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

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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 is one of the most important approaches for emulating computer simulations. However, the stationarity assumption common to Gaussian process emulation and the computational intractability for large-scale data sets limit accuracy and feasibility in practice. In this article, we propose a clustered Gaussian process model that *simultaneously* segments the input data into multiple clusters and fits a Gaussian process model in each cluster. The model parameters and the clusters are learned through the efficient stochastic expectation-maximization, which allows for emulations 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 using cross-validation to ensure accurate predictions. In our simulations and 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 the data by discovering clusters. An `R` package for the proposed methodology is provided in an open repository.

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

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

1. Introduction

Gaussian processes (GPs) are popular modeling tools in various research areas, including 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). GPs provide flexibility for a prior probability distribution over functions in Bayesian inference, and the posterior can be used both to estimate the unknown function at an unknown point, and to quantify the 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, for the following reasons. First, the GP posterior involves $O(N^3)$ computational complexity and $O(N^2)$ storage, where N is the sample size, so that a GP emulation becomes infeasible for moderately large data sets, say $N = 10^3$. Second, a GP model often uses 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 consists of two different functions: a relatively rough function in the region $x \in [0, 10]$ and

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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 Higdon et al. (1999), Paciorek and Schervish (2006), and Bui-Thanh et al. (2012) for additional examples.

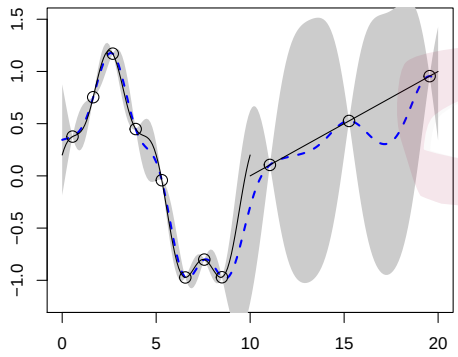

Figure 1: An example of stationary Gaussian process emulation applied to a nonstationary function. The solid line is the true function, and dots represent collected data. The dashed line represents a stationary Gaussian process emulator, with the shaded region providing a pointwise 95% confidence band.

These two challenges to GP modeling are common in practice, and thus have attracted much attention. The computational issue for large data sets is addressed by 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 designs (Plumlee, 2014), and multi-resolution approximation (Nychka et al., 2015), among others. For nonstationarity, Higdon et al. (1999), Higdon (2002), Paciorek and Schervish (2006), Plagemann et al. (2008),

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

and Plumlee and Apley (2017) adopted nonstationary covariance functions for GPs. Tresp (2001), Rasmussen and Ghahramani (2002), Kim et al. (2005), and Gramacy and Lee (2008) considered multiple GPs by segmenting in the input spaces. Ba and Joseph (2012) proposed a composite of two GPs that respectively capture a smooth global trend and local details. However, few of these works address the nonstationarity 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 GP approximation, which selects a small subsample to fit a GP model for each predictive location (Gramacy and Apley, 2015).

In this article, we propose a clustered GP to address the two challenges simultaneously. The clustered GP segments the input data into clusters using a hard-assignment clustering approach, and fits a GP to each cluster. This makes the computation more tractable for large-scale data sets, while retaining the mixture model structure to address the nonstationarity issue. Because latent variable models often suffer from computational difficulties, the stochastic expectation-maximization (SEM) algorithm (Celeux and Diebolt, 1985) is employed to learn the clusters efficiently. Although combining a mixture GP and the efficient SEM algorithm has been shown to potentially be able to simultaneously address the nonstationarity and computation challenges, this approach has not been studied

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

carefully. In addition, the number of clusters plays a crucial role for a mixture GP model, controlling the flexibility and the nonstationarity of the model, and thus a systematic criterion for selecting the tuning parameter is necessary; however, few works examine such a criterion. The cross-validation criterion, which retains efficient computation, is studied carefully here. Importantly, unlike many existing methods, the clustered GP retains the features of unsupervised learning approaches that reveal hidden patterns in the data, leading to interesting model interpretations and important insights about the underlying problem by showing grouping structures.

