et al., 2005; Santner

et al., 2018; Gramacy, 2020), machine learning (Rasmussen and Williams, 2006), and robot control (Nguyen-Tuong and Peters, 2011). Gaussian processes provide the flexibility for a prior probability distribution over functions in Bayesian inference, and the posterior can be used not only to estimate the unknown function at an unknown point but also to quantify uncertainty in this estimate. This explicit probabilistic formulation for GPs has proved to be powerful for general function learning problems. However, its use is often limited due to the following challenges. First, GP posterior involves $O(N^3)$ computational complexity and $O(N^2)$ storage where N is the sample size, so that GP emulation becomes infeasible for moderately large datasets, say $N = 10^3$. Second, a GP model often utilizes a stationary covariance function, in the sense that the outputs with the same separation of any two inputs are assumed to have an equal covariance. We call a GP with a stationary covariance function a stationary GP throughout this article. This assumption is violated in many practical applications. Figure 1 demonstrates an illustrative example in Gramacy and Lee (2009) where a stationary GP may perform very poorly when the underlying function indeed consists of two different functions: a relatively rough function in the region $x \in [0, 10]$ and a simple linear function in the region $x \in [10, 20]$. Figure 1 shows that a stationary GP results in very poor prediction particularly in the region $x \in [10, 20]$ with very high uncertainty. See more examples in Higdon et al. (1999); Paciorek and Schervish (2006); Bui-Thanh et al. (2012).

These two challenges to GP modeling are common in practice and have attracted lots of attention lately. To name a few, sparse approximation (Quiñonero-Candela and Rasmussen, 2005; Sang and Huang, 2012), covariance tapering (Furrer et al., 2006), inducing inputs (Snelson and Ghahramani, 2006; Titsias, 2009), multi-step interpolation (Haaland and Qian, 2011), special

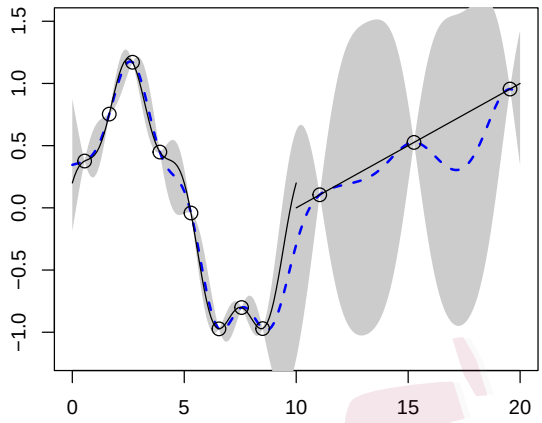

Figure 1: An example of stationary Gaussian process emulation applied to a non-stationary function. Black line is the true function, black dots represent collected data. Blue dashed line represents a stationary Gaussian process emulator, with the gray shaded region providing a pointwise 95% confidence band.

designs (Plumlee, 2014), and multi-resolution approximation (Nychka et al., 2015), address the computational issue for large datasets. For non-stationarity, Higdon et al. (1999); Higdon (2002); Paciorek and Schervish (2006); Plagemann et al. (2008); Plumlee and Apley (2017) adopted nonstationary covariance functions for Gaussian processes. Tresp (2001); Rasmussen and Ghahramani (2002); Kim et al. (2005); Gramacy and Lee (2008) considered multiple Gaussian processes by segmentation in the input spaces. Ba and Joseph (2012) proposed a composite of two Gaussian processes, which respectively capture a smooth global trend and local details. However, only few of them are able to tackle the non-stationarity and computational issues simultaneously. Exceptions include the multi-resolution functional ANOVA approximation (Sung et al., 2020), which uses a group lasso algorithm to identify important basis functions, and the local Gaussian process approximation, which selects a small subsample to fit a Gaussian process model for each predictive location (Gramacy and Apley, 2015).

In this article, we propose a clustered Gaussian process (clustered GP) to address the two

challenges simultaneously. The clustered GP makes use of the divide-and-conquer idea, which segments the input data into clusters with a hard-assignment clustering approach, in each of which a Gaussian processes is fitted. This makes the computation more tractable for large-scale datasets while retaining the mixture model structure to address the non-stationarity issue. As latent variable models often suffer from computational difficulties, the stochastic EM algorithm (Celeux and Diebolt, 1985) is employed to learn the clusters efficiently. Although combining mixture GP and efficient SEM algorithm has shown to have a potential to simultaneously address non-stationarity and computation challenges, it has not been carefully studied. In addition, the number of the clusters plays a crucial role for a mixture GP model, which controls the flexibility and non-stationarity of the model, and thus a systematic criteria to select the tuning parameter is necessary; however, little attention has been paid in this regard. The cross-validation criterion, which retains efficient computation, is carefully studied in this article. Importantly, unlike many existing methods, the clustered GP retains the features of unsupervised learning approaches which reveal hidden patterns in the data that can lead to interesting model interpretation, and provide important insights about the underlying aspects of the problem by showing some grouping structures.

It is worth noting that, unlike traditional unsupervised learning, such as K-means clustering and the GP clustering of Kim and Lee (2007), which aims to partition the observations into groups based on their similarities in the input space and does not make use of information contained in the output, the main purpose of clustered GP is to build a flexible model that can produce accurate prediction at new input locations, and the assignments to each cluster are determined by both inputs and outputs. These clusters indicate that the observations within each

of the clusters share similar behavior of input-output relationships, and they can be used for data compression in a supervised fashion to save computational and storage costs as in Joseph and Mak (2021).

The remainder of this article is organized as follows. In Section 2, the clustered GP model is introduced, along with its relationship to existing methods. In Section 3, our estimation and prediction to fit the clustered GP model using a stochastic expectation-maximization algorithm is described. Computational details are discussed in Section 4. In Section 5, some synthetic examples are demonstrated to show the tractability and prediction performance of the proposed method. A real data application for predicting solar irradiance over the United States is presented in Section 6. Some potential future work is discussed in Section 7. Mathematical proofs are given in Supplementary Materials, and an `R` package, `GPcluster`, is provided in an open repository for practitioners to implement the methodology.

2. Clustered Gaussian Process

2.1 Preliminary: Gaussian Processes

A brief review for Gaussian processes is first given in this section. A Gaussian process (GP) is a stochastic process whose finite dimensional distributions are defined via a mean function $\mu(x)$ and a covariance function $\Sigma(x, x')$ for d-dimensional $x, x' \in \chi \subseteq \mathbb{R}^d$. If the function $y(\cdot)$ is a draw from a GP, then we write

$$y(\cdot) \sim \mathcal{GP}(\mu(\cdot), \Sigma(\cdot, \cdot)).$$

In particular, given n inputs $X = (x_1, \ldots, x_n)$, if $y(\cdot)$ is a GP, then the outputs $Y = (y(x_1), \ldots, y(x_n))$ have a multivariate normal distribution,

$$Y|X \sim \mathcal{N}(\mu(X), \Sigma(X, X)),$$

where $\mu(X) \in \mathbb{R}^n$ and $\Sigma(X, X) \in \mathbb{R}^{n\times n}$ are defined as $(\mu(X))_i = \mu(x_i)$ and $(\Sigma(X, X))_{i,j} = \Sigma(x_i, x_j)$, respectively. Conventionally, $\mu(\cdot)$ is often assumed to be a constant mean, i.e., $\mu(\cdot) = \mu$, and $\Sigma(\cdot, \cdot)$ is assumed to have the form $\sigma^2\Phi_\gamma(\cdot, \cdot)$, where Φ_γ is a correlation function with $\Phi_\gamma(x, x) = 1$ for any $x \in \chi$ and contains the unknown parameter γ. In addition, Φ_γ is often assumed to depend on the displacement between two input locations, that is, $\Phi_\gamma(x, x') = R(x - x')$ for some positive-definite function R. Such a correlation function is called *stationary* correlation function which implies the process $y(\cdot)$ is stationary, since $y(x_1), \ldots, y(x_L)$ and $y(x_1 + h), \ldots, y(x_L + h)$ have the same distribution for any $h \in \mathbb{R}^d$ and $x_1, \ldots, x_L, x_1 + h, \ldots, x_L + h \in \chi$. A common choice for Φ_γ is a power correlation function

$$\Phi_\gamma(x, x') = \exp\{-\|\gamma^T(x - x')\|_2^p\}, \tag{2.1}$$

where p is often fixed to control the smoothness of the output surface, and $\gamma = (\gamma_1, \ldots, \gamma_d)^T$ controls the decay of correlation with respect to the distance between x and x'. Hence, the parameters include $\mu(\cdot), \sigma^2$ and γ and can be estimated by either maximum likelihood estimation or Bayesian estimation. See Fang et al. (2005), Rasmussen and Williams (2006) and Santner et al. (2018) for more details. Importantly, when the interest is in the prediction at an untried x_{new}, whose response could be denoted as y_{new}, the predictive distribution of y_{new} can

be derived by the conditional multivariate normal distribution. In particular, one can show that $y_{\text{new}}|Y, X, x_{\text{new}} \sim \mathcal{N}(\mu^*, (\sigma^*)^2)$, where

$$\mu^* = \mu(x_{\text{new}}) + \Phi_\gamma(x_{\text{new}}, X)\Phi_\gamma(X, X)^{-1}(Y - \mu(X)) \quad \text{and} \tag{2.2}$$

$$(\sigma^*)^2 = \sigma^2 \left(1 - \Phi_\gamma(x_{\text{new}}, X)\Phi_\gamma(X, X)^{-1}\Phi_\gamma(X, x_{\text{new}})\right). \tag{2.3}$$

In practice, the unknown parameters $\mu(\cdot), \sigma^2$ and γ in (2.2) and (2.3) are replaced by their estimates.

