Received: 27 December 2018 | Revised: 3 April 2020 | Accepted: 30 April 2020

DOI: 10.1002/env.2631

RESEARCH ARTICLE

WILEY

Detecting British Columbia coastal rainfall patterns by clustering Gaussian processes

F. Paton | **P.D. McNicholas**

Department of Mathematics and Statistics, McMaster University, Hamilton, Ontario, Canada

Correspondence
P.D. McNicholas, Department of Mathematics and Statistics, McMaster University, Hamilton, ON, Canada L8S 4L8.
Email: mcnicholas@math.mcmaster.ca

Funding information
Canada Research Chairs, Grant/Award Number: 950-230797; Natural Sciences and Engineering Research Council of Canada, Grant/Award Number: RGPIN05255-17

Abstract

Functional data analysis is a statistical framework where data are assumed to follow some functional form. This method of analysis is commonly applied to time series data, where time, measured continuously or in discrete intervals, serves as the location for a function's value. Gaussian processes are a generalization of the multivariate normal distribution to function space and, in this article, they are used to shed light on coastal rainfall patterns in British Columbia (BC). Specifically, this work addressed the question over how one should carry out an exploratory cluster analysis for the BC, or any similar, coastal rainfall data. An approach is developed for clustering multiple processes observed on a comparable interval, based on how similar their underlying covariance kernel is. This approach provides interesting insights into the BC data, and these insights can be framed in terms of Pacific Ocean temperatures. From one perspective, the results show that clustering annual rainfall can potentially be used to identify extreme weather patterns.

KEYWORDS

British Columbia, clustering, coastal rainfall, El Niño, extreme weather, Gaussian processes, La Niña, mixture model

1 | INTRODUCTION

In contrast to predictable yearly seasonal changes, El Niño, a well-studied teleconnection, does not occur at regular intervals. This is of particular interest for policy makers, businesses, and people who rely on calculable, foreseeable weather patterns. For example, seasonal changes can specifically alter food production plans, such as when to deploy fishing vessels or harvest crops. While El Niño is primarily categorized through warming temperatures in the eastern and equatorial Pacific Ocean, its effects can be seen around the globe through teleconnections. Teleconnections are an environmental phenomena that describe correlated large-scale atmospheric changes over noncontiguous geographic regions. While some teleconnections are well established, others rely on observing statistical irregularities (Gudmundson & Babkina, 2003; Ward et al., 2014), namely, El Niño's effect on precipitation patterns. Sir Gilbert Walker, a 20th century English scientist, for example, first identified the link between Asian monsoons and Pacific coastal barometer readings (Gudmundson & Babkina, 2003). However, the study of teleconnections, such as precipitation patterns, is notoriously complex because of intricate spatial and temporal correlations. While spatial data are often modeled under the assumption that geographical points near one another share more information than those far apart, work has been done to model both nearby and remote geographical correlations. Hewitt, Hoeting, Done, and Towler (2018) predict precipitation in Colorado by modeling both locally available data and remote Pacific Ocean sea surface temperatures. Understanding

El Niño's effect on distant precipitation can shed light on these patterns. Specifically, classifying irregular precipitation patterns in the Americas can potentially help understand El Niño's impact on local weather systems.

A question of particular interest is: Which years exhibit distinct rainfall patterns? For prediction, knowing how Pacific Ocean changes affect rainfall can have significant implications (O'Gorman, 2015). A model for clustering yearly rainfall data will also give insight to research on the mechanistic properties of teleconnections. This has overlap in the statistical field of functional data clustering, where data are assumed to follow some functional form. Herein, mixture model-based clustering provides an effective approach to cluster data, and Gaussian processes provide a model for rainfall. Much recent work has been done modeling climatological data with spatial and time dependence (Armal, Devineni, & Khanbilvardi, 2018; Cabral, Ferreira, & Friederichs, 2019; Gupta, Jitkajornwanich, Elmasri, & Fegaras, 2016). Specifically, the use of Gaussian processes (GPs) gives a probabilistic starting point. A GP is a stochastic process that generalizes a finite-dimensional normal distribution to function space. GPs have been used to successfully solve complex nonlinear regression and classification problems (Rasmussen, 2005; Roberts et al., 2013). When multiple functions exist on the same interval, usually compact $[0,T]$ and finite, it can be useful to classify them into a finite number of mutually exclusive groups. Here the process would be defined on the index set time, specifically the months of the year.

2 | BACKGROUND

2.1 | Gaussian process

Rasmussen (2005) defines a GP as: "… a collection of random variables, any finite number of which have a joint Gaussian distribution." A GP $\mathbf{f}(\mathbf{x})$ creates a set of random variables evaluated at $\mathbf{x}$. In essence, a GP is a distribution over functions, that is,

$$\mathbf{f}(\mathbf{x}) \sim \mathrm{GP}\left(m(\mathbf{x}),\ k(\mathbf{x},\mathbf{x}^\top)\right), \tag{1}$$

where $m(\mathbf{x}) = \mathbb{E}[\mathbf{f}(\mathbf{x})]$ and $k(\mathbf{x},\mathbf{x}^\top) = \mathbb{E}[(\mathbf{f}(\mathbf{x}) - m(\mathbf{x}))(\mathbf{f}(\mathbf{x}^\top) - m(\mathbf{x}^\top))]$ are the mean function and covariance kernel, respectively, $\mathbf{x} = (x_1,\dots,x_n)^\top$ is the function's index, and $\mathbf{f}(\mathbf{x}) = (f_1,\dots,f_n)^\top$ is the function's output, that is, (x_i, f_i) is a point in $\mathbb{R}^2$. A common GP, and the kernel considered in this article, is defined with mean function $\mathbf{0}$ and squared exponential (SE) covariance function

$$k(x_i, x_j) = \sigma^2 \exp\left\{-\frac{1}{2l^2}(x_i - x_j)^2\right\}, \tag{2}$$

where σ^2 and l are hyperparameters that control the shape of the process; specifically, σ^2 controls the amount of variation in $\mathbf{f}(\mathbf{x})$ and l, the length-scale parameter, controls the correlation. The SE kernel is a widely used (Rasmussen, 2005). One convenient property of the SE kernel is infinite differentiability, which is useful because the first derivative is needed for hyperparameter estimation. Here, the use of the term hyperparameter refers to a set of parameters that make up the "nonparametric model." That is, the hyperparameters only appear in the model's prior and, as shown later in (3), are integrated out of the final model (posterior).