Traditional unsupervised learning, such as K-means clustering and the GP clustering of Kim and Lee (2007), aims to partition observations into groups based on their similarities in the input space, and does not use information contained in the output. In contrast, the main purpose of the clustered GP is to build a flexible model that can produce accurate predictions at new input locations, and the assignments to each cluster are determined by both inputs and outputs. The observations within each cluster share similar behavior in terms of input–output relationships, and 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.

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

In Section 3, we describe our estimation and prediction for fitting the clustered GP model using an SEM algorithm. 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. Section 7 concludes the article. Mathematical proofs are given in the Supplementary Material, and an `R` package, `GPcluster`, is provided in an open repository for practitioners to implement the methodology.

2. Clustered GP

2.1 Preliminary: GPs

We begin by briefly reviewing GPs. A GP is a stochastic process in which the 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

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

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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, that is, $\mu(\cdot) = \mu$, 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. This is called a *stationary* correlation function, which implies that the process $y(\cdot)$ is stationary, because $y(x_1), \ldots, y(x_L)$ 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 the power correlation function

$$\Phi_\gamma(x, x') = \exp\{-\|\gamma \odot (x - x')\|_2^p\}, \tag{2.1}$$

where p is often fixed to control the smoothness of the output surface, $\gamma = (\gamma_1, \ldots, \gamma_d)$ controls the decay of the correlation with respect to the distance be-

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

tween x and x' in each input coordinate, and $\odot$ denotes the element-wise product of two vectors. Hence, the parameters include $\mu(\cdot), \sigma^2$, and γ, and can be estimated using either a maximum likelihood estimation or a Bayesian estimation; see Fang et al. (2005), Rasmussen and Williams (2006), and Santner et al. (2018) for more details. Importantly, when the research interest is a prediction at an untried x_{new}, the response of which is denoted as y_{new}, the predictive distribution of y_{new} can be derived using 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 GP

In practice, we might expect the unknown function that we are trying to approximate to exhibit some degree of nonstationarity. A natural conceptual model that accounts for this is the mixture GP, where each component of the mixture acts as an approximately stationary model with high accuracy for a subset of the data.

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

That is,

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

where $\mu_k(\cdot), \sigma_k^2$, and Φ_{γ_k} are the mean function, variance, and stationary correlation function, respectively, of the kth GP, and $g_k(x, \varphi_k)$ is the probability that $z(x) = k$, with the unknown parameter φ_k satisfying $\sum_{k=1}^{K} g_k(x; \varphi_k) = 1$ for any x. In this model, $z(\cdot)$ plays the role of a latent function that assigns $y(\cdot)$ to one of the K GPs. These models introduce nonstationarity by assuming different parameters of the stationary correlation functions in each cluster, dependent on the input space. This allows for the local smoothness of the function of interest, whereas the conventional GP lacks the ability to adapt the smoothness in the function. This input-dependent smoothness is essential in applications such as geo-science, traffic simulations, and robotics (Plagemann et al., 2008). For example, modeling the solar irradiance in Section 6 requires dealing with a varying data density and accounting for the local smoothness being potentially dependent on the input locations, where discontinuities may arise at geographic features such as mountain ranges. Such features can help scientists discover interesting insights that differentiate these clusters.

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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 depends on the response values and locations of the other cluster members, in addition to the 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 the input location x_i, so that the (unobserved) cluster assignment likelihood is given by

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

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

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

is

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

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 similarities, their application is limited in large-scale data settings, owing to their costly MCMC sampling. Other works such as Nguyen-Tuong et al. (2009) and Zhang et al. (2019), choose assignments to clusters based on traditional unsupervised clustering methods, such as K-means clustering.

Our modeling approach belongs to the popular model-based clustering approach that uses latent variables within an expectation-maximization (EM) frame-

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

work (e.g., Fraley and Raftery, 2002). A likelihood-based EM approach to estimate the unknown parameters is, however, not straightforward, because strong dependencies between observations due to the GP correlation structure make computation difficult. One may want to compute the cluster probability $f(Z|X, Y)$, whether to implement the E-step in the EM algorithm (soft assignment), or to update the 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, even though each z_i is independent of the others. In the next section, we adopt an SEM algorithm to address this issue, and present the computational details associated with our approach.