2.2 Clustered Gaussian Process

In practice, we might expect the unknown function that we are trying to approximate to exhibit some degree of non-stationarity. A natural conceptual model to take into account such a circumstance would be a mixture GP, where each component of the mixture acts as an approximately stationary model with high accuracy for a subset of the data. That is,

$$\begin{aligned} y(\cdot) \mid z(\cdot) = k &\sim \mathcal{GP}(\mu_k(\cdot), \sigma_k^2 \Phi_{\gamma_k}(\cdot, \cdot)), \quad k = 1, \ldots, K, \\ \Pr(z(x) = k) &= g_k(x; \varphi_k), \quad k = 1, \ldots, K, \end{aligned} \tag{2.4}$$

where $\mu_k(\cdot), \sigma_k^2$ and Φ_{γ_k} are the mean function, variance, and stationary correlation function of the k-th GP, and $g_k(x, \varphi_k)$ is the probability that $z(x) = k$ with unknown parameter φ_k satisfying $\sum_{k=1}^K g_k(x; \varphi_k) = 1$ for any x. It can be seen that in this model, $z(\cdot)$ takes the role of a latent function, which assigns $y(\cdot)$ to one of the K GPs. These models introduce a

non-stationarity by assuming different parameters of the stationary correlation functions in each cluster, dependent on the input space, which allows for the local smoothness of the function of interest, while the conventional GP lacks the ability to adapt the smoothness in the function. This input-dependent smoothness is essential in various applications, such as geo-science, traffic simulations, and robotics (Plagemann et al., 2008). For example, modeling the solar irradiance in Section 6 requires the ability to deal with a varying data density and to account for the local smoothness potentially dependent on the input locations, where the discontinuities may arise at geographic features such as mountain ranges. Such features can help scientists to discover interesting insights that differentiate these clusters.

Now, a little notation is introduced. Given n inputs $X = (x_1, \ldots, x_n)$, denote the corresponding outputs as $Y = (Y(x_1), \ldots, Y(x_n))$. For cluster $k = 1, \ldots, K$, let $\mathcal{P}_k = \{i : z(x_i) = k\}$ denote the set of indices of the observations in cluster k. Additionally, let $Y_{\mathcal{P}_k}$ and $X_{\mathcal{P}_k}$ respectively denote the (ordered) responses and input locations for the observations from cluster k. Then, given $Z = (z_1, \ldots, z_n) \equiv (z(x_1), \ldots, z(x_n))$, the output $Y_{\mathcal{P}_k}$ in each cluster k has the multivariate normal distribution

$$Y_{\mathcal{P}_k} | X_{\mathcal{P}_k} \sim \mathcal{N}(\mu_k(X_{\mathcal{P}_k}), \sigma_k^2 \Phi_{\gamma_k}(X_{\mathcal{P}_k}, X_{\mathcal{P}_k})), \tag{2.5}$$

where the observed y_i's depend on the response values and locations of the other cluster members, in addition to their corresponding input location x_i within each cluster. The latent cluster/mixture component assignments z_i is assumed to be independent across observations i but dependent on input location x_i, so that the (unobserved) cluster assignment likelihood is given

by

$$f(Z|X) = \Pr(z(x_1) = z_1, \ldots, z(x_n) = z_n)$$
$$= \prod_{i=1}^{n} g_{z_i}(x_i; \varphi_{z_i}) = \prod_{k=1}^{K} \prod_{i \in \mathcal{P}_k} g_k(x_i; \varphi_k). \tag{2.6}$$

Then, by combining (2.5) and (2.6), the likelihood function of complete data is

$$f(Y, Z|X) = f(Y|X, Z) f(Z|X)$$
$$= \left(\prod_{k=1}^{K} f_k(Y_{\mathcal{P}_k} | X_{\mathcal{P}_k}; \theta_k) \right) \left(\prod_{k=1}^{K} \prod_{i \in \mathcal{P}_k} g_k(x_i; \varphi_k) \right), \tag{2.7}$$

where f_k is the probability density function of a multivariate normal distribution with parameters $\theta_k \equiv \{\mu_k(\cdot), \sigma_k^2, \gamma_k\}$.

The clustered GP in (2.4) is related to some existing methods. If $z(\cdot)$ is a Bayesian treed model (Chipman et al., 1998, 2002), the model becomes similar to the Bayesian treed GP of Gramacy and Lee (2008). If $z(\cdot)$ assigns cluster memberships based on a Voronoi tessellation, the model bears some similarity to the model of Kim et al. (2005). When $z(\cdot)$ is assumed to be a Dirichlet process or a generalized GP, the model becomes similar to the mixtures of GPs of Tresp (2001) and Rasmussen and Ghahramani (2002), respectively. Despite the similarity, their application is limited in large-scale data setting due to their costly MCMC sampling. Some other work, such as Nguyen-Tuong et al. (2009); Zhang et al. (2019), chose the assignment based on traditional unsupervised clustering methods, such as K-means clustering.

Our modeling approach belongs to the popular model based clustering approach using la-

tent variables within an Expectation-Maximization (EM) framework (e.g., Fraley and Raftery, 2002). A likelihood-based EM approach to estimate the unknown parameters is, however, not straightforward, because strong dependencies among observations due to the GP correlation structure makes computation difficult. One may want to compute the cluster probability $f(Z|X,Y)$, whether for implementing the E-step in the EM algorithm (soft assignment), or updating cluster membership in a K-means type algorithm (hard assignment). Unfortunately, the cluster probability $f(Z|X,Y)$ does not factor beyond being proportional to (2.7), so we cannot compute the cluster membership for each observation separately from one another even though z_i is independent of each other. In the next section, we adopt a stochastic EM algorithm to address this issue, along with computational details associated with our approach.

3. Statistical Inference via Stochastic EM Algorithm

In this section, we present our estimation and prediction approach for the model in (2.4). Our proposed method addresses the aforementioned challenges using the stochastic EM algorithm (SEM, Celeux and Diebolt, 1985). SEM algorithm is particularly suitable for our challenges as it leads to a computationally efficient algorithm in clustered GP while avoiding insignificant local maxima of likelihood functions. SEM herein is a general approach to calculate the conditional expectation required in the E-step of the EM algorithm, while recent studies, such as Cappé and Moulines (2009) and Chen et al. (2018), particularly focuses on the stochastic approximation of the gradient when optimizing the parameters in the M-step, which is applicable to independent observations but is not straightforward for dependent observations like the data herein.

3.1 Stochastic E-step

In the EM-algorithm, the E-step computes the expected value of the log posterior of complete data given the observed data Y:

$$\mathbb{E}[\log f(Y, Z|X)|X, Y, \boldsymbol{\theta}, \boldsymbol{\varphi}] + \log \pi(\boldsymbol{\theta}) + \log \pi(\boldsymbol{\varphi}), \tag{3.8}$$

where $\boldsymbol{\theta} = \{\theta_k\}_{k=1}^K$, $\boldsymbol{\varphi} = \{\varphi_k\}_{k=1}^K$, while $\pi(\boldsymbol{\theta})$ and $\pi(\boldsymbol{\varphi})$ are priors of $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$. We assume θ_k and φ_k are mutually independent through $k = 1, \ldots, K$ so

$$\log \pi(\boldsymbol{\theta}) = \sum_{k=1}^K \log \pi(\theta_k) \quad \text{and} \quad \log \pi(\boldsymbol{\varphi}) = \sum_{k=1}^K \log \pi(\varphi_k). \tag{3.9}$$

Computing the expected value requires the cluster probabilities $f(Z|X, Y)$, which cannot be explicitly evaluated. Instead, we adopt a Gibbs sampling, or iterative stochastic hard assignment. The key quantity for this approach is the cluster membership probability for observation i given the data X, Y and the other cluster memberships Z_{-i},

$$\begin{aligned} f(z_i = k|X, Y, Z_{-i}) &\propto f(Y|X, Z_{-i}, z_i = k) f(z_i = k|X, Z_{-i}) \\ &= \left(f_k(Y_{\mathcal{P}_k \cup \{i\}}|X_{\mathcal{P}_k \cup \{i\}}; \theta_k) \prod_{j \neq k} f_j(Y_{\mathcal{P}_j \setminus \{i\}}|X_{\mathcal{P}_j \setminus \{i\}}; \theta_j) \right) g_k(x_i; \varphi_k). \end{aligned} \tag{3.10}$$

Despite our highly dependent situation, (3.10) can be calculated in a simple form as shown in Proposition 1. The proof is deferred to Supplementary Material S1.

