Bayesian online change point detection with Hilbert space approximate Student-t process

Jeremy Sellier [1] **Petros Dellaportas** [1 2 3]

Abstract

In this paper, we introduce a variant of Bayesian online change point detection with a reduced-rank Student-t process (TP) and dependent Student-t noise, as a nonparametric time series model. Our method builds and improves upon the state-of-the-art Gaussian process (GP) change point model benchmark of Saatçi et al. (2010). The Student-t process generalizes the concept of a GP and hence yields a more flexible alternative. Additionally, unlike a GP, the predictive variance explicitly depends on the training observations, while the use of an entangled Student-t noise model preserves analytical tractability. Our approach also uses a *Hilbert space* reduced-rank representation of the TP kernel, derived from an eigenfunction expansion of the Laplace operator (Solin & Särkkä, 2020), to alleviate its computational complexity. Improvements in prediction and training time are demonstrated with real-world data sets.

1. Introduction

Sequential data often exhibit instances of abrupt change in generative parameters. Failing to detect these specific *change points* at which the underlying distribution changes, significantly alters predictive performance of stationary parametric models. Change point detection (CPD) methods have proven useful in finance (Chib, 1998; Koop & Potter, 2004; Kummerfeld & Danks, 2013), quality control (Aroian & Levene, 1950), climate modelling (Manogaran & Lopez, 2018), cybersecurity (Polunchenko et al., 2012), genetics (Caron et al., 2012) and speech recognition (Panda & Nayak, 2016).

Bayesian Online change point Detection (BOCPD) was introduced simultaneously by Adams & MacKay (2007) and Fearnhead & Liu (2007). The first Bayesian approach to CPD relied on retrospective inference mostly via segmentation to generate samples from the posterior distribution over change point location (Barry & Hartigan, 1993; Green, 1995; Xuan & Murphy, 2007). In contrast, BOCPD allows online inference by generating a predictive distribution of the next datum given the data already observed.

In BOCPD, we assume a sequence of observations $\mathbf{x}_{1:T} = \{x_i\}_{i=1}^{T}$ that can be partitioned into sub-groups separated by possible change points. We are concerned with estimating recursively for every $t \in [1, T]$ the predictive distribution of the current *run length* r_t, namely the time since the last change point given the data observed so far i.e. $p(r_t|\mathbf{x}_{1:t})$ for $r_t \in [1, t]$. We need to define an *underlying predictive model* (UPM) to evaluate the posterior predictive distribution of the next datum given the possible previous run length i.e $p(x_t|r_{t-1}, \mathbf{x}_{1:t-1})$. The UPM can be seen as a base model, the parameters of which change for every run length. BOCPD has received considerable recent interest. Work has addressed performance improvement (Saatçi et al., 2010), model selection (Knoblauch & Damoulas, 2018), hyperparameter learning (Turner et al., 2009; Wilson et al., 2010; Caron et al., 2012) and change point prediction (Agudelo-España et al., 2019).

The original BOCPD algorithm makes the assumptions that observations are i.i.d. within each run length. Saatçi et al. (2010) extend BOCPD with a more flexible nonparametric UPM based on Gaussian processes (GPs). They propose a non-linear auto-regressive GP-based model (GPAR) and a time-deterministic GP model (GPTS) with change points, both of which improve predictive performance. The use of a GP comes at a cost of a $O(T^5)$ running complexity for naive implementation, prohibitive in most applications. Saatçi et al. (2010) introduce filtering techniques and computational tricks to reduce computation complexity.

In this paper, we build on the GP-based approach of Saatçi et al. (2010), introducing an alternative UPM based on a Student-t process (TP) with Student-t noise (Shah et al.,

[1]Department of Statistical Science, University College London, UK [2]Department of Statistics, Univ. of Econ. and Business, Athens, Greece [3]The Alan Turing Institute, UK. Correspondence to: Jeremy Sellier <jeremy.sellier.18@ucl.ac.uk>, Petros Dellaportas <p.dellaportas@ucl.ac.uk>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

2014) and Hilbert space reduced-rank kernel proposed by Solin & Särkkä (2020). Benefiting from its fatter tails, a TP offers inherent robustness against outliers, surpassing GPs in this regard. Specifically, in the context of BOCPD, TPs exhibit a lower propensity for generating false alarms when detecting change points caused by outliers. Additionally, TPs offer more adaptive predictive variance in comparison to GPs, adjusting more effectively to the variance of past observations. We will explore this aspect further in Section 2.3. Lastly, a TP introduces greater flexibility compared to a GP, as it represents the most general elliptical process with a tractable density (Shah et al., 2014)