3. Statistical Inference Using an SEM 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 an SEM algorithm (Celeux and Diebolt, 1985). The SEM algorithm is particularly suitable because it leads to a computationally efficient algorithm in a clustered GP, while avoiding the insignificant local maxima of the likelihood functions. The SEM presented here is a general approach for calculating the conditional expectation required in the E-step of the EM algorithm. In contrast, recent stud-

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

ies, such as Cappé and Moulines (2009) and Chen et al. (2018), focus 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, as in our case.

3.1 Stochastic E-step

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

$$\mathbb{E}[\log f(Y, Z|X)|X, Y, \boldsymbol{\theta}, \boldsymbol{\varphi}] + \log \pi(\boldsymbol{\theta}) + \log \pi(\boldsymbol{\varphi}), \tag{3.1}$$

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

$$\log \pi(\boldsymbol{\theta}) = \sum_{k=1}^{K} \log \pi(\theta_k) \quad \text{and} \quad \log \pi(\boldsymbol{\varphi}) = \sum_{k=1}^{K} \log \pi(\varphi_k). \tag{3.2}$$

Computing the expected value requires the cluster probabilities $f(Z|X, Y)$, which cannot be evaluated explicitly. 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 clus-

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

ter memberships Z_{-i},

$$\begin{aligned} f(z_i = k|X, Y, Z_{-i}) &\propto f(Y|X, Z_{-i}, z_i = k) f(z_i = k|X, Z_{-i}) \\ &= \left(f_k(Y_{\mathcal{P}_k \cup \{i\}}|X_{\mathcal{P}_k \cup \{i\}}; \theta_k) \prod_{j \neq k} f_j(Y_{\mathcal{P}_j \setminus \{i\}}|X_{\mathcal{P}_j \setminus \{i\}}; \theta_j) \right) g_k(x_i; \varphi_k). \end{aligned} \tag{3.3}$$

Despite our highly dependent situation, (3.3) can be calculated in a simple form, as shown in Proposition 1. The proof is deferred to the Supplementary Material S1.

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

$$f(z_i = k|X, Y, Z_{-i}) \propto \phi((y_i - \mu_k^*)/\sigma_k^*) g_k(x_i; \varphi_k), \quad \textit{where} \tag{3.4}$$

$$\begin{aligned} \mu_k^* &= \mu_k(x_i) + \Phi_{\gamma_k}(x_i, X_{\mathcal{P}_k \setminus \{i\}}) \Phi_{\gamma_k}(X_{\mathcal{P}_k \setminus \{i\}}, X_{\mathcal{P}_k \setminus \{i\}})^{-1} \left(Y_{\mathcal{P}_k \setminus \{i\}} - \mu_k(X_{\mathcal{P}_k \setminus \{i\}}) \right), \\ (\sigma_k^*)^2 &= \sigma_k^2 \left(1 - \Phi_{\gamma_k}(x_i, X_{\mathcal{P}_k \setminus \{i\}}) \Phi_{\gamma_k}(X_{\mathcal{P}_k \setminus \{i\}}, X_{\mathcal{P}_k \setminus \{i\}})^{-1} \Phi_{\gamma_k}(X_{\mathcal{P}_k \setminus \{i\}}, x_i) \right), \end{aligned} \tag{3.5}$$

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

Proposition 1 implies that the cluster is assigned 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

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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 the kth class if y_i is closer to μ_k^* with regard to the scale σ_k^* and g_k has a high mass probability at location x_i.

Once (3.4) 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 a detailed balance (assuming none of the probabilities/densities in (3.4) equal zero), so eventually this process produces samples from $f(Z|X,Y)$. Hence, the cluster membership samples can be used to approximate quantities that depend on $f(Z|X,Y)$, such as the expectation in (3.1). Furthermore, partitioned matrix inverse and determinant formulae (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 the 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 the computational burden by controlling the maximum number of observations in each cluster, denoted by $n_{\max}$; in this case, the total computation becomes $O(Kn_{\max}^3)$. The computation in this step can be distributed easily over multiple cores; in particular, (3.5) can be done separately for different k. The detailed algorithm is given in the Stochastic E-step of the Supplementary Material S3.