Proposition 1. *Under the complete data likelihood given in* (2.7)*,*

$$f(z_i = k|X, Y, Z_{-i}) \propto \phi((y_i - \mu_k^*)/\sigma_k^*) g_k(x_i; \varphi_k), \quad \textit{where} \tag{3.11}$$

$$\begin{aligned} \mu_k^* &= \mu_k(x_i) + \Phi_{\gamma_k}(x_i, X_{\mathcal{P}_k\setminus\{i\}}) \Phi_{\gamma_k}(X_{\mathcal{P}_k\setminus\{i\}}, X_{\mathcal{P}_k\setminus\{i\}})^{-1} \left(Y_{\mathcal{P}_k\setminus\{i\}} - \mu_k(X_{\mathcal{P}_k\setminus\{i\}})\right), \\ (\sigma_k^*)^2 &= \sigma_k^2 \left(1 - \Phi_{\gamma_k}(x_i, X_{\mathcal{P}_k\setminus\{i\}}) \Phi_{\gamma_k}(X_{\mathcal{P}_k\setminus\{i\}}, X_{\mathcal{P}_k\setminus\{i\}})^{-1} \Phi_{\gamma_k}(X_{\mathcal{P}_k\setminus\{i\}}, x_i)\right), \end{aligned} \tag{3.12}$$

where ϕ is the density probability function of a standard normal distribution.

Proposition 1 implies the cluster is assigned very intuitively. For an unknown predictive location x_i, the predictive distribution of each cluster k is a normal distribution with mean μ_k^* and variance $(\sigma_k^*)^2$ as in (2.2) and (2.3). Thus, the membership of z_i can be determined from the probability density function of cluster k at y_i, and the probability mass function g_k of membership k at x_i. The membership is likely to be assigned to kth class if (a) y_i is closer to μ_k^* with regard to the scale σ_k^*; (b) g_k has a high mass probability at location x_i.

Once (3.11) is available for each i and k, a random cluster assignment can be drawn from a multinomial distribution. Each step of this Gibbs scheme satisfies detailed balance (assuming none of the probabilities/densities in (3.11) equal zero), so eventually this process produces samples from $f(Z|X, Y)$. Hence, the cluster membership samples can be used to approximate quantities depending on $f(Z|X, Y)$, such as the expectation in (3.8). Further, partitioned matrix inverse and determinant formulas (Harville, 1998) allow one to update the augmented and diminished Gaussian densities in $O(n_k^2)$ time, where n_k is the number of observations in cluster k. The details are provided in Supplementary Material S2. In total, each iteration going through

all the observations would take at most $O(\sum_{k=1}^{K} n_k^3)$. One may ease computational burden by controlling the maximum number of observations in each cluster, denoted by $n_{\max}$, then the total computation becomes $O(Kn_{\max}^3)$. Computation in this step can be easily distributed over multiple cores, in particular, (3.12) can be done separately for different k. The detailed algorithm is given in Stochastic E-step of Supplementary Material S3.

3.2 M-step

Once a random assignment drawn from $\tilde{\mathcal{P}}_k = \{i : \tilde{z}_i = k\}$ is available from the stochastic E-step, we can proceed to the M-step. Let $\tilde{Z}$ denote the random assignment, and $\tilde{\mathcal{P}}_k = \{i : \tilde{z}_i = k\}$ the set of indices of the observations in cluster k assigned in $\tilde{Z}$, respectively. From (2.7) and (3.9), the log posterior of complete data in (3.8) is approximately by

$$\begin{aligned}
&\log f(Y, \tilde{Z}|X, \boldsymbol{\theta}, \boldsymbol{\varphi}) + \log \pi(\boldsymbol{\theta}) + \log \pi(\boldsymbol{\varphi}) \\
=&\sum_{k=1}^{K} \log f_k(Y_{\tilde{\mathcal{P}}_k}|X_{\tilde{\mathcal{P}}_k}; \theta_k) + \sum_{k=1}^{K} \sum_{i \in \tilde{\mathcal{P}}_k} \log g_k(x_i; \varphi_k) + \sum_{k=1}^{K} \log \pi(\theta_k) + \sum_{k=1}^{K} \log \pi(\varphi_k).
\end{aligned}$$

The maximum a posteriori probability (MAP) estimate $\{\hat{\theta}_k\}_{k=1}^K$ and $\{\hat{\varphi}_k\}_{k=1}^K$ can then be obtained by maximizing

$$\sum_{k=1}^{K} \log \left(f_k(Y_{\tilde{\mathcal{P}}_k}|X_{\tilde{\mathcal{P}}_k}; \theta_k)\pi(\theta_k) \right) \quad \text{and} \quad \sum_{k=1}^{K} \left(\sum_{i \in \tilde{\mathcal{P}}_k} \log g_k(x_i; \varphi_k) + \log \pi(\varphi_k) \right),$$

respectively. In particular, $\sum_{k=1}^{K} \log \left(f_k(Y_{\tilde{\mathcal{P}}_k}|X_{\tilde{\mathcal{P}}_k}; \theta_k)\pi(\theta_k) \right)$ can be optimized by maximizing each component $f_k(Y_{\tilde{\mathcal{P}}_k}|X_{\tilde{\mathcal{P}}_k}; \theta_k)\pi(\theta_k)$, which is proportional to the posterior distribution of the

k-th GP. The choice for the prior of θ_k and its resulting posterior can be found in Chapters 3 and 4 of Santner et al. (2018). The computation for M-step can be done for K clusters separately, which can be efficiently parallelized as in Supplementary Material S3.

3.3 Prediction

Predicting the responses y_{new} at a new input location x_{new} can be challenging, since the cluster assignment z_{new} at the new location is unknown. Given the assignment $\tilde{Z} = (\tilde{z}(x_1), \ldots, \tilde{z}(x_n))$ and the estimates $\{\hat{\theta}_k, \hat{\varphi}_k\}_{k=1}^K$ returned in the SEM algorithm, we perform the predictive distribution of y_{new} by weighted averaging across the clustered GPs:

$$\begin{aligned} f(y_{\text{new}}|x_{\text{new}}, X, Y, \tilde{Z}) &= \sum_{k=1}^{K} f(y_{\text{new}}|z_{\text{new}} = k, x_{\text{new}}, X, Y, \tilde{Z}) f(z_{\text{new}} = k|x_{\text{new}}, X, Y, \tilde{Z}) \\ &= \sum_{k=1}^{K} \phi((y_{\text{new}} - \hat{\mu}_k^*)/\hat{\sigma}_k^*) g_k(x_{\text{new}}; \hat{\varphi}_k), \end{aligned}$$

where

$$\begin{aligned} \hat{\mu}_k^* &= \hat{\mu}_k(x_{\text{new}}) + \Phi_{\hat{\gamma}_k}(x_{\text{new}}, X_{\tilde{\mathcal{P}}_k}) \Phi_{\hat{\gamma}_k}(X_{\tilde{\mathcal{P}}_k}, X_{\tilde{\mathcal{P}}_k})^{-1} \left(Y_{\tilde{\mathcal{P}}_k} - \hat{\mu}_k(X_{\tilde{\mathcal{P}}_k})\right), \\ (\hat{\sigma}_k^*)^2 &= \hat{\sigma}_k^2 \left(1 - \Phi_{\hat{\gamma}_k}(x_{\text{new}}, X_{\tilde{\mathcal{P}}_k}) \Phi_{\hat{\gamma}_k}(X_{\tilde{\mathcal{P}}_k}, X_{\tilde{\mathcal{P}}_k})^{-1} \Phi_{\hat{\gamma}_k}(X_{\tilde{\mathcal{P}}_k}, x_{\text{new}})\right). \end{aligned}$$