Although the SE is the most popular, different kernel functions can be used. Because the kernel is needed to populate a multivariate normal distribution covariance matrix, the kernel is restricted to produce positive semidefinite matrices. While the SE covariance kernel is a popular choice, modeling the shape of the covariance kernel is an open-ended problem. One systematic solution can be formed by considering a Bayesian model selection framework as discussed in Rasmussen (2005). Duvenaud, Lloyd, Grosse, Tenenbaum, and Zoubin (2013) also provide a solution by considering automatic kernel selection through searching over appropriate kernel structures. For the SE covariance kernel, σ^2 controls the height or the amplitude of the GP and l controls the correlation between observations. This kernel is used to construct a matrix, $\mathbf{K}$, which will serve as the covariance matrix in a multivariate normal distribution introduced in the next section:

$$\mathbf{K} = \begin{pmatrix} k(x_1, x_1) & \dots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \dots & k(x_n, x_n) \end{pmatrix}.$$

While a GP is defined on the entire real line, we only observe a finite number n of realizations $\mathbf{x} = (x_1, \ldots, x_n)^\top$, and corresponding $\mathbf{y} = (y_1, \ldots, y_n)^\top$, where $\mathbf{y} \triangleq \mathbf{f}(\mathbf{x})$. The vector $\mathbf{x}$ is commonly called the input and represents the location of the process, that is, observation $y_i = f(x_i)$. The vector $\mathbf{y}$ is referred to as the output, and is the function evaluated at location $\mathbf{x}$. This allows for a generalization to a multivariate normal distribution via $\mathbf{f}(\mathbf{x}) \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$. This is possible because marginalizing a Gaussian distribution is trivial: the resulting distribution is Gaussian and we can ignore the (x,y) pairs that are unobserved or missing. Here changing the kernel affects the shape of the function, effectively controlling the magnitude of which observations x_i and x_j are correlated. Formulating the problem in this way, we can see the kernel is our prior on the function space, and the (marginal) likelihood for the GP comes after conditioning on the realized points. As shown in the next section, the hyperparameters are often estimated to maximize the likelihood of the GP.

2.2 | Likelihood

The previous section introduced the kernel function and how it relates to a prior on function space and how the hyperparameters affect the correlation between the input $\mathbf{x}$ and outcome $\mathbf{y}$. Now, the likelihood for a GP will be introduced and strategies for choosing the hyperparameters will be illustrated. From the definition of a GP, $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$, which will be shown formally below. First, let $\mathbf{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$, where $\mathbf{K}$ is the covariance matrix constructed from the SE kernel shown in (2).

The likelihood for a GP is conditioned on the observed values to obtain a marginal likelihood, using ϕ to denote the normal density function:

$$p(\mathbf{f} \mid \mathbf{x}) = \underbrace{\phi(\mathbf{f} \mid \mathbf{0}, \mathbf{K})}_{\text{function prior}}, \quad p(\mathbf{y} \mid \mathbf{f}) = \underbrace{\prod_{i=1}^{n} \phi(y_i \mid f_i)}_{\text{likelihood}}.$$

We get the marginal for $\mathbf{y}$ by using Bayes' rule and integrating over $\mathbf{f}$:

$$p(\mathbf{y}|\mathbf{x}) = \int p(\mathbf{y}|\mathbf{f}, \mathbf{x})\, p(\mathbf{f} \mid \mathbf{x}) d\mathbf{f}. \tag{3}$$

Taking the natural logarithm of (3) gives

$$\log p(\mathbf{y}|\mathbf{x}) = -\frac{1}{2}\left\{\mathbf{y}\mathbf{K}^{-1}\mathbf{y}^\top + \log|\mathbf{K}| + n\log 2\pi\right\}. \tag{4}$$

This likelihood can be broken down into three main components: the data fit term, model complexity term, and a constant term:

$$\log p(\mathbf{y}|\mathbf{x}) = -\frac{1}{2}\left\{\underbrace{\mathbf{y}\mathbf{K}^{-1}\mathbf{y}^\top}_{\text{data fit}} + \underbrace{\log|\mathbf{K}|}_{\text{complexity}} + \underbrace{n\log 2\pi}_{\text{constant}}\right\}. \tag{5}$$

The data fit and complexity component share an interesting trade-off. For small length-scale values l, the model will fit the data well and the data fit component will be small. However, points will not be considered "near" each other, resulting in a high model complexity. Conversely, if l is large (suggesting no correlation between points), then the complexity will be small but the data fit term will be large (Murphy, 2012). This is because the SE kernel will converge to σ^2, turning $\mathbf{K}$ into a diagonal matrix. Because GPs have these inherent penalty terms for over- and underfitting, crossvalidation is generally not used to estimate kernel hyperparameters.

2.3 | Predictive distribution

GPs are commonly used in supervised regression tasks for their ability to nonparametrically approximate complex functions and solve functional engineering problems (Bin & Wenlai, 2013). It is often of interest to infer the

function's value outside of the paired training data $(\mathbf{x}, \mathbf{y})$. To do this, a predictive distribution can be constructed. Let $\mathbf{y}_* = \mathbf{f}(\mathbf{x}_*)$ be the unobserved outputs to be inferred at locations $\mathbf{x}_*$. The joint distribution can be derived through probabilistic terms,

$$\mathbf{f}(\mathbf{x}) \sim \mathcal{N}(\mathbf{0}, \mathbf{K}(\mathbf{x}, \mathbf{x})), \tag{6}$$

$$\mathbf{f}(\mathbf{x}_*) \sim \mathcal{N}(\mathbf{0}, \mathbf{K}(\mathbf{x}_*, \mathbf{x}_*)), \tag{7}$$

$$\begin{bmatrix} \mathbf{f}(\mathbf{x}) \\ \mathbf{f}(\mathbf{x}_*) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}(\mathbf{x}, \mathbf{x}) & \mathbf{K}(\mathbf{x}, \mathbf{x}_*) \\ \mathbf{K}(\mathbf{x}_*, \mathbf{x}) & \mathbf{K}(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix}\right). \tag{8}$$

Note that (8) is the joint distribution of the observed pairs $(\mathbf{x}, \mathbf{y})$ and unobserved pairs $(\mathbf{x}_*, \mathbf{f}(\mathbf{x}_*))$. The expected value for $\mathbf{f}(\mathbf{x}_*)$ can be derived using conditional properties, which leads to

$$\hat{\mathbf{f}}(\mathbf{x}_*) \triangleq \mathbb{E}[\mathbf{f}(\mathbf{x}_*)|\mathbf{x}, \mathbf{y}, \mathbf{x}_*] = \mathbf{K}(\mathbf{x}_*, \mathbf{x})[\mathbf{K}(\mathbf{x}, \mathbf{x})]^{-1}\mathbf{y}; \tag{9}$$

a complete derivation is given by Rasmussen (2005). Then, (9) can then be used to compute estimates for the value of the function $\mathbf{f}(\mathbf{x}_*)$ at location $\mathbf{x}_*$.

2.4 | Model-based clustering

Clustering, a.k.a. unsupervised classification, is an unsupervised machine learning task that attempts to classify (unlabeled) data points into distinct groups. Commonly, clustering is defined as assigning data into groups such that data in the same cluster are more similar to each other than to data in different clusters. Initially this definition seems intuitive; however, practically there are some problems. Namely, grouping each data point into its own cluster would satisfy this definition. Following several others (e.g., Tiedeman, 1955; Wolfe, 1963), McNicholas (2016a) provides a definition not based on similarity: "a cluster is a unimodal component within an appropriate finite mixture model," where the word 'appropriate' requires consideration and, specifically, that the component densities have the necessary flexibility to fit the data (see McNicholas, 2016a, for further discussion). Whatever definition of a cluster one may prefer, many methods have been developed to tackle this problem in unsupervised learning. Model-based clustering refers to the use of finite mixture models for clustering.