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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 a random assignment, and $\tilde{\mathcal{P}}_k = \{i : \tilde{z}_i = k\}$ is the set of indices of the observations in cluster k assigned in $\tilde{Z}$. From (2.7) and (3.2), the log posterior of the complete data in (3.1) 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) estimates $\{\hat{\theta}_k\}_{k=1}^K$ 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 kth 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 the M-step can be done for K clusters separately, which

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

can be efficiently parallelized, as in the Supplementary Material S3.

3.3 Prediction

Predicting the responses y_{new} at a new input location x_{new} can be challenging, because 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 from 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

$$\hat{y}_{\text{new}} := \mathbb{E}[y_{\text{new}}|x_{\text{new}}, X, Y, \tilde{Z}] = \sum_{k=1}^{K} \hat{\mu}_k^* g_k(x_{\text{new}}; \hat{\varphi}_k), \tag{3.6}$$

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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 qth quantile of y_{new}, which is used to construct 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 by generating Monte Carlo samples.

4. Computational Details

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

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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. Among several possibilities for $z(\cdot)$, one may consider a less flexible model, because the GP itself is fairly flexible. For example, the K-class multinomial logistic regression, which produces linear decision boundaries, can be considered. Then, the overall complexity and flexibility of the clustered GP can be determined by carefully selecting the number of clusters, which is described in Section 4.4. The simple decision boundaries are useful for interpreting the clusters, as shown in Sections 5 and 6. The K-class multinomial logistic regression has the form

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

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 consider the linear discriminant analysis (LDA) or quadratic

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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 probability density function of a (multivariate) normal distribution with mean ν_k and covariance Σ_k. LDA assumes $\Sigma_1 = \ldots = \Sigma_K$, whereas QDA assumes the covariances can be different. The multinomial logistic regression and LDA methods are closely connected, often resulting 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 nonparametric or machine learning approaches, such as random forest classification, to model 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 the main advantage of the clustered GP is the flexibility of the GP assisted by the cluster structure, so g_k of an excessively complex form may not help much. As such, we present only the K-class multinomial logistic regression hereinafter.

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

4.2 Initialization

The SEM algorithm can be sensitive to the initialization. Running many initializations and selecting the one that gives the optimal criterion is computationally expensive, especially for large data sets. One potential initialization is the K-means clusters or other unsupervised clustering algorithms based solely 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 those of other clusters, which often leads to nice model interpretation. Although this initialization may end up with a local optimum, the cluster structure still improves the model performance by efficiently exchanging the class assignment over the iterations. As such, in Sections 5 and 6, we use the K-means clusters as the initialization.

4.3 Stopping criteria

The iteration in the SEM algorithm in the Supplementary Material S3 needs a stopping criterion to determine a convergence. For this purpose, we propose using 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 a prediction for the held-out location. Although LOOCV is often too expensive to implement in many situations, because the model has to fit n times

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

in each iteration, the clustered GP has an efficient shortcut that makes LOOCV very affordable. Specifically, denote $\tilde{y}_i$ as the prediction mean based on all data except the ith observation, and y_i as the real output of ith observation. Then, based on (3.6), $\tilde{y}_i = \sum_{k=1}^{K} \hat{\mu}_k^{(-i)} g_k(x_i; \hat{\varphi}_k)$, 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 i that 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 i that belong to $\tilde{\mathcal{P}}_k$, it can be simplified to

$$\hat{\mu}_k^{(-i)} = \hat{\mu}_k(x_i) - \frac{1}{q_{ii}} \sum_{j \neq i}^{n_k} q_{ij} (y_j - \hat{\mu}_k(x_j)), \tag{4.1}$$

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 $O(K n_{\max}^3)$, which is same as the SEM algorithm.