Thus, the prediction mean of y_{new} is

$$\hat{y}_{\text{new}} := \mathbb{E}[y_{\text{new}}|x_{\text{new}}, X, Y, \tilde{Z}] = \sum_{k=1}^{K} \hat{\mu}_k^* g_k(x_{\text{new}}; \hat{\varphi}_k), \tag{3.13}$$

with its variance

$$\begin{aligned}\mathbb{V}[y_{\text{new}}|x_{\text{new}}, X, Y, \tilde{Z}] =& \mathbb{E}[\mathbb{V}[y_{\text{new}}|z_{\text{new}}, x_{\text{new}}, X, Y, \tilde{Z}]] + \mathbb{V}[\mathbb{E}[y_{\text{new}}|z_{\text{new}}, x_{\text{new}}, X, Y, \tilde{Z}]] \\ =& \sum_{k=1}^{K} (\hat{\sigma}_k^*)^2 g_k(x_{\text{new}}; \hat{\varphi}_k) + \sum_{k=1}^{K} (\hat{\mu}_k^*)^2 g_k(x_{\text{new}}; \hat{\varphi}_k) - \left(\sum_{k=1}^{K} \hat{\mu}_k^* g_k(x_{\text{new}}; \hat{\varphi}_k) \right)^2.\end{aligned}$$

The q-th quantile of y_{new}, which will be used for constructing confidence intervals, has no closed form but can be calculated by finding the value of y for which $\int_{-\infty}^{y} f(t|x_{\text{new}}, X, Y, \tilde{Z})\mathrm{d}t = q$, which is equivalent to solving

$$\sum_{k=1}^{K} \left(\int_{-\infty}^{y} \phi((t - \hat{\mu}_k^*)/\hat{\sigma}_k^*)\mathrm{d}t \right) g_k(x_{\text{new}}; \hat{\varphi}_k) = q.$$

The summation and integration are interchangeable because the probability density function is finite. The equation can be solved numerically, for example, using a line search or generating Monte Carlo samples.

4. Computational details

In this section, we provide some computational details for the proposed SEM that we have provided in Section 3. In particular, we discuss the possible choices of each element in the algorithm, with the focus on the specific implementation that we have adopted.

4.1 Choices for class assignment model

The model for $z(\cdot)$ in (2.4) determines the latent class distribution of the cluster assignment, where g_k is the conditional probability that $z(x) = k$ given an input x. The function g_k determines the decision boundaries between the clusters, and their flexibility controls the bias-variance trade-off of the clustered GP. Amongst several possibilities for $z(\cdot)$, one may consider a less flexible model because GP itself is fairly flexible. For example, K-class multinomial logistic regression, which produces linear decision boundaries, can be considered. Then overall complexity and flexibility of the clustered GP can be determined by carefully selecting the number of clusters, which will be described in Section 4.4. The simple decision boundaries are useful for interpreting the clusters, which will be illustrated in Sections 5 and 6. The K-class multinomial logistic regression has the form of

$$\Pr(z(x) = k) = g_k(x; \varphi_k) = \frac{\exp\{\beta_{0,k} + \beta_k^T x\}}{\sum_{j=1}^{K} \exp\{\beta_{0,j} + \beta_j^T x\}},$$

for $k = 1, \ldots, K-1$ and $\Pr(z(x) = K) = 1 - \sum_{j=1}^{K-1} \Pr(z(x) = j)$, where $\beta_{0,k}$ is the intercept, β_k is a d-dimensional coefficient of x, and $\varphi_k = (\beta_1, \ldots, \beta_{K-1})$. Alternatively, one can also consider the linear discriminant analysis (LDA) or quadratic discriminant analysis (QDA) methods by assuming

$$g_k(x; \varphi_k) = \phi(x; \nu_k, \Sigma_k) \quad \text{for} \quad k = 1, \ldots, K,$$

where $\phi(x; \nu_k, \Sigma_k)$ is the density probability function of a (multivariate) normal distribution with mean ν_k and covariance Σ_k. LDA assumes $\Sigma_1 = \ldots = \Sigma_K$, while QDA assumes the covariances can be different. The multinomial logistic regression and LDA methods are closely connected, which often result in similar linear decision boundaries of the K classes. QDA methods, on the other hand, result in quadratic decision boundaries. From our preliminary investigation, the clustered GP with these models give similar prediction results. It is also possible to apply non-parametric or machine learning approaches, such as random forest classification, for modeling g_k. However, our preliminary investigation shows that these approaches have similar prediction performance, and they tend to result in less interpretable clusters in low-dimensional settings. This is because clustered GP's main advantage is from combining flexibility of GP assisted by the cluster structure, so g_k of an excessively complex form may not help much. As such, we only present K-class multinomial logistic regression hereinafter.

4.2 Initialization

The SEM algorithm can be sensitive to the initialization. One may run many initializations and select the one that gives the optimal criterion. This is, however, computational intensive especially for large data sets. One potential initialization is the K-means clusters or other unsupervised clustering algorithms solely based on the input X. This initialization enables the clustered GP to make the input locations of each cluster close to each other and distant from the ones of other clusters, which often leads to nice model interpretation. Although this initialization may end up with a local optimum, the cluster structure still further improves the model performance by efficiently exchanging the class assignment over the iterations. As such, in

Sections 5 and 6, the initialization of K-means clusters are used.

4.3 Stopping criteria

The iteration in the SEM algorithm in Supplementary Material S3 needs a stopping criterion to determine a convergence. For this purpose, we propose to use leave-one-out cross-validation (LOOCV), so that the algorithm stops when the cross-validated prediction error does not improve. LOOCV iteratively holds out one particular location, trains on the remaining data at other locations, and then makes prediction for the held-out location. Although LOOCV is often too expensive to implement in many situations as the model has to fit n times in each iteration, the clustered GP has an efficient shortcut that makes the LOOCV very affordable. Specifically, denote $\tilde{y}_i$ as the prediction mean based on all data except i-th observation and y_i as the real output of i-th observation, then based on (3.13), $\tilde{y}_i = \sum_{k=1}^{K} \hat{\mu}_k^{(-i)} g_k(x_i; \hat{\varphi}_k)$, where

$$\hat{\mu}_k^{(-i)} = \hat{\mu}_k(x_i) + \Phi_{\hat{\gamma}_k}(x_i, X_{\tilde{\mathcal{P}}_k \setminus \{i\}}) \Phi_{\hat{\gamma}_k}(X_{\tilde{\mathcal{P}}_k \setminus \{i\}}, X_{\tilde{\mathcal{P}}_k \setminus \{i\}})^{-1} \left(Y_{\tilde{\mathcal{P}}_k \setminus \{i\}} - \hat{\mu}_k(X_{\tilde{\mathcal{P}}_k \setminus \{i\}}) \right).$$

For those is which do not belong to $\tilde{\mathcal{P}}_k$, $\hat{\mu}_k^{(-i)}$ becomes

$$\hat{\mu}_k^{(-i)} = \hat{\mu}_k(x_i) + \Phi_{\hat{\gamma}_k}(x_i, X_{\tilde{\mathcal{P}}_k}) \Phi_{\hat{\gamma}_k}(X_{\tilde{\mathcal{P}}_k}, X_{\tilde{\mathcal{P}}_k})^{-1} \left(Y_{\tilde{\mathcal{P}}_k} - \hat{\mu}_k(X_{\tilde{\mathcal{P}}_k}) \right),$$

and for those is which belong to $\tilde{\mathcal{P}}_k$, it can be simplified to

$$\hat{\mu}_k^{(-i)} = \hat{\mu}_k(x_i) - \frac{1}{q_{ii}} \sum_{j \neq i}^{n_k} q_{ij} (y_j - \hat{\mu}_k(x_j)), \tag{4.14}$$

where q_{ij} is the (i,j)-th element of $\Phi_{\hat{\gamma}_k}(X_{\tilde{\mathcal{P}}_k}, X_{\tilde{\mathcal{P}}_k})^{-1}$. Then, the LOOCV root-mean-squared error (RMSE) is

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \sum_{k=1}^{K}\hat{\mu}_k^{(-i)} g_k(x_i;\hat{\varphi}_k)\right)^2}.$$

This computation costs at most $O(Kn_{\max}^3)$, which is same as the SEM algorithm.

4.4 The choice of K

The number of clusters K plays an important role for the degree of non-stationarity of approximation functions and flexibility of the model, which in turn controls the bias-variance trade-off of the model that can affect the prediction accuracy. That is, a too large K could lead to an over-flexible model and a too small K could lead to an under-flexible model. A natural choice is using cross-validation with different K's to target a small prediction error, such as the LOOCV RMSE described in Section 4.3. Other choices using bootstrap techniques to estimate prediction error also can be considered, such as the 632+ bootstrap method of Efron and Tibshirani (1997). Kohavi (1995) explicitly discussed the comparison between cross-validation and bootstrap from bias and variance point of view, and comprehensive numerical experiments were conducted therein. For the purpose of saving computational cost, we choose the K that gives the lowest LOOCV RMSE, because LOOCV RMSE can be computed efficiently for clustered GPs as given in (4.14).