2.5 | Finite mixture model

The finite mixture model is a popular tool for (model-based) clustering—see Bouveyron and Brunet-Saumard (2014) and McNicholas (2016b) for recent reviews. The density of a G-component finite mixture model is

$$f(\mathbf{x}|\boldsymbol{\varphi}) = \sum_{g=1}^{G} \pi_g f_g(\mathbf{x}|\boldsymbol{\theta}_g), \tag{10}$$

where $\pi_g > 0$, with $\sum_{g=1}^{G} \pi_g = 1$, is the gth mixing proportion, $f_g(\mathbf{x}|\boldsymbol{\theta}_g)$ is the gth component density, and $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_G)$ are the component density-specific parameters, with $\boldsymbol{\varphi} = (\boldsymbol{\pi}, \boldsymbol{\theta})$ and $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_G)$. The likelihood for $\mathbf{x}_1, \ldots, \mathbf{x}_n$ in a model-based clustering paradigm, using a Gaussian mixture model, is given by

$$\mathcal{L}(\boldsymbol{\varphi}) = \prod_{i=1}^{n} \sum_{g=1}^{G} \pi_g \, \phi(\mathbf{x}_i \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g), \tag{11}$$

where $\phi(\mathbf{x}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ is the density of a multivariate Gaussian distribution with mean $\boldsymbol{\mu}_g$ and covariance matrix $\boldsymbol{\Sigma}_g$.

2.6 | Expectation–maximization algorithm

Model-based clustering requires estimating the unknown model parameters from the likelihood in (11). The expectation-maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977) provides a good starting point for this problem. Each iteration of the EM algorithm starts by computing the expectation of the complete-data log-likelihood (E-step), then maximizes the conditional expectation of the complete-data log-likelihood (M-step). The E- and M-steps are iterated until some stopping rule is met. Consider a Gaussian model-based clustering complete-data likelihood, denoted by $\mathcal{L}_c(\boldsymbol{\varphi})$, where $\boldsymbol{\varphi}$ denotes all the parameters and the complete-data comprise the observed $\mathbf{x}_1,\dots,\mathbf{x}_n$ together with the missing labels $\mathbf{z}_1,\dots,\mathbf{z}_n$ defined so that $\mathbf{z}_i = (z_{i1},\dots,z_{iG})$ and $z_{ig} = 1$ if observation i belongs to component g and $z_{ig} = 0$ otherwise. Now, for the Gaussian mixture model-based clustering paradigm—corresponding to the likelihood in (11)—we have complete-data likelihood

$$\mathcal{L}_c(\boldsymbol{\varphi}) = \prod_{i=1}^{n}\sum_{g=1}^{G}[\pi_g\ \phi(\mathbf{x}_i \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)]^{z_{ig}}. \tag{12}$$

In the E-step, we compute

$$\hat{z}_{ig} := \mathbb{E}[Z_{ig}|\ \mathbf{x}_i] = \frac{\hat{\pi}_g\phi(\mathbf{x}_i \mid \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Sigma}}_g)}{\sum_{h=1}^{G}\hat{\pi}_h\phi(\mathbf{x}_i \mid \hat{\boldsymbol{\mu}}_h, \hat{\boldsymbol{\Sigma}}_h)} \tag{13}$$

conditional on the current parameter updates (estimates). Next, in the M-step, the parameters are updated. This amounts to estimating the covariance matrix and mean vector for a Gaussian mixture model. In the M-step, the updates are:

$$\hat{\pi}_g = \frac{n_g}{n}, \tag{14}$$

$$\hat{\boldsymbol{\mu}}_g = \frac{1}{n_g}\sum_{i=1}^{n}\hat{z}_{ig}\mathbf{x}_i, \tag{15}$$

$$\hat{\boldsymbol{\Sigma}}_g = \frac{1}{n_g}\sum_{i=1}^{n}\hat{z}_{ig}(\mathbf{x_i} - \hat{\boldsymbol{\mu}}_g)(\mathbf{x_i} - \hat{\boldsymbol{\mu}}_g)^\top, \tag{16}$$

where $n_g = \sum_{i=1}^{n}\hat{z}_{ig}$. After parameter estimation is completed, the clustering results are expressed through the probabilities $\hat{z}_{ig}$, that is, $\hat{z}_{ig}$ is the probability that $\mathbf{x}_i$ belongs to component g. These soft probabilities $\hat{z}_{ig} \in [0, 1]$ are often converted into hard classifications via maximum a posteriori probabilities:

$$\text{MAP}(\hat{z}_{ig}) = \begin{cases} 1 & \text{if } g = \text{argmax}_h(\hat{z}_{ih}), \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

Extensive details on model-based clustering and parameter estimation are given by McNicholas (2016a).

3 | MODEL TO CLUSTER FUNCTIONAL DATA

3.1 | Model formulation

We have seen that the log-likelihood for a GP with the observed output vector $\mathbf{y}$ and corresponding input vector $\mathbf{x}$ was distributed according to a multivariate Gaussian distribution. When clustering GPs, the goal will be to find clusters that contain processes that have similar paths. The meaning of similar path refers not only to how close two processes' values are but also to how similar their shapes are (smooth, wiggly, etc.). Now let us define the notation used for the model: the ith GP will have output vector $\mathbf{y}_i$, input vector $\mathbf{x}_i$, and

$$p(\mathbf{y}_i|\boldsymbol{\theta}_i, \mathbf{x}_i) = \exp\left\{-\frac{1}{2}\left(\mathbf{y}_i\mathbf{K}^{-1}\mathbf{y}_i^\top + \log|\mathbf{K}| + n\log 2\pi\right)\right\}. \tag{18}$$

The density in (18) is the probability density function, that is, $p_g(\mathbf{y}_i|\boldsymbol{\theta}_g, \mathbf{x}_i)$, used as the component density for the finite mixture model

$$p(\mathbf{y}_i|\boldsymbol{\theta}, \mathbf{x}_i) = \sum_{g=1}^{G} \pi_g p_g(\mathbf{y}_i|\boldsymbol{\theta}_g, \mathbf{x}_i), \tag{19}$$

where $\boldsymbol{\theta}_g = \{l_g, \sigma_g\}$ denotes the hyperparameters for the gth cluster. Because the likelihood of a GP is a Gaussian distribution with covariance matrix $\mathbf{K}$, the complete-data likelihood is given by

$$\mathcal{L}_c(\boldsymbol{\varphi}) = \prod_{i=1}^{n} \sum_{g=1}^{G} [\pi_g \phi(\mathbf{y}_i \mid \mathbf{0}, \mathbf{K}_g)]^{z_{ig}}, \tag{20}$$

where $\mathbf{K}_g$ is the covariance matrix corresponding to cluster g and $\phi(\mathbf{y}_i|\mathbf{0},\mathbf{K}_g)$ is the Gaussian density with mean $\mathbf{0}$ and covariance $\mathbf{K}_g$. An SE covariance kernel is used as the prior on the function space, that is,

$$k(x_i, x_j) = \sigma^2 \exp\left\{-\frac{1}{2l^2}(x_i - x_j)^2\right\}. \tag{21}$$

The goal is to recover the G pairs of kernel hyperparameters $\boldsymbol{\theta}_g = \{l_g, \sigma_g^2\}$ and the mixing parameters $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_G)$, and thence to estimate the latent variables $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$.