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

4.4 The choice of K

The number of clusters K plays an important role for the degree of nonstationarity of the approximation functions and the flexibility of the model, and thus controls the bias–variance trade-off of the model, which can affect the prediction accuracy. That is, too large a K could lead to an over-flexible model, and too small a K could lead to an under-flexible model. A natural choice is to use cross-validation with different K to target a small prediction error, such as the LOOCV RMSE described in Section 4.3. Other choices that use bootstrap techniques to estimate the prediction error also can be considered, such as the 632+ bootstrap method of Efron and Tibshirani (1997). Kohavi (1995) explicitly compares cross-validation and bootstrapping from bias and variance points of view, and conducts comprehensive numerical experiments. For the purpose of saving computational cost, we choose the K that gives the lowest LOOCV RMSE, because the LOOCV RMSE can be computed efficiently for clustered GPs, as given in (4.1).

4.5 Remarks on alternative implementations and asymptotic properties

The SEM and prediction can be modified in a more fully Bayesian fashion using 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. However, the computational burden

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

for this direction 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 a 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 an approach. For the asymptotic properties of the parameter inference, refer to Nielsen (2000).

5. Numerical Study

In this section, we present several exemplar functions to demonstrate the effectiveness of clustered GPs. We first present examples with lower dimensional inputs to visually present the cluster structure and the benefit of nonstationary modeling, followed by 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 $\tilde{Z}$ that results in the lowest LOOCV RMSE during the iterations; see Section 5.2.

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

The 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, respectively, 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 solid 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, the left panel of Figure 2 shows that the prediction within the region $[10, 20]$ performs poorly with large uncertainty. Ba and Joseph (2012) explained that the constant mean assumption for the GP is violated, so the predictor tends to revert to the global mean, estimated as 0.208 by maximum likelihood estimation in this example. This consequence is observed frequently, especially at locations far from the in-

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

put locations. Moreover, the constant variance assumption for the 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, the clustered GP introduces some degree of nonstationarity by considering a mixture GP, which is shown in the right panel of Figure 2. Two subsets of the data are represented as squares and triangles, which are given by the assignment $\tilde{Z}$ returned by the SEM algorithm, and both are fitted using stationary GPs. The mean estimates of the GPs are -0.045 and 0.529, respectively. It can be seen that the predictor outperforms a stationary GP, especially at locations within the 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 this region. A potential way to improve the accuracy on the boundaries is 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 nonstationary issue. It shows that the prediction and uncertainty quantification are more accurate than the stationary GP, but less accurate than the clustered GP.

Two additional one-dimensional synthetic data generated from the exemplar

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

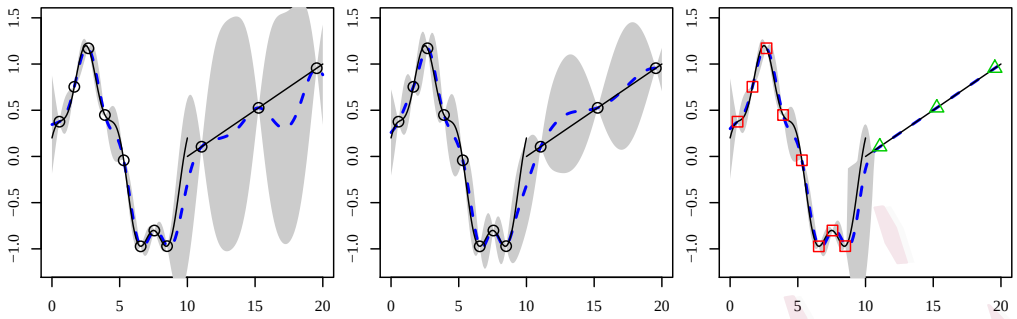

Figure 2: One-dimensional synthetic data. The left, middle, and right panels illustrate the predictors by the stationary GP, composite GP (Ba and Joseph, 2012), and clustered GP, respectively. The solid line is the true function, circles are input locations, and dashed lines are the predictors, with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

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

5.2 Two-dimensional synthetic data

In this section, we demonstrate the selection of K and the stopping rule using the LOOCV RMSE. Consider a wavy function, as in Ba and Joseph (2012) and Montagna and Tokdar (2016). The wavy function is

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

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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 one. 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 dots. The stationary GP, composite GP (Ba and Joseph, 2012), and clustered GP with $K = 3$ are fitted on these locations, the predictive surfaces of which are shown in Figures 3(b–d). It can be seen that the stationary GP and the composite GP perform fairly poorly when x_1 and x_2 are small, whereas 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. The interval scores of their 95% prediction intervals (see equation (43) in Gneiting and Raftery (2007)) are 0.6950, 0.9635,

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

and 2.0915, respectively (the lower the better).