4.5 Remarks on alternative implementations and asymptotic properties

The SEM and prediction can be modified in a more fully Bayesian fashion using the Monte Carlo samples from the posterior distribution of $\{z(x_i)\}_{i=1}^n$, $\{\theta_k, \varphi_k\}_{k=1}^K$ with a Gibbs routine to generate predictions. The computational burden for this direction, however, can be prohibitively heavy in a large-data context. In particular, saving samples from the posteriors requires enormous amounts of storage for large data sets. Using the returned assignment $\tilde{Z}$ and the MAPs $\{\hat{\theta}_k, \hat{\varphi}_k\}_{k=1}^K$ can be an efficient alternative with representative samples for more efficient fitting and prediction procedures.

The MAP estimation in the M-step can be replaced by maximum likelihood (ML) estimation, simply by letting the prior distributions of $\{\theta_k\}_{k=1}^K$ and $\{\varphi_k\}_{k=1}^K$ be uniform. Under some regularity conditions, the ML estimators $\{\hat{\theta}_k\}_{k=1}^K$ and $\{\hat{\varphi}_k\}_{k=1}^K$ can be shown to have an asymptotically normal distribution in such approach. We refer the asymptotic properties of the parameter inference to Nielsen (2000).

5. Numerical study

In this section we present several exemplar functions to demonstrate the effectiveness of clustered Gaussian processes. We first present examples with lower dimensional inputs to visually present the cluster structure and the benefit from non-stationary modeling and then to an example with higher-dimension inputs. Throughout, the iteration in the SEM algorithm stops when LOOCV does not improve, or the number of iterations exceeds the preset maximum. We select the assignment $\tilde{Z}$ which results in the lowest LOOCV RMSE during the iterations, which will be illustrated in Section 5.2. Power correlation function of (2.1) with $p = 2$ is chosen. Both of

the mean functions $\mu(\cdot)$ and $\mu_k(\cdot)$ of the stationary GP and the clustered GP are assumed to be constant. For each cluster, a small nugget, 10^{-6}, is added when fitting a GP model for numerical stability. In addition, we let the prior distributions of $\{\theta_k\}_{k=1}^K$ and $\{\varphi_k\}_{k=1}^K$ be uniform.

5.1 One-dimensional synthetic data

Consider an example from Gramacy and Lee (2009), which is a modification of the example in Higdon (2002). Suppose that the true function is

$$f(x) = \begin{cases} \sin(0.2\pi x) + 0.2\cos(0.8\pi x), & \text{if } x < 10. \\ 0.1x - 1, & \text{otherwise} \end{cases}$$

and 11 unequally spaced points from $[0, 20]$ are chosen. The black lines in Figure 2 demonstrate this function, and it can be seen that the function is discontinuous at $x = 10$. When the data are modeled by a stationary GP, it can be seen in the left panel of Figure 2 that the prediction within region $[10, 20]$ performs poorly with large uncertainty. Ba and Joseph (2012) explained that the constant mean assumption for GP is violated so the predictor tends to revert to the global mean, whose estimate is 0.208 by maximum likelihood estimation in this example. This consequence is frequently observed especially at the locations far away from input locations. Moreover, the constant variance assumption for GP is also violated. The function in the region $[0, 10]$ is rougher than that in the region $[10, 20]$. Therefore, the variance estimate for region $[10, 20]$ tends to be inflated by averaging with that of region $[0, 10]$, which leads to the erratic prediction in this region. On the other hand, clustered GP introduces some degree of non-stationarity by

considering a mixture GP, which is shown in the right panel of Figure 2. Two subsets of the data are represented as red and green dots, which are given by the assignment $\tilde{Z}$ returned in the SEM algorithm, and both are fitted by stationary GPs. The mean estimates of the GPs are -0.045 and 0.529, respectively. It can be seen that the predictor performs much better than a stationary GP, especially at the locations within region $[10, 20]$, in terms of prediction accuracy and uncertainty quantification. The most uncertain region is located on the boundary of two clusters, which is expected because the assignment of cluster membership is more uncertain in the region. One potential remedy of improving the accuracy on the boundaries will be discussed in Section 7. The middle panel illustrates the composite GP of Ba and Joseph (2012), which is a popular method in the computer experiment literature for addressing the non-stationary issue. It shows that the prediction and uncertainty quantification are more accurate than the stationary GP, but less accurate than the clustered GP.

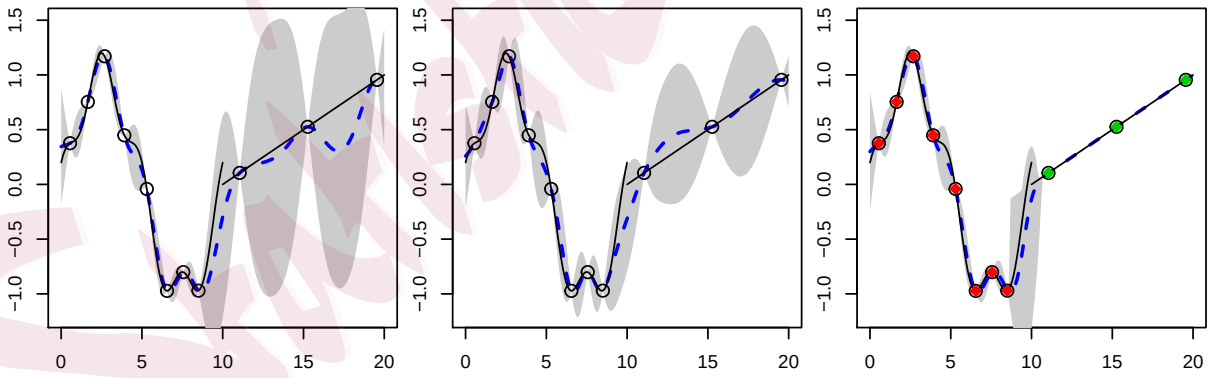

Figure 2: One-dimensional synthetic data. The left, middle and right panels illustrate the predictors by the stationary GP, the composite GP (Ba and Joseph, 2012), and the clustered GP, respectively. Black line is the true function, black circles are input locations, and blue dotted lines are the predictors, with the gray shaded region providing a pointwise 95% confidence band. Red and green dots in the right panels represent different clusters.

Two more one-dimensional synthetic data generated from the exemplar functions of Xiong

et al. (2007) and Montagna and Tokdar (2016) are presented in Supplementary Material S4, in which both examples show that the clustered GP yields better prediction accuracy than the stationary GP and the composite GP.

5.2 Two-dimensional synthetic data

In this section, the selection of K and the stopping rule using LOOCV RMSE will be demonstrated. Consider a wavy function, which also appeared in Ba and Joseph (2012) and Montagna and Tokdar (2016). The wavy function is

$$f(x_1, x_2) = \sin\left(\frac{1}{x_1 x_2}\right),$$

where $x_1, x_2 \in [0.3, 1]$. The function is illustrated in Figure 3(a), in which it fluctuates rapidly when x_1 and x_2 are small and gets smoother as they increase toward 1. A 40-run maximin distance Latin hypercube design (Morris and Mitchell, 1995) from $[0.3, 1]^2$ is chosen to select the input locations at which the wavy function is evaluated. These locations are shown as black dots. The stationary GP, the composite GP (Ba and Joseph, 2012), and the clustered GP with $K = 3$ are fit on these locations, whose predictive surfaces are shown in Figures 3(b-d). It can be seen that the stationary GP and the composite GP performs fairly poorly as x_1 and x_2 are small, while the clustered GP generally has better prediction performance over the input space. To evaluate the prediction performance quantitatively, we predict the responses at 1296 ($= 36 \times 36$) equally spaced points from $[0.3, 1]^2$ as the test points, and compute their RMSEs

by

$$\left(\frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \left(f(x_1, x_2) - \hat{f}(x_1, x_2) \right)^2 \right)^{1/2},$$

where n_{test} is the number of test points and $\hat{f}(x_1, x_2)$ is the predicted value at x_1 and x_2. In this example, the clustered GP outperforms the composite GP and the stationary GP in terms of prediction accuracy, where their RMSEs are 0.2081, 0.2284 and 0.3959, respectively, and the interval scores of their 95% prediction intervals (see equation (43) in Gneiting and Raftery (2007)) are 0.6950, 0.9635 and 2.0915 (the lower the better).