3.2 | GP parameter estimation

The first step is to estimate each GP's kernel hyperparameters. Herein, the kernel hyperparameters for the ith GP is denoted by $\boldsymbol{\Theta}_i = \{l_i, \sigma_i\}$. In this step, the maximized kernel hyperparameters for each GP, $l_i^{\max}$ and $\sigma_i^{\max}$, are estimated. To find these maximized hyperparameters, an MLE solution is found using gradient ascent, starting with the log-likelihood, that is,

$$\log p(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\Theta}_i) = -\frac{1}{2}\left\{\mathbf{y}_i \mathbf{K}^{-1} \mathbf{y}_i^\top + \log|\mathbf{K}| + n \log 2\pi\right\}. \tag{22}$$

The derivative is then taken w.r.t. to the kernel hyperparameters

$$\frac{\partial}{\partial \boldsymbol{\Theta}_i} \log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\Theta}_i) = \frac{1}{2}\mathbf{y}^\top \mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \boldsymbol{\Theta}_i} \mathbf{K}^{-1}\mathbf{y} - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \boldsymbol{\Theta}_i}\right) = \frac{1}{2}\mathrm{tr}\left\{\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^\top - \mathbf{K}^{-1}\right)\frac{\partial \mathbf{K}}{\partial \boldsymbol{\Theta}_i}\right\}, \tag{23}$$

where $\boldsymbol{\alpha} = \mathbf{K}^{-1}\mathbf{y}$. The partial derivatives for l_i and σ_i are calculated from the first derivatives of the kernel function:

$$\frac{\partial \mathbf{K}}{\partial l_i} = \sigma_i^2 \exp\left\{-\frac{1}{2l_i^2}(x_i - x_j)^2\right\}(x_i - x_j)^2 l_i^{-3}, \tag{24}$$

$$\frac{\partial \mathbf{K}}{\partial \sigma_i} = 2\sigma_i \exp\left\{-\frac{1}{2l^2}(x_i - x_j)^2\right\}. \tag{25}$$

After finding the gradient for the likelihood, a gradient ascent algorithm is used to find a sufficiently close solution. This algorithm is given by repeating the following updates until a stopping rule is satisfied:

$$\begin{aligned} l_i^{(t+1)} &= l_i^{(t)} + \lambda \frac{\partial}{\partial l_i} \log p(\mathbf{y}_i|\mathbf{x}, l_i^{(t)}), \\ \sigma_i^{(t+1)} &= \sigma_i^{(t)} + \lambda \frac{\partial}{\partial \sigma_i} \log p(\mathbf{y}_i|\mathbf{x}, \sigma_i^{(t)}), \end{aligned} \tag{26}$$

where superscript (t) denotes iteration t. After maximizing the kernel hyperparameters, we have $\hat{\boldsymbol{\Theta}} = \{\hat{\boldsymbol{\Theta}}_1, \hat{\boldsymbol{\Theta}}_2, \ldots, \hat{\boldsymbol{\Theta}}_n\}$, where $\hat{\boldsymbol{\Theta}}_1 = \{l_1^{\max}, \sigma_1^{\max}\}$ denotes the maximized kernel hyperparameters for the first GP, $\hat{\boldsymbol{\Theta}}_2$ denotes the hyperparameters for the second GP, and so on.

3.3 | Cluster parameter estimation

The model seeks to cluster the processes and make inferences on the latent variables. A modified EM approach is used. First, the mixing proportions and cluster hyperparameters are initialized randomly, that is, we initialize $\boldsymbol{\pi}$, l, and $\boldsymbol{\sigma}$, where $l = \{l_1,\dots,l_G\}$, $\boldsymbol{\sigma} = \{\sigma_1,\dots,\sigma_G\}$, and $\boldsymbol{\pi} = \{\pi_1,\dots,\pi_G\}$ randomly. Next, each GP's $(\text{GP}_1,\dots,\text{GP}_n)$ responsibilities are calculated for each of the G clusters to get $n \times G$ responsibilities

$$\hat{r}_{ig} = \frac{\pi_g \ \phi(\mathbf{y}_i|\mathbf{0}, \mathbf{K}_g)}{\sum_{h=1}^{G} \ \pi_h \ \phi(\mathbf{y}_i|\mathbf{0}, \mathbf{K}_h)}. \tag{27}$$

Note that $\hat{r}_{ig}$ represents the responsibility, or conditional expected value, of the ith process belonging to the gth cluster, and $\hat{r}_{ig} \triangleq \hat{z}_{ig}$. After the responsibilities are calculated, the mixing proportions $\pi_1,\dots,\pi_G$ are conditionally maximized on these responsibilities. The update for the gth mixing proportion is

$$\pi_g = \frac{m_g}{m},$$

where $m_g = \sum_{i=1}^{n} r_{ig}$ is the responsibility for cluster g and $m = \sum_{g=1}^{G} m_g$. The cluster-specific kernel hyperparameters, that is, l_g and σ_g, are then updated, where l_g is the length-scale parameter for cluster g and σ_g is the height parameter for cluster g:

$$\hat{l}_g = \frac{1}{m_g} \sum_{i=1}^{n} \hat{r}_{ig} \ l_i^{\max}, \quad \hat{\sigma}_g = \frac{1}{m_g} \sum_{i=1}^{n} \hat{r}_{ig} \ {\sigma_i}^{\max},$$

that is, we are weighting the maximized hyperparameters, $\sigma_i^{\max}$ and $l_i^{\max}$, by their respective cluster responsibility $\hat{r}_{ig}$.

This scheme, for calculating the responsibilities then updating the cluster parameters, is repeated until some stopping rule is met. In this case, when the change in expected complete-data log likelihood at iteration t becomes small, that is, until

$$\left| \mathcal{Q}(\boldsymbol{\varphi}^{(t)}, \boldsymbol{\varphi}^{(t-1)}) - \mathcal{Q}(\boldsymbol{\varphi}^{(t-1)}, \boldsymbol{\varphi}^{(t-2)}) \right| < \epsilon,$$

where $\mathcal{Q}(\boldsymbol{\varphi}^{(t)}, \boldsymbol{\varphi}^{(t-1)}) = \mathbb{E}[\log\left(\mathcal{L}_c(\boldsymbol{\varphi}^{(t)})\right) | \mathbf{y}, \boldsymbol{\varphi}^{(t-1)}]$ is the expectation of the complete-data log likelihood.

3.4 | Numerical issues

At each iteration of gradient ascent, the GP's likelihood gradient needs to be computed:

$$\frac{\partial}{\partial \boldsymbol{\Theta}_i} \log p(\mathbf{y}_i|\mathbf{x}, \boldsymbol{\Theta}_i) = \frac{1}{2} \text{tr} \left\{ \left(\boldsymbol{\alpha}\boldsymbol{\alpha}^\top - \mathbf{K}^{-1} \right) \frac{\partial \mathbf{K}}{\partial \boldsymbol{\Theta}_i} \right\}. \tag{28}$$

This operation requires inverting a $t \times t$ matrix $\mathbf{K}^{-1}$. Inverting large matrices is notoriously computationally unstable, especially when the matrices are not full rank (or sufficiently close) and eigenvalues become very large or very small. One solution is to first decompose the matrix into lower triangular form, that is, $\mathbf{K} = \mathbf{L}\mathbf{L}^\top$ and then invert $\mathbf{K}$ via $\mathbf{K}^{-1} = (\mathbf{L}^{-1})^\top \mathbf{L}^{-1}$. We use the R (R Core Team, 2018) package `FastGP`(Gopalan & Bornn, 2016), which implements the package `RcppEigen`(Bates & Eddelbuettel, 2013) to invert the lower triangular matrix $\mathbf{L}$.