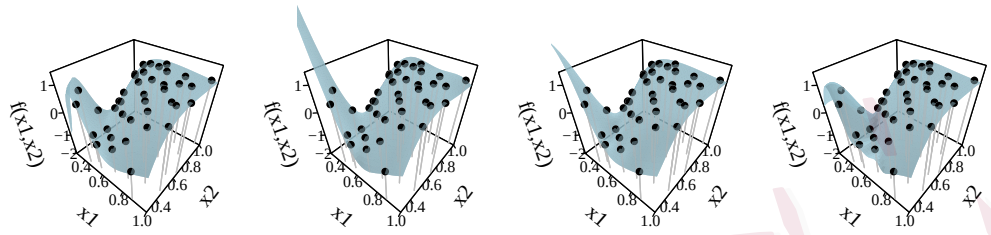

Figure 3: Two-dimensional example: (a) the true wavy function, (b) the stationary GP, (c) the composite GP (Ba and Joseph, 2012), and (d) the clustered GP, where the input locations are shown as dots.

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 the 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 the 36th iteration. For each choice of K, we chose the assignment of the iteration that resulted in the minimum LOOCV RMSE as the final assignment $\tilde{Z}$ for the prediction. The right panel presents the minimum LOOCV RMSE of each choice of K in the 100 iterations. Here, $K = 3$ gives the lowest LOOCV RMSE, and so is selected in this example. Figure 5 demonstrates the assignments at iteration 0, 4, and 36 when $K = 3$. The assignment at iteration 0 represents the initial assignment, which is the K-means clusters, as described in Section 4.2, with a

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

LOOCV RMSE of 0.294. The LOOCV RMSE then drops dramatically in the fourth 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 switched from the circle to the square cluster, and the point $x_1 = 0.702, x_2 = 0.866$ is switched from the triangle to the 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 that of an supervised learning approach, K-means clustering (left panel of Figure 5), which has an RMSE of 0.2728. This is larger than that of the clustered GP, namely, 0.2081. Thus, when the goal is making predictions, our clustering that integrates output information can efficiently improve unsupervised learning clustering, which does not use output information.

5.3 Borehole function

In the section, we consider a borehole function, a more complex exemplar function with eight-dimensional input, to examine the scalability of the clustered GP. The borehole function models water flow through a borehole, and is commonly

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

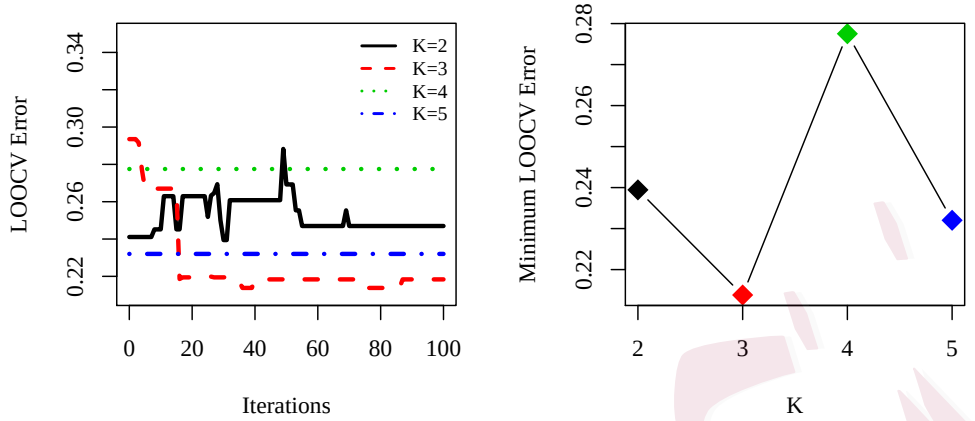

Figure 4: The LOOCV RMSEs with $K = 2, 3, 4$, and 5 during the 100 iteration of the SEM algorithm (left), and the minimum LOOCV RMSE of each choice of K (right).