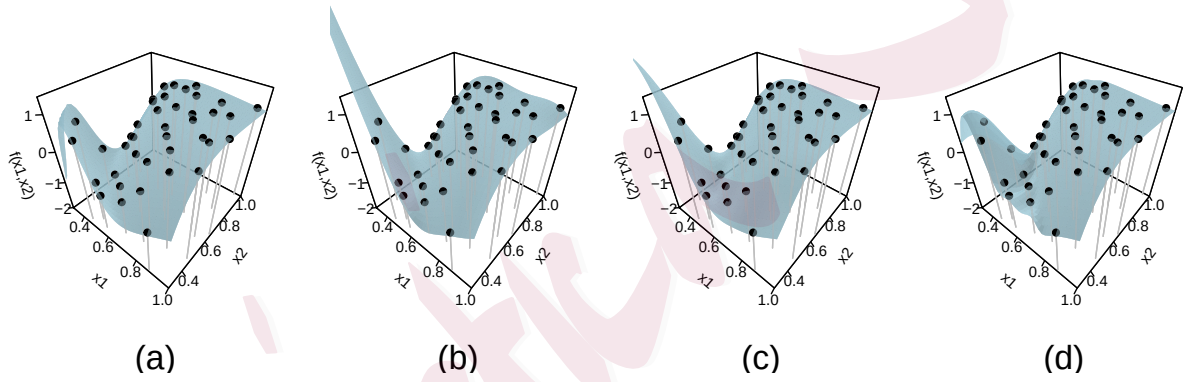

Figure 3: Two-dimensional example: (a) the true wavy function, (b) the stationary GP, (c) the composite GP (Ba and Joseph, 2012), and (d) the clustered GP, where the input locations are shown as black dots.

Figure 4 demonstrates the stopping rule and the selection of K discussed in Section 4. The left panel presents the LOOCV RMSEs of $K = 2, 3, 4$ and 5 during the 100 iterations of the SEM algorithm. It shows that even though the LOOCV RMSE of initial iteration of $K = 3$ is larger than other choices of K, the error drops rapidly and ends up with a lower LOOCV error at 36-th iteration. For each choice of K, we chose the assignment of the iteration that results in the minimum LOOCV RMSE as the final assignment $\tilde{Z}$ for prediction. The right panel presents the

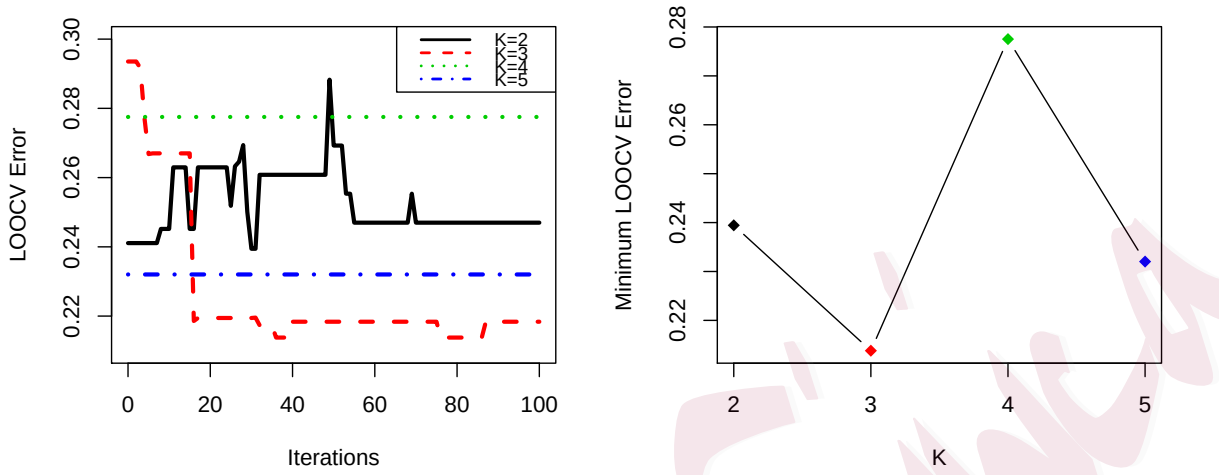

Figure 4: The LOOCV RMSEs with $K = 2, 3, 4$ and 5 during the 100 iteration of the SEM algorithm (left), and the minimum LOOCV RMSE of each choice of K (right).

minimum LOOCV RMSE of each choice of K in the 100 iterations, and it shows that $K = 3$ gives the lowest LOOCV RMSE so it was selected in this example. Figure 5 demonstrates the assignments at iteration 0, 4, and 36 when $K = 3$. The assignment at iteration 0 represents initial assignment, which is the K-means clusters as described in Section 4.2, whose LOOCV RMSE is 0.294. The LOOCV RMSE then drops dramatically in the 4-th iteration from 0.294 to 0.277 with only two assignments switched, that is, the point $x_1 = 0.726, x_2 = 0.482$ is from circle to square cluster and the point $x_1 = 0.702, x_2 = 0.866$ is from triangle to square cluster. With more iterations and more assignments switched, the LOOCV error decreases to 0.214 at iteration 36. The final assignment gives an intuitive explanation: the points when both of x_1 and x_2 are small, where the true function has a sharp change, appear to belong to the same cluster (see the circle cluster). To demonstrate the advantage of the clustering in terms of prediction accuracy, we further compare the true RMSE with an supervised learning approach, K-means clustering (left panel of Figure 5), whose RMSE is 0.2728, which is larger than the one of the

clustered GP, 0.2081. This shows that, when the goal is making predictions, our clustering that integrates output information can efficiently improve unsupervised learning clustering that does not make use the output information.

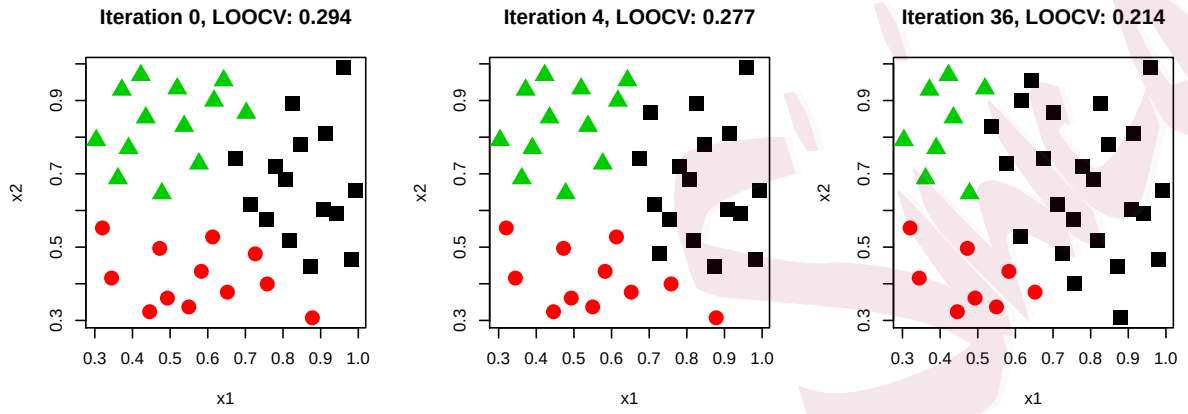

Figure 5: The cluster assignments at iteration 0, 4, and 36 of the SEM algorithm and their LOOCV RMSEs.

5.3 Borehole function

In the section, a borehole function, a more complex exemplar function with 8-dimensional input, is considered to examine the scalability of clustered GP. The borehole function models water flow through a borehole, and has been commonly used for testing methods in computer experiments because of its quick evaluation. The borehole function is given by

$$f(x) = \frac{2\pi T_u (H_u - H_l)}{\ln(r/r_w)\left(1 + \frac{2LT_u}{\ln(r/r_w) r_w^2 K_w} + \frac{T_u}{T_l}\right)}, \tag{5.15}$$

where $r_w, r, T_u, H_u, T_l, H_l, L$ and K_w are the eight inputs. We refer the detailed description of these input variable to Morris et al. (1993).

Consider n uniformly distributed input locations in the input space described above and $n_{\text{test}} = 10,000$ random input locations in the same input space for examining prediction accuracy, whose outputs are evaluated from (5.15). Four methods are compared, including a stationary GP, local GP (Gramacy and Apley, 2015), multi-resolution functional ANOVA (MRFA) (Sung et al., 2020), and clustered GP. These methods are implemented using `R` (R Core Team, 2015) via packages `mlegp` (Dancik, 2013), `laGP` (Gramacy, 2015), `MRFA` (Sung, 2019), `clusterGP`, on a MacBook Pro laptop with 2.6 GHz Intel Core i7 and 16GB of RAM. For the purpose of demonstration, $K = n/200$ was chosen for all the cases. For `laGP`, `MRFA` and `clusterGP`, 10 CPU threads were utilized via `foreach` (Revolution Analytics and Weston, 2015) for parallel computing.