4 | SIMULATION STUDIES

This section will first look at two cases of simulated data. The hyperparameters $\boldsymbol{\theta} = \{\boldsymbol{l}, \boldsymbol{\sigma}\}$ and the mixing proportions $\boldsymbol{\pi}$ will vary based on the simulated sets. The method developed in the previous section will then be applied to recover the hyperparameters and classify each GP into their respective groups. For the two simulation studies, noiseless-squared exponential covariance functions will be used, which in effect means that a perfectly interpolated, noiseless process is observed for the simulation.

4.1 | Simulation I

The first simulation starts with generating 30 GPs. The processes are generated on the interval [0,10] with $T = 7$ evenly spaced realizations, that is, each process has seven values spread evenly on the interval. In all, 10 of the 30 GPs are generated from a multivariate normal distribution using the R package `mvtnorm` (Genz et al., 2009). Where the covariance matrix was constructed using an SE covariance kernel with hyperparameters $l = 1$ and $\sigma = 3$. The remaining 20 were generated similarly but with a covariance matrix constructed with hyperparameters $l = 3$ and $\sigma = 3$.

After running the algorithm described in Section 3, estimates for the set of hyperparameters and mixing proportion were recorded (Table 1). The mixing proportion is easily identified and accurately estimated. Using the MAP classification, the algorithm was able to correctly classify each process. Table 1 gives the mean parameter estimates and standard errors. This was done by randomly starting the algorithm 10 times—that is, initializing the parameters from a random uniform draw– and calculating the mean and standard error from these 10 starts.

Once the processes are colored by their MAP classification (Figure 1), one can visually see the difference between the two process clusters. The processes ($g = 2$, blue) with the larger length-scale $l = 3$ are smoother compared with those generated from the process with length-scale $l = 1$.

The length-scale parameter l was also readily recovered in this scenario, producing similar estimates to the true hyperparameter. The hyperparameter σ_1, which, recall, controls the function's variance (in y, the function's output), is not near the true parameter value. One reason for this could because this cluster has a comparatively small length-scale $l = 1$, which models the relative correlation between the points.

Parameter	True value	Mean estimate	Standard error
π_1	0.33	0.33	0
π_2	0.67	0.67	0
l_1	1	1.23	0.02
l_2	3	3.08	0.03
σ_1	3	2.18	0.07
σ_2	1	1.43	0.09

TABLE 1 Mean values for recovered hyperparameters, with standard errors, for Simulation I

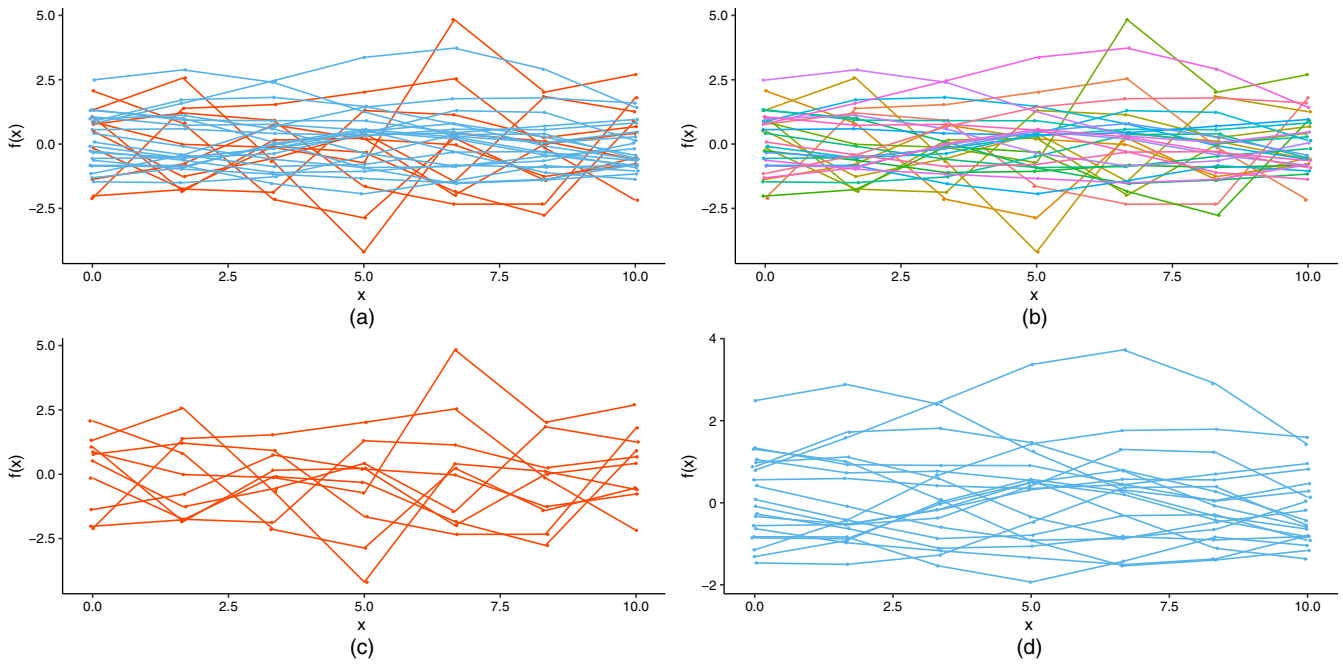

FIGURE 1 The 30 GPs from Simulation I: (a) colored by MAP classification, red lines are Cluster 1 and blue lines are Cluster 2. (b) Colored by individual GP. (c) GPs from Cluster 1. (d) GPs from Cluster 2

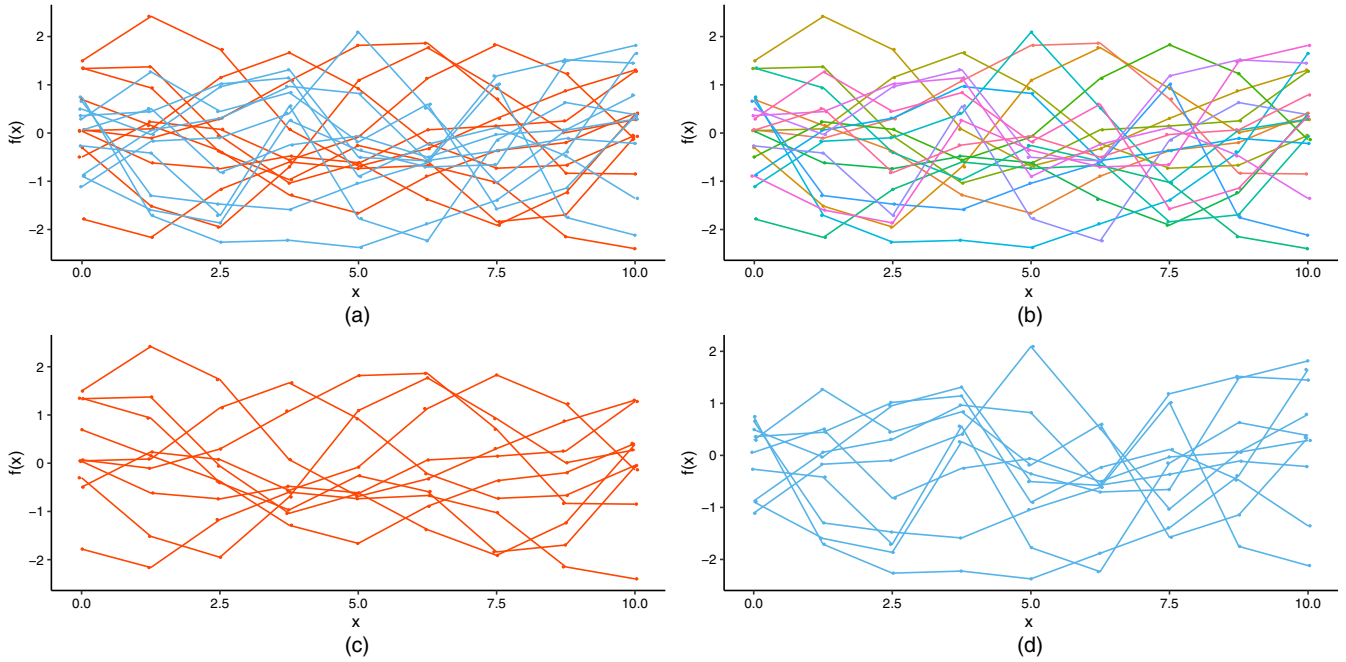

FIGURE 2 The 20 GPs from Simulation II: (a) Colored by their MAP classification, red lines are Cluster 1, blue lines are Cluster 2. (b) Colored by individual GP. (c) GPs from Cluster 1. (d) GPs from Cluster 2