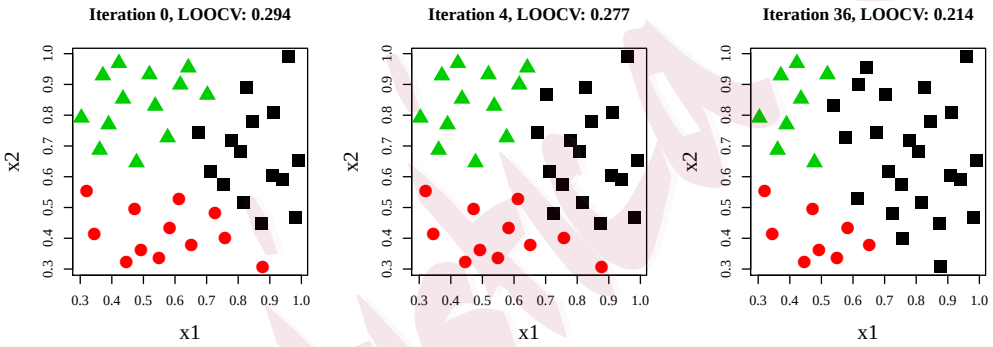

Figure 5: The cluster assignments at iteration 0, 4, and 36 of the SEM algorithm and their LOOCV RMSEs.

used to test methods in computer experiments because of its quick evaluation.

The borehole function is given by

$$f(x) = \frac{2\pi T_u (H_u - H_l)}{\ln(r/r_w)\left(1 + \frac{2LT_u}{\ln(r/r_w) r_w^2 K_w} + \frac{T_u}{T_l}\right)}, \tag{5.1}$$

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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

Consider n uniformly distributed input locations in the input space, and $n_{\text{test}} = 10,000$ random input locations in the same input space in order to examine the prediction accuracy; the outputs are evaluated from (5.1). Four methods are compared, namely, 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 the packages `mlegp` (Dancik, 2013), `laGP` (Gramacy, 2015), `MRFA` (Sung, 2019), and `clusterGP`, respectively, on a MacBook Pro laptop with 2.6 GHz Intel Core i7 and 16 GB 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 $n = 1,000$, whereas the other three methods can incorporate larger n. Even when a stationary GP is feasible, the accuracy is worse than that of `MRFA` and `clusterGP`. Of the four methods, `clusterGP` has better accuracy with reasonable computation time. `MRFA` has slightly larger predictive errors

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

with faster computation. On the other hand, the local GP has larger predictive errors, even though the computation is faster. One may consider a different setting for the local GP (e.g., the size of a subsample), which may lead to better accuracy. Although the proposed method yields better prediction accuracy with a 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 data sets. Some potential remedies for improving the computational efficiency are discussed in Section 7.

6. Solar irradiance prediction

We leverage statistical developments to predict solar irradiance. Predicting solar irradiance, or the power per unit area produced by electromagnetic radiation, plays an important role in power balancing and determining the viability of potential sites for harvesting solar power. We use a data set from simulations of 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 these stations

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

are not uniformly distributed. Figure S2 visualizes the available locations and their corresponding solar irradiance, with the average taken over one year. As shown, many promising locations for solar farms are sparsely covered, particularly in the Midwest. These locations of interest are considered for solar energy forecasting. A detailed description of the data set can be found in Hwang et al. (2018) and Sun et al. (2019b). Similarly to Sun et al. (2019b), we work with average irradiance values over one year from the NAM simulations for each of the 1,535 spatial locations (as shown in Figure S2), and the research interest of this study is making accurate predictions of the solar irradiance at unavailable locations.