Table S1 shows the performance of the four methods, in terms of computation time and prediction accuracy. It can be seen that the stationary GP is feasible only when $n = 1,000$, while other three methods can incorporates larger n. Even when a stationary GP is feasible, the accuracy is worse than `MRFA` and `clusterGP`. Among the four methods, `clusterGP` has better accuracy with reasonable computation time. `MRFA` has slightly larger predictive errors with faster computation. On the other hand, local GP has larger predictive errors, even though the computation is faster. One may consider a different setting for local GP (e.g., the size of subsample) which may lead to better accuracy. While the proposed method yields better prediction accuracy with reasonable prediction time, which is the main goal of emulation for computer simulations, the model fitting time and storage can be demanding particularly for very large-scale datasets. Some potential remedies of improving the computational efficiency will be discussed in Section 7.

6. Solar irradiance prediction

We leverage the statistical developments to predict solar irradiance. Predicting solar irradiance, or the power per unit area produced by electromagnetic radiation, plays a very important role in power balancing and determining the viability of potential sites for harvesting solar power. One dataset can be brought to bear on this problem is the simulations from the North American Mesoscale Forecast System (NAM) (Rogers et al., 2009), which is one of the major weather models run by the National Centers for Environmental Prediction (NCEP) for producing weather forecasts. We extract the solar irradiance (global horizontal irradiance) simulations from the NAM model at the locations of 1,535 Remote Automatic Weather Station (RAWS) (Zachariassen et al., 2003) sites in the contiguous United States. Note that the RAWS stations are not uniformly distributed. Figure S2 visualizes the available locations and their corresponding solar irradiance with the average taken over one year, which can be seen that many promising locations for solar farms are sparsely covered particularly in the Midwest. These locations of interest are considered for solar energy forecasting. Detail description of the dataset can be found in Hwang et al. (2018) and Sun et al. (2019b). Similar to Sun et al. (2019b), here we work with average irradiance values over one year from the NAM simulations for each of 1535 spatial locations (as shown in Figure S2), and the research interest of this study is making accurate prediction for solar irradiance at those unavailable locations.

In Figure S2, it appears that some relatively high solar irradiance are measured compared to their neighborhood, such as at the location on the coordinate $(-93.57, 45.99)$, and some relatively low solar irradiance are measured such as at the location on the coordinate $(-93.16, 33.69)$.

These instances may suggest that heterogeneity rather than homogeneity in the input-output relationships should be considered. The assumption of identical covariance function throughout the input domain for stationary GPs, therefore, is likely to fail and may result in poor performance, as shown in Section 5.

A clustered GP is performed on this dataset, where similar setup in Section 5.2 was used. We first use the LOOCV to determine the number of clusters K. The left panel of Figure S3 shows the LOOCV RMSEs of $K = 15, 25, 35, 45$ during 20 iterations of the SEM algorithm, and the right panel shows the minimum LOOCV RMSEs with respect to different choices of K. Based on the right panel, it appears that $K = 35$ has the lowest LOOCV RMSE among $K = 10, 15, 20, 25, 30, 35, 40, 45, 50$, which suggests that $K = 35$ is a good choice for predicting solar irradiance. Similar to the numerical study in Section 5, we chose the assignment of the iteration which results in the lowest LOOCV RMSE as the final assignment $\tilde{Z}$. The assignment $\tilde{Z}$ is visualized in Figure S4, where the 35 clusters are presented as different colors and numbers. It appears that the clusters reveal interesting hidden patterns in the input-output relationship. For example, cluster 26 are mostly located on Michigan and part of Pennsylvania and New York, which tells us that some common aspects of the solar irradiance are shared in those areas adjacent to Great Lakes, even though they are not spatially connected. The example shows that the clustering can provide a useful insight for discovering groups and identifying interesting insight of a dataset.

To examine its prediction accuracy, we use LOOCV RMSEs as the prediction error and compare with a recent emulation method in Sun et al. (2019a), where they proposed a multi-resolution global/local GP emulation by extending the idea of local GP (Gramacy and Apley,

2015), and their latter work in Sun et al. (2019b) applied this method to the same NAM simulation data herein. Sun et al. (2019b) reported the LOOCV errors of the multi-resolution global/local GP emulation as well as the ordinary stationary GP. The results together with our proposed method are presented in Figure S5. The figure presents the true solar irradiance (top left) and the LOOCV predictions of the stationary GP (top right), the multi-resolution global/local GP (bottom left), and the clustered GP with $K = 35$ (bottom right), along with their corresponding LOOCV RMSEs in the titles. It can be seen that, the stationary GP does a poor job in predicting the solar irradiance, the LOOCV predictions of which are all essentially equal which implies that almost all of the pattern remains in the errors, which in turn gives a high LOOCV RMSE (23.20). Performances of the multi-resolution global/local GP as well as the clustered GP on the other hand are very good, the result of which may suggest that the non-stationarity should be taken into account for this dataset. Although the LOOCV predictions are visually similar, the LOOCV RMSE of the clustered GP is slightly lower than the multi-resolution global/local GP (9.11 and 9.74, respectively). In particular, it appears that the clustered GP has better prediction accuracy in the Northeast and Southeast, whereas the multi-resolution global/local GP tends to be more smooth over the whole space.

7. Discussion

In this paper, we proposed a clustered Gaussian process that can simultaneously reduce computational burden and incorporate non-stationarity, which effectively address two of major limitations of stationary GP. Unlike traditional unsupervised clustering methods, the clusters in the clustered GP are *supervised* by the response - the clustered GP makes use of the response in

order to partition the input domain that not only clusters the observations that have similar features, but also that have the same stationary process in the response. This clustering algorithm is implemented using a stochastic EM algorithm, which is available in an open repository. Examples including the application of solar irradiance simulations show that the method not only has advantages in computation and prediction accuracy, but also enables discovery of interesting insights by interpreting the clusters.

The clustered GP shows several avenues for future research. First, the stochastic EM algorithm can be modified in an online fashion. That is, if the data is available in a sequential order, then the algorithm can be modified to update the clusters and the best predictor for future data at each step instead of starting from the new dataset augmented with the additional data. For example, the solar irradiance simulations are available every hour, so a modified algorithm could be used to update the clusters and predict future data in real time, which may save substantial computational cost and storage especially when the training sample size is extremely large. In addition to the online stochastic EM, sub-sampling methods can be naturally applied to the clustered GP that can alleviate the storage limitations for large-scale data. The CURE algorithm (Guha et al., 2001) provides an efficient way for large-scale datasets for traditional clustering algorithms, which employs a combination of random sampling and partitioning. It is conceivable to apply this technique to the our clustering algorithm. Moreover, the flexible structure of the proposed model can be easily generalized to other applications in computer experiments. For instance, although the focus of this paper is on the emulation for deterministic computer simulations, the proposed method can be naturally applied to stochastic computer simulations by including a nugget term or heteroscedastic variance function (Ankenman et al., 2010; Binois

et al., 2018) in each of the GPs. Last but not the least, to reduce the prediction uncertainty on the boundary between two regions (see, for example, $x = 10$ in Figure 2), it is conceivable to apply the idea of "patchwork" in Park and Apley (2018) by patching the GPs on the boundary, which can mitigate the discontinuous problem that may degrade the prediction accuracy. We leave these to our future work.

Supplementary Materials

The online supplementary materials contain the detailed proof of Proposition 1, the detailed SEM algorithm in Section 3, supporting tables and figures for Sections 5 and 6. An `R` package `GPcluster` for implementing the proposed method is available at `https://github.com/ChihLi/GPcluster`.

Acknowledgements

The authors gratefully acknowledge helpful advice from the associate editor, two anonymous referees. This work was partly supported by NSF DMS 2113407, and partly by National Center for Theoretical Sciences.

References

Ankenman, B., Nelson, B. L., and Staum, J. (2010). Stochastic kriging for simulation metamodeling. *Operations Research*, 58(2):371–382.

Ba, S. and Joseph, V. R. (2012). Composite Gaussian process models for emulating expensive functions. *The Annals of Applied Statistics*, 6(4):1838–1860.

Binois, M., Gramacy, R. B., and Ludkovski, M. (2018). Practical heteroscedastic Gaussian process modeling for large simulation experiments. *Journal of Computational and Graphical Statistics*, 27(4):808–821.