TABLE 2 Mean values for recovered hyperparameters, with standard errors, for Simulation II

Parameter	True value	Mean estimate	Standard error
π_1	0.5	0.52	0.004
π_2	0.5	0.48	0.004
l_1	1	1.30	0.032
l_2	2	2.10	0.018
σ_1	1	2.01	0.081
σ_2	2	2.05	0.082

4.2 | Simulation II

The second simulation was carried out by first generating 20 GPs. Ten were generated from an SE covariance kernel with hyperparameters $l = 1$ and $\sigma = 1$. The remaining 10 GPs were generated from a covariance kernel with hyperparameters $l = 2$ and $\sigma = 2$. Similarly to Simulation I, the GPs were generated first by constructing the covariance matrix, then by generating random samples using the R package `mvtnorm`. In all, $T = 9$ equally spaced observed values were recorded for each GP (Figure 2). Based on the plot in Figure 2, there seems to be no clear distinction or natural groups of processes. After coloring the processes by their (correct) classifications (Figure 2), there is still ambiguity about the two groups separation.

5 | COASTAL RAINFALL IN BRITISH COLUMBIA

This section will look at historical monthly precipitation data for coastal regions of British Columbia (BC), Canada. These data are recorded by the Government of Canada and collected from the weather stations: Tofino A, Vancouver International Airport, Port Hardy A, and Victoria International Airport (Figure 3). These data were derived from the following resources available in the public domain: http://climate.weather.gc.ca/climate_data/monthly_data_e.html (Government of Canada, 2007).

FIGURE 3 Map of four weather stations, three of which are located on Vancouver Island

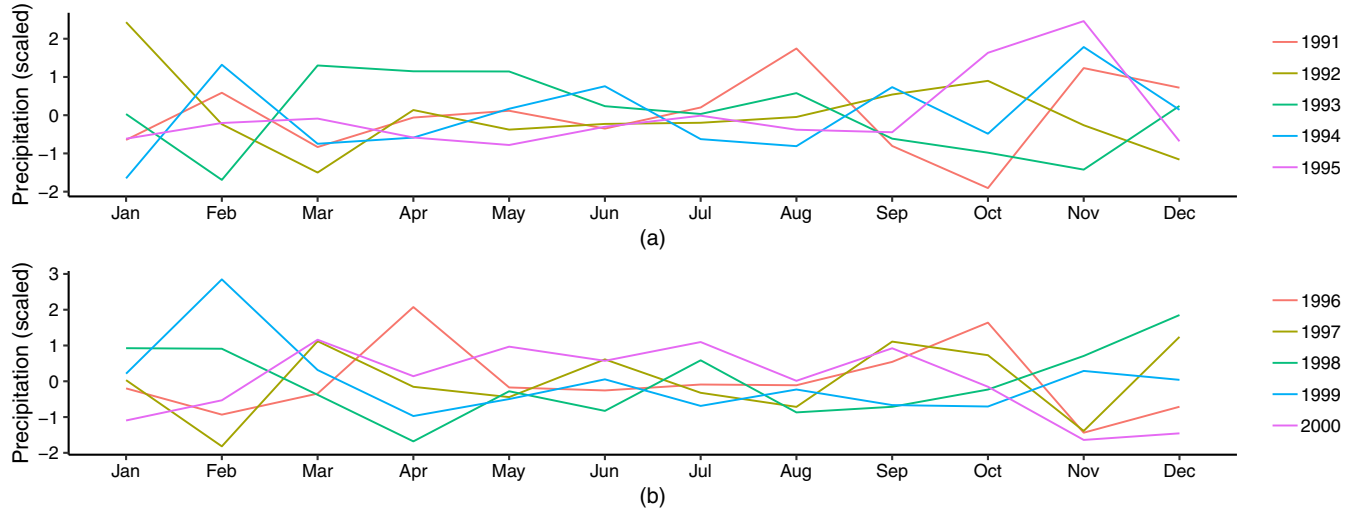

FIGURE 4 Scaled and season trend removed monthly precipitation for the Tofino coastal region of British Columbia, Canada. The points are (artificially) connected between months for illustrative purposes, there are 12 measurements per year

Again, the method accurately recovers the mixing parameter and length-scale (Table 2). However, for Cluster 1, the length-scale l is slightly overestimated and the method inflates σ to account for the variance in the function's output. The parameter estimates were calculated by randomly starting the algorithm 10 times and using the mean. Notably, the processes look very similar between groups, so much so that this solution might seem unconvincing if true group labels were unknown.

Total monthly precipitation was recorded from January 1991 to December 2000. The 10 years will be treated as independent GPs and each station will be treated separately for clustering purposes. The objective is to find $G = 2$ cluster solutions for each station and compare the resulting clusters—because there are only 10 processes per station, $G > 2$ cluster solutions were not explored. The data were processed first by removing the seasonality, that is, the residual precipitation after a 10 year monthly average was removed. The data were then centered and scaled such that the mean is 0 and standard deviation is 1. These analyses will use the SE covariance function discussed earlier. Thus, $\boldsymbol{\theta} = \{\boldsymbol{l}\}$ will be estimated and modeled holding $\boldsymbol{\sigma} = 1$ constant. Note that the observed outputs $\mathbf{y}$ are (artificially) connected by lines for illustrative purposes (Figure 4).