The first mention of a TP can be found in Rasmussen & Williams (2005) and early applications in Archambeau & Bach (2011) and Yu et al. (2007). However, Rasmussen & Williams (2005) concluded that a TP is not practicable, due to the intractability of the posterior when adding noise (since the Student-t distribution is not closed under addition). TPs have received greater recent attention since Shah et al. (2014) proposed a derivation from a Wishart prior, and introduced a dependent Student-t noise preserving tractability. The benefit of a TP compared to a GP has since been demonstrated for regression (Shah et al., 2014; Tang et al., 2016; 2017; Li & Ma, 2021), state-space models (Solin & Särkkä, 2015) and Bayesian optimization (Tracey & Wolpert, 2018).

To overcome the GP computational complexity, several schemes have been proposed in the literature. Reduced-rank approximation methods which approximate the kernel Gram matrix with another matrix of smaller rank have been popular (see Chapter 8 Rasmussen & Williams, 2005). Common examples include the *Nyström* method (see Williams & Seeger, 2001) and *Random Fourier Features* (Rahimi & Recht, 2007). Solin & Särkkä (2020) introduced an *Hilbert space* method for reduced-rank which approximates the eigendecomposition of stationary kernels in terms of an eigenfunction expansion of the Laplace operator. In their original paper, Solin & Särkkä (2020) adapt the method for a GP approximation referred as HSGPs, that has been used in the context of GP regression (Solin & Särkkä, 2020; Riutort-Mayol et al., 2022) and GP-based state-space models (Svensson et al., 2016; Svensson & Schön, 2017). The choice of an *Hilbert space* approach is particularly convenient in a BOCPD context. In the approximation, features vectors are independent of the covariance function, yielding computational advantages detailed later.

Our method is referred to as HSSPAR, an abbreviation for Hilbert Space t-Student Process Auto-Regressive Model, maintaining consistency with previous naming convention. Combining a Student-t process predictor model and Hilbert space reduced-rank kernels, HSSPAR shows systematic improvement in predictive performance and hyperparameter learning time for real-world data sets presented in Section 5.

2. Model

In this section, we first provide a brief reminder of the original BOCPD algorithm for change point detection proposed by Adams & MacKay (2007).

2.1. BOCPD Algorithm

Given a *hazard function* and a UPM, inference is done recursively at every time step using

$$\begin{aligned} p(r_t|\mathbf{x}_{1:t}) &\propto \sum_{r_{t-1}} p(r_t|r_{t-1})p(x_t|r_{t-1},\mathbf{x}_{1:t-1})p(r_{t-1}|\mathbf{x}_{1:t-1}) \\ &\propto \sum_{r_{t-1}} \underbrace{p(r_t|r_{t-1})}_{\text{Hazard}} \underbrace{p(x_t|\mathbf{x}_{t-1}^{(r)})}_{UPM} p(r_{t-1}|\mathbf{x}_{1:t-1}) \end{aligned} \tag{1}$$

where $\mathbf{x}_{t-1}^{(r)}$ indicates the last r_{t-1} observations prior to x_t. The normalizing constant of $p(r_t|\mathbf{x}_{1:t})$ in Equation (1) is obtained by summing up all its evaluation instances, since r_t is a discrete random variable.

The marginal predictive distribution is then obtained by

$$p(x_t|\mathbf{x}_{1:t-1}) = \sum_{r_{t-1}} p(x_t|\mathbf{x}_{t-1}^{(r)})p(r_{t-1}|\mathbf{x}_{1:t-1}). \tag{2}$$

The conditional prior $p(r_t|r_{t-1})$ is defined as

$$p(r_t|r_{t-1}) = \begin{cases} H(r_{t-1}) & \text{if } r_t = 0 \\ 1 - H(r_{t-1}) & \text{if } r_t = r_{t-1} + 1 \\ 0 & \text{otherwise} \end{cases}$$

where the *hazard function* $H(t)$ verifies

$$H(\tau) = \frac{P_{\text{change}}(\tau)}{\sum_{t=\tau}^{\infty} P_{\text{change}}(t)}$$

and P_{change} denotes the probability distribution over the interval between change points. A simple case arises where $P_{\text{change}}(\cdot)$ is a discrete exponential geometric distribution with scale parameter $1/h$, which yields $H(\tau) = h$.