From Figure S2, it appears that there are some relatively high solar irradiance measures, relative to their neighborhood, such as the location on the coordinate $(-93.57, 45.99)$, and some relatively low solar irradiance measures, such as the location on the coordinate $(-93.16, 33.69)$. These instances may suggest heterogeneity rather than homogeneity in the input–output relationships. Therefore, the assumption of identical covariance functions throughout the input domain for stationary GPs is likely to fail, and may result in poor performance, as shown in Section 5.

A clustered GP is performed on this data set, using a similar setup to that in Section 5.2. We first use LOOCV to determine the number of clusters K. The

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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. Similarly to the numerical study in Section 5, we chose the assignment of the iteration that 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. The clusters reveal interesting hidden patterns in the input–output relationship. For example, cluster 26 is mostly located in Michigan and part of Pennsylvania and New York, which tells us that some common aspects of solar irradiance are shared in those areas adjacent to the Great Lakes, even though they are not spatially connected. This example shows that clustering can provide useful insights for discovering groups and identifying other interesting aspects of a data set.

To examine its prediction accuracy, we use the LOOCV RMSEs as the prediction error and compare the results with those of a recent emulation method in Sun et al. (2019a), who proposed a multi-resolution global/local GP emulation by extending the idea of a local GP (Gramacy and Apley, 2015). Then, Sun et al. (2019b) applied this method to the same NAM simulation data we exam-

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

ine here. Sun et al. (2019b) reported the LOOCV errors of the multi-resolution global/local GP emulation and those of the ordinary stationary GP. The results, together with those of 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. Its LOOCV predictions 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). In contrast, the multi-resolution global/local GP and the clustered GP perform well, which may suggest that nonstationarity should be taken into account for this data set. Although the LOOCV predictions are visually similar, the LOOCV RMSE of the clustered GP is slightly lower than that of 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.

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

7. Conclusion

In this paper, we have proposed a clustered GP that can simultaneously reduce the computational burden and incorporate nonstationarity, which effectively addresses two major limitations of the stationary GP. Unlike traditional unsupervised clustering methods, the clusters in the clustered GP are *supervised* by the response. The clustered GP uses the response to partition the input domain such that the observations in a cluster have similar features and the same stationary process in the response. This clustering algorithm is implemented using an SEM algorithm, which is available in an open repository. Examples, including solar irradiance simulations, show that the method not only has advantages in terms of computation and prediction accuracy, but also enables the discovery of interesting insights by interpreting the clusters.

The clustered GP offers several avenues for further research. First, the SEM algorithm can be modified in an online fashion. That is, if the data are 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 data set augmented with the additional data. For example, the solar irradiance simulations are available every hour. Thus, a modified algorithm could be used to update the clusters and predict future data in real time. This may save substantial computational cost and storage, especially when the training sample

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

size is extremely large. In addition to the online SEM, subsampling methods can be naturally applied to the clustered GP to alleviate the storage limitations for large-scale data. The CURE algorithm (Guha et al., 2001) provides an efficient method for large-scale data sets for traditional clustering algorithms, employing a combination of random sampling and partitioning. This technique could be applied to the proposed 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 study is on emulations 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. Lastly, to reduce the prediction uncertainty on the boundary between two regions (see, e.g., $x = 10$ in Figure 2), it may be possible 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 topics to future work.

Supplementary Material

The online Supplementary materials contains a detailed proof of Proposition 1, the detailed SEM algorithm in Section 3, and supporting tables and figures for

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

Sections 5 and 6. An `R` package `GPcluster` for implementing the proposed method is available at `https://github.com/ChihLi/GPcluster`.

Acknowledgments

The authors gratefully acknowledge the helpful advice from the associate editor, and two anonymous referees. The authors are grateful to Dr. Furong Sun for sharing the `R` code that implements the multi-resolution global/local GP in Section 6. This work was partly supported by NSF DMS 2113407, and partly by the Mathematics Division of the National Center for Theoretical Sciences in Taiwan.

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.

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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.

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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 approxi-

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

mate 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.

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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.

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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.

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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.

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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

Statistica Sinica: Preprint
doi:10.5705/ss.202020.0456

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.

E-mail: lus@us.ibm.com