Bui-Thanh, T., Ghattas, O., and Higdon, D. (2012). Adaptive Hessian-based nonstationary Gaussian process response surface method for probability density approximation with application to bayesian solution of large-scale inverse problems. *SIAM Journal on Scientific Computing*, 34(6):A2837–A2871.

Cappé, O. and Moulines, E. (2009). On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B*, 71(3):593–613.

Celeux, G. and Diebolt, J. (1985). The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics Quarterly*, 2(1):73–82.

Chen, J., Zhu, J., Teh, Y. W., and Zhang, T. (2018). Stochastic expectation maximization with variance reduction. In *32nd Conference on Neural Information Processing Systems*, pages 7978–7988.

Chipman, H. A., George, E. I., and McCulloch, R. E. (1998). Bayesian CART model search. *Journal of the American Statistical Association*, 93(443):935–948.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2002). Bayesian treed models. *Machine Learning*, 48(1):299–320.

Dancik, G. M. (2013). *mlegp: Maximum Likelihood Estimates of Gaussian Processes*. R package version 3.1.4.

Efron, B. and Tibshirani, R. (1997). Improvements on cross-validation: the 632+ bootstrap method. *Journal of the American Statistical Association*, 92(438):548–560.

Fang, K.-T., Li, R., and Sudjianto, A. (2005). *Design and Modeling for Computer Experiments*. CRC Press.

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, 97(458):611–631.

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

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378.

Gramacy, R. B. (2015). laGP: large-scale spatial modeling via local approximate Gaussian processes in R. *Journal of Statistical Software (available as a vignette in the laGP package)*.

Gramacy, R. B. (2020). *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*. CRC Press.

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

Gramacy, R. B. and Lee, H. K. H. (2008). Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119–1130.

Gramacy, R. B. and Lee, H. K. H. (2009). Adaptive design and analysis of supercomputer experiments. *Technometrics*, 51(2):130–145.

Guha, S., Rastogi, R., and Shim, K. (2001). Cure: an efficient clustering algorithm for large databases. *Information Systems*, 26(1):35–58.

Haaland, B. and Qian, P. Z. G. (2011). Accurate emulators for large-scale computer experiments. *The Annals of Statistics*, 39(6):2974–3002.

Harville, D. A. (1998). *Matrix Algebra from a Statistician's Perspective*. Springer, New York, NY.

Higdon, D. (2002). Space and space-time modeling using process convolutions. *Quantitative Methods for Current Environmental Issues*, pages 37–56.

Higdon, D., Swall, J., and Kern, J. (1999). Non-stationary spatial modeling. *Bayesian Statistics*, 6(1):761–768.

Hwang, Y., Lu, S., and Kim, J.-K. (2018). Bottom-up estimation and top-down prediction: Solar energy prediction combining information from multiple sources. *Annals of Applied Statistics*, 12(4):2096–2120.

Joseph, V. R. and Mak, S. (2021). Supervised compression of big data. *Statistical Analysis and Data Mining*, 14(3):217–229.

Kim, H.-C. and Lee, J. (2007). Clustering based on Gaussian processes. *Neural computation*, 19(11):3088–3107.

Kim, H.-M., Mallick, B. K., and Holmes, C. C. (2005). Analyzing nonstationary spatial data using piecewise Gaussian processes. *Journal of the American Statistical Association*, 100(470):653–668.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1137–1145.

Montagna, S. and Tokdar, S. T. (2016). Computer emulation with nonstationary Gaussian processes. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):26–47.

Morris, M. D. and Mitchell, T. J. (1995). Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference*, 43(3):381–402.

Morris, M. D., Mitchell, T. J., and Ylvisaker, D. (1993). Bayesian design and analysis of computer experiments: use of derivatives in surface prediction. *Technometrics*, 35(3):243–255.

Nguyen-Tuong, D. and Peters, J. (2011). Model learning for robot control: a survey. *Cognitive processing*, 12(4):319–340.

Nguyen-Tuong, D., Peters, J., and Seeger, M. (2009). Local Gaussian process regression for real time online model learning. In *Advances in Neural Information Processing Systems 21*, pages 1193–1200.

Nielsen, S. F. (2000). The stochastic EM algorithm: estimation and asymptotic results. *Bernoulli*, 6(3):457–489.

Nychka, D., Bandyopadhyay, S., Hammerling, D., Lindgren, F., and Sain, S. (2015). A multi-resolution Gaussian process model for the analysis of large spatial data sets. *Journal of Computational and Graphical Statistics*, 24(2):579–599.

Paciorek, C. J. and Schervish, M. J. (2006). Spatial modelling using a new class of nonstationary covariance functions. *Environmetrics*, 17(5):483–506.

Park, C. and Apley, D. (2018). Patchwork kriging for large-scale Gaussian process regression. *The Journal of Machine Learning Research*, 19(1):269–311.

Plagemann, C., Kersting, K., and Burgard, W. (2008). Nonstationary Gaussian process regression using point estimates of local smoothness. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 204–219. Springer.

Plumlee, M. (2014). Fast prediction of deterministic functions using sparse grid experimental designs. *Journal of the American Statistical Association*, 109(508):1581–1591.

Plumlee, M. and Apley, D. W. (2017). Lifted brownian kriging models. *Technometrics*, 59(2):165–177.

Quiñonero-Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959.

R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rasmussen, C. E. and Ghahramani, Z. (2002). Infinite mixtures of Gaussian process experts. In *Advances in neural information processing systems*, pages 881–888.

Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian processes for machine learning*, volume 1. MIT press Cambridge.

Revolution Analytics and Weston, S. (2015). *foreach: Provides Foreach Looping Construct for R*. R package version 1.4.3.

Rogers, E., DiMego, G., Black, T., Ek, M., Ferrier, B., Gayno, G., Janjic, Z., Lin, Y., Pyle, M., Wong, V., et al. (2009). The ncep north american mesoscale modeling system: Recent changes and future plans. In *23rd Conference on Weather Analysis and Forecasting/19th Conference on Numerical Weather Prediction, Omaha, NE*.

Sang, H. and Huang, J. Z. (2012). A full scale approximation of covariance functions for large spatial data sets. *Journal of the Royal Statistical Society: Series B*, 74(1):111–132.

Santner, T. J., Williams, B. J., and Notz, W. I. (2018). *The Design and Analysis of Computer Experiments*. Springer-Verlag New York, 2 edition.

Snelson, E. and Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, pages 1257–1264.

Stein, M. L. (2012). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Science & Business Media.

Sun, F., Gramacy, R. B., Haaland, B., Lawrence, E., and Walker, A. (2019a). Emulating satellite drag from large simulation experiments. *SIAM/ASA Journal on Uncertainty Quantification*.

Sun, F., Gramacy, R. B., Haaland, B., Lu, S., and Hwang, Y. (2019b). Synthesizing simulation and field data of solar irradiance. *Statistical Analysis and Data Mining*, 12(4):311–324.

Sung, C.-L. (2019). *MRFA: Fitting and Predicting Large-Scale Nonlinear Regression Problems using Multi-Resolution Functional ANOVA (MRFA) Approach*. R package version 0.4.

Sung, C.-L., Wang, W., Plumlee, M., and Haaland, B. (2020). Multi-resolution functional ANOVA for large-scale, many-input computer experiments. *Journal of the American Statistical Association*, 115(530):908–919.

Titsias, M. (2009). Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574.

Tresp, V. (2001). Mixtures of Gaussian processes. In *Advances in neural information processing systems*, pages 654–660.

Xiong, Y., Chen, W., Apley, D., and Ding, X. (2007). A non-stationary covariance-based kriging method for metamodelling in engineering design. *International Journal for Numerical Methods in Engineering*, 71(6):733–756.

Zachariassen, J., Zeller, K. F., Nikolov, N., and McClelland, T. (2003). A review of the forest

service remote automated weather station (raws) network. *General Technical Report*. No. RMRS-GTR-119.

Zhang, Y., Ghosh, S., Asher, I., Ling, Y., and Wang, L. (2019). Learning uncertainty using clustering and local Gaussian process regression. In *AIAA Scitech 2019 Forum*, page 1730.

Department of Statistics and Probability 619 Red Cedar Rd, East Lansing, MI USA.

E-mail: sungchih@msu.edu

Department of Population Health Sciences 295 Chipeta Way Salt Lake City, UT 84108, USA.

E-mail: ben.haaland@hsc.utah.edu

Paul H. Chook Department of Information Systems and Statistics 55 Lexington Ave at 24th Street, New York, NY 10010, USA.

E-mail: Youngdeok.Hwang@baruch.cuny.edu

IBM Thomas J. Watson Research Center Yorktown Heights, New York 10598 USA, USA.

E-mail: lus@us.ibm.com