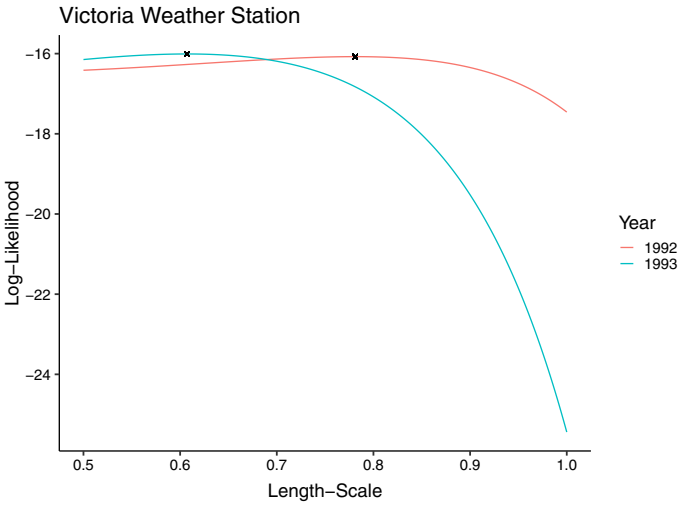

FIGURE 5 The length scale parameter was chosen by running a grid search over values between 0 and 2. Two years are shown here with each year's respective maximum denoted by the black point

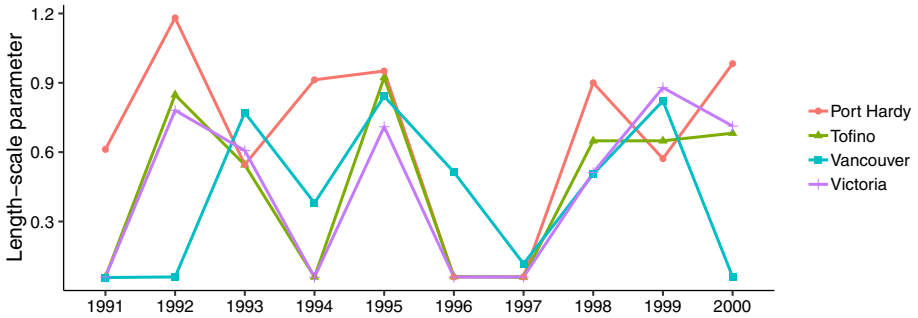

FIGURE 6 Optimized length-scale hyperparameter for the 10 years of precipitation data, by weather station. The years in Group 2 have larger length scale values

Figure 4 illustrates the scaled Tofino data plotted by year. Next, the maximized hyperparameter l is fitted and results are shown in Figure 6. Because of multimodal likelihoods in the gradient ascent approach to hyperparameter fitting, a grid search was used to choose the optimal value. An example likelihood from this is shown in Figure 5.

Instead of applying the EM algorithm to estimate mixture parameters, the relatively small number of processes for each station meant that an exhaustive search could be used. That is, each two-group combination of the 10 years was considered. This amounted to maximizing $2^{10} = 1{,}024$ likelihoods and selecting the model with the highest (complete-data) likelihood. Using these results, there seem to be two groups emerging, one group with a smaller length-scale and one group with a larger length-scale parameter. Group 2 has a larger group length-scale parameter ($0.842 \leq l_2 \leq 0.985$) than Group 1 ($0.365 \leq l_1 \leq 0.401$). Group 2 is labeled the "irregular group" as, with the exception of Port Hardy, the mixing proportion suggests only about 10% of the data come from this group (see Table 4). In this case, the years with a smaller length-scale indicate monthly rainfall is less correlated month-to-month than those with a larger length-scale.

Vancouver, Tofino, and Port Hardy all had the year 1995 assigned to Group 2 (Table 3). In addition, Port Hardy had the years 1992, 1994, 1998, and 2000 assigned to Group 2. Two of the years assigned to Group 2 in at least one weather station (1995, 1998) share an interesting characteristic related to Pacific Ocean temperatures. Specifically, El Niño and La Niña are events classified using the Oceanic Niño Index (ONI), an index that measures irregular ocean temperature changes over a 3-month moving average. An El Niño (irregularly warm) event immediately preceded a La Niña (irregularly cold) event twice during the studied time period. Once in 1995 and again in 1998. Both times the year started with warm enough ocean temperatures to classify it as an El Niño period, and by the end of the calendar year the ocean had cooled enough to be classified as La Niña (NOAA, 2019). The other years in the irregular group also differed in terms of regular Pacific Ocean temperatures. Generally, years clustered into the irregular group tend to correspond to falling Equatorial Pacific Ocean temperatures as shown in Figure 7a. These irregular years also coincide with years where the middle region of the Pacific Ocean (5S–5N and 170–120W) started the calendar year warmer than it ended, as illustrated in Figure 7b.

Weather station	Years in group two
Vancouver	1995
Tofino	1995
Port Hardy	1992, 1994, 1995, 1998, 2000
Victoria	1999

TABLE 3 Years in Group 2, for each weather station, based on maximizing the complete-data log-likelihood, where remaining years belong to Group 1

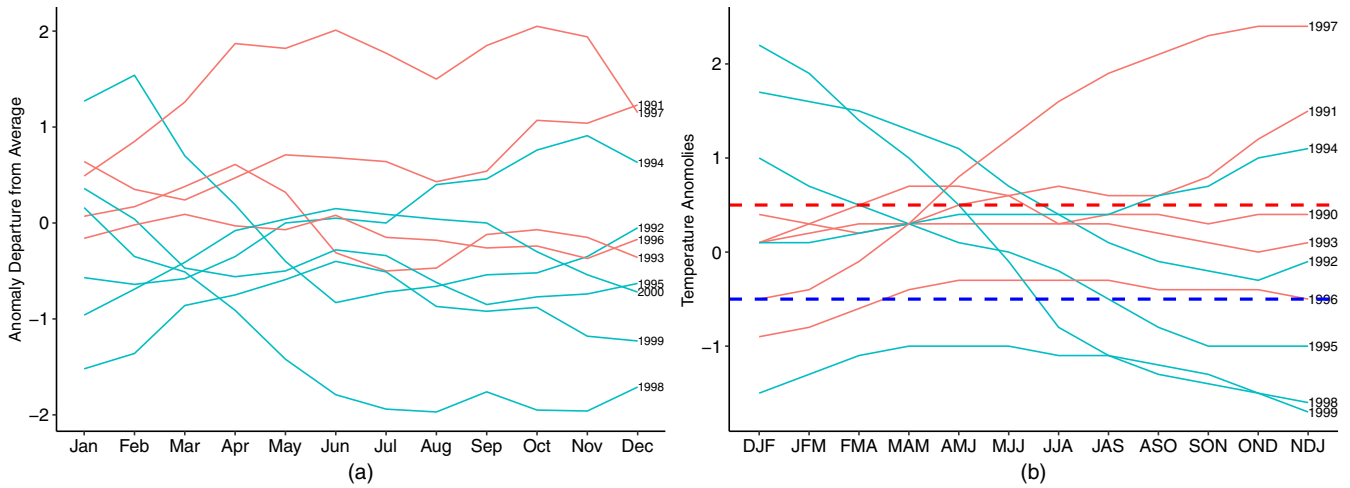

FIGURE 7 (a) Equatorial (160E–80W) upper (surface to 300M) ocean temperature average anomaly based on 1981–2010 climatology. Shows monthly temperature relative to 30 year average. Blue lines (1992, 1994, 1995, 1998, 1999, 2000) represent years that were clustered into Group 2 for at least one weather station. Red lines represent the remaining years. Group 2 years tend to coincide with cooling temperatures. (b) Pacific Ocean temperatures 3 month running mean of anomalies for region 3.4 (Middle Pacific Ocean, 5S–5N and 170–120W). Blue lines represent years that at least one weather station clustered into Group 2 (irregular). Red lines represent the remaining years. Warm (red dashed line) and cold (blue dashed line) are a ±0.5 threshold for the Oceanic Niño Index (ONI)

From further consideration of the estimated cluster parameters in Table 4, Group 2's years tend toward a larger length-scale compared with Group 1. This suggests that, in years with cooling Pacific Ocean temperatures, rainfall patterns change more smoothly (i.e., are more correlated) across months as opposed to regular weather years.

6 | DISCUSSION

A method for clustering functional data has been introduced to cluster coastal rainfall data from BC. First, the hyperparameters that make up a GP were optimized through a gradient-based maximum likelihood optimizer. Because of computational feasibility, parameter estimates were obtained by considering every two-group combination of years and choosing the maximum likelihood. For the simulation studies, the usual EM algorithm for finite Gaussian mixture models was modified. Instead of maximizing the standard covariance matrix, hyperparameters for a kernel function that measures correlation in $\mathbf{x}$ were optimized. The covariance matrix was then constructed from the optimal kernel parameters. Herein, we fix $G = 2$ as known; however, if one were to consider data with more processes per station then $G > 2$ could be considered. Notably missing, or incomplete, data can easily be handled by the proposed approach, either by using the predictive distribution to impute the missing data or by ignoring the missing values. This is possible because the model makes inference on the underlying hyperparameters of the kernel, and not the particular index set of the process.

Two simulation studies were performed. When GPs from different distributions had a large difference in their length-scale parameter l (i.e., 1 vs. 3), parameters were readily recovered. When GPs had similar length-scale parameters, l was recovered but σ tended to shrink toward a common estimate between both clusters. The application to the rainfall data from the coastal region of BC discovered two groups of years, one which contained "regular" years and the other

TABLE 4 Clustering results, parameters recovered

Tofino		Victoria	
Parameter	**Mean estimate**	**Parameter**	**Mean estimate**
π_1	0.90	π_1	0.90
π_2	0.10	π_2	0.10
l_1	0.401	l_1	0.395
l_2	0.923	l_2	0.879
Vancouver		**Port Hardy**	
Parameter	**Mean estimate**	**Parameter**	**Mean estimate**
π_1	0.90	π_1	0.50
π_2	0.10	π_2	0.50
l_1	0.365	l_1	0.370
l_2	0.842 l_2	l_2	0.9850

Note: These estimates are the result of considering each possible two-group combination and selecting the one with the greatest (complete-data) likelihood. The clusters differed mainly with respect to their length-scale parameter l_1 and l_2.

"irregular" years. The irregular years included years where there was a transition from El Niño to La Niña -- the year 1995 is perhaps particularly notable in this regard because it appears in the "irregular" group for three of the four stations -- and, more generally, are associated with cooling Pacific Ocean temperatures. These results suggest that cooling Pacific Ocean temperatures have some effect on kernel hyperparameters. The data were standardized to have a zero mean function, implying correlation between rainfall patterns month-to-month can discriminate here (as opposed to magnitude of rainfall).

The most obvious direction for future work is to apply the approach developed herein to other rainfall data. In terms of the BC coastal rainfall data, one could carry out a clustering of all locations combined to see whether some years stand out from others. The BC data could be treated as matrix variate data and clustered accordingly, perhaps after the fashion of Gallaugher and McNicholas (2018). In both cases, it is of interest to observe whether $G > 2$ clusters emerge.

ACKNOWLEDGEMENTS

The authors are grateful to an associate editor and anonymous reviewers for very helpful comments. This work was supported by the Canada Research Chairs program and a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada.

ORCID

P.D. McNicholas https://orcid.org/0000-0002-2482-523X

REFERENCES

Armal, S., Devineni, N., & Khanbilvardi, R. (2018). Trends in extreme rainfall frequency in the contiguous United States: Attribution to climate change and climate variability modes. *Journal of Climate*, *31*, 369–385.

Bates, D., & Eddelbuettel, D. (2013). Fast and elegant numerical linear algebra using the RcppEigen package. *Journal of Statistical Software*, *52*(5), 1–24.

Bin, S., & Wenlai, Y. (2013). *Application of Gaussian process regression to prediction of thermal comfort index*. Paper presented at: Proceedings of the 2013 IEEE 11th International Conference on Electronic Measurement Instruments (vol 2, pp. 958–961).

Bouveyron, C., & Brunet-Saumard, C. (2014). Model-based clustering of high-dimensional data: A review. *Computational Statistics and Data Analysis*, *71*, 52–78.

Cabral, R., Ferreira, A., & Friederichs, P. (2019). Space–time trends and dependence of precipitation extremes in North-Western Germany. *Environmetrics*, *2019*, e2605.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, *39*, 1–38.

Duvenaud, D., Lloyd, J., Grosse, R., Tenenbaum, J., & Zoubin, G. (2013). *Structure discovery in nonparametric regression through compositional kernel search*. Proceedings of the 30th International Conference on Machine Learning (vol 28, pp. 1166–1174).

Gallaugher, M. P. B., & McNicholas, P. D. (2018). Finite mixtures of skewed matrix variate distributions. *Pattern Recognition*, *80*, 83–93.

Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., & Hothornm T. (2009). *mvtnorm: Multivariate normal and T distributions*. R package version 1.0-10.

Gopalan, G., & Bornn, L. (2016). *FastGP: Efficiently Using Gaussian Processes with Rcpp and RcppEigen*. R package version 1.2.

Government of Canada (2007). Monthly precipitation data[dataset]. Retrieved from http://climate.weather.gc.ca/climate_data/monthly_data_e.html. Accessed May 08, 2018.

Gudmundson, C., & Babkina, A. M. (2003). *El Niño overview and bibliography*. Hauppauge, and New York, NY: Nova Science Publishers, Inc.

Gupta, U., Jitkajornwanich, K, Elmasri, R., & Fegaras, L. (2016). *Adapting K-means clustering to identify spatial patterns in storms*. Proceedings of the 2016 IEEE International Conference on Big Data (Big Data) (pp. 2646-2654).

Hewitt, J., Hoeting, J. A., Done, J. M., & Towler, E. (2018). Remote effects spatial process models for modeling teleconnections. *Environmetrics*, *29*, e2523.

McNicholas, P. D. (2016a). *Mixture model-based classification*. Boca Raton, FL: Chapman & Hall/CRC Press.

McNicholas, P. D. (2016b). Model-based clustering. *Journal of Classification*, *33*(3), 331–373.

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. Cambridge, MA: The MIT Press.

NOAA (2019). Cold & warm episodes by season. NOAA/ national weather service. National centers for environmental prediction. Retrieved from https://origin.cpc.ncep.noaa.gov/products/analysis_monitoring/ensostuff/ONI_v5.php. Accessed May 05, 2019.

O'Gorman, P. A. (2015). Precipitation extremes under climate change. *Current Climate Change Reports*, *1*, 49–59.

R Core Team. (2018). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.

Rasmussen, C. E. (2005). *Gaussian processes for machine learning*. Cambridge, MA: MIT Press.

Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N., and Aigrain, S. (2013). Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *371*, (1984), 20110550.

Tiedeman, D. V. (1955). *On the study of types*. In S. B. Sells (Ed.), *Symposium on pattern analysis*. Texas: Randolph Field, Air University, U.S.A.F. School of Aviation Medicine.

Ward, P. J., Jongman, B., Kummu, M., Dettinger, M. D., Sperna Weiland, F. C., & Winsemius, H. C. (2014). Strong influence of el nino Southern oscillation on flood risk around the world. *Proceedings of the National Academy of Sciences*, *111*(44), 15659–15664.

Wolfe, J. H. (1963). *Object cluster analysis of social areas* (Master's thesis). University of California, Berkeley.

How to cite this article: Paton F, McNicholas PD. Detecting British Columbia coastal rainfall patterns by clustering Gaussian processes. *Environmetrics*. 2020;31:e2631. https://doi.org/10.1002/env.2631