The original BOCPD algorithm (Adams & MacKay, 2007) makes the assumptions that the observations are i.i.d within each run length with respect to an exponential family distribution. In their experiments, the authors adopt Gaussian i.i.d. assumptions with a Normal-Inverse-Gamma prior on parameters (i.e. yielding a Student-t predictive). Saatçi et al. (2010) used a GP UPM where the time index serves as input (GPTS) and an auto-regressive GP UPM (GPAR)

of order p which takes values $\mathbf{x}_{t-p:t-1}$ as input at time t. GPTS effectively utilizes time as an index, enabling it to accommodate irregular time intervals. Moreover, GPTS has been shown to possess an equivalent linear autoregressive (AR) representation (Murray-Smith & Girard, 2001). In contrast, although GPAR imposes a uniform time step constraint, it generalizes GPTS in its ability to handle nonlinearity. Consequently, GPAR is capable of modeling more complex data dynamics.

2.2. Student-t Process (TP)

We review the properties of the Student-t distribution and process, which serves in later sections as our predictive model.

Definition 2.1. An n-dimensional vector $\mathbf{y}$ is multivariate Student-t-distributed with ν degrees of freedom, mean vector $\boldsymbol{\mu} \in \mathbb{R}^n$ and covariance matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$, if its joint probability density is given by

$$St(\mathbf{y}|\boldsymbol{\mu},\mathbf{K},\nu) = \frac{\Gamma((\nu+n)/2)}{\Gamma(\nu/2)((\nu-2)\pi)^{n/2}|\mathbf{K}|^{1/2}}\times \left(1+\frac{1}{\nu-2}(\mathbf{y}-\boldsymbol{\mu})^\top \mathbf{K}^{-1}(\mathbf{y}-\boldsymbol{\mu})\right)^{-\frac{\nu+n}{2}}. \quad (3)$$

As for the Gaussian distribution, the conditional distribution for a multivariate Student-t has an analytical form. The following result can be found in Kotz & Nadarajah (2004) and Shah et al. (2014).

Lemma 2.2. *Let* $\mathbf{y} \sim St(\mathbf{y}|\boldsymbol{\mu},\mathbf{K},\nu)$ *and partition* $\mathbf{y}$ *into two sub-vectors* $\mathbf{y}_1 \in \mathbb{R}^{n_1}$ *and* $\mathbf{y}_2 \in \mathbb{R}^{n_2}$ *such that* $\boldsymbol{\mu}_p = \mathbb{E}[\mathbf{y}_p]$ *and* $\mathbf{K}_{p|p} = cov[\mathbf{y}_p,\mathbf{y}_p]$ *for* $p = 1, 2$. *Then the conditional density for* $\mathbf{y}_1|\mathbf{y}_2$ *has an analytical form* $\mathbf{y}_1|\mathbf{y}_2 \sim St(\boldsymbol{\mu}_{1|2},\mathbf{K}_{1|2},\nu_{1|2})$ *with* $\boldsymbol{\mu}_{1|2} = \mathbf{K}_{1,2}\mathbf{K}_{2,2}^{-1}(\mathbf{y}_2-\boldsymbol{\mu}_2)+\boldsymbol{\mu}_1$, *covariance* $\mathbf{K}_{1|2} = \frac{\nu-2+\beta}{\nu-2+n_2}(\mathbf{K}_{1,1}-\mathbf{K}_{1,2}\mathbf{K}_{2,2}^{-1}\mathbf{K}_{2,1})$, $\beta = (\mathbf{y}_2-\boldsymbol{\mu}_2)^\top\mathbf{K}_{22}^{-1}(\mathbf{y}_2-\boldsymbol{\mu}_2)$ *and* $\nu_{1|2} = \nu+n_2$ *degrees of freedom.*

As described in Shah et al. (2014), the Student-t process (TP) can be derived by placing an inverse Wishart process prior on the kernel function of a GP. We provide more information in Appendix A.

Definition 2.3. A random real-valued function $f : \mathcal{X} \to \mathbb{R}$ is said to follow a Student-t process $f \sim \mathcal{TP}(\mu,k,\nu)$, with ν degrees of freedom, mean function $\boldsymbol{\mu} \in \mathcal{X}$ and covariance function $k : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$, if any collection of function values has a joint multivariate Student-t distribution such that

$$(f(x_1),\cdots,f(x_n)) \sim St(\boldsymbol{\mu},\mathbf{K},\nu) \quad (4)$$

where $\mathbf{K}$ is a covariance matrix with entries $\mathbf{K}_{i,j} = k(x_i,x_j)$ for $i,j = 1,\cdots,n$.

Student-t Noise Model Unfortunately, with a TP, adding Student-t noise removes analytical tractability. To overcome this issue, Shah et al. (2014) and Zhang & Yeung (2010) propose to add an uncorrelated but dependent noise term, which preserves tractability.

We assume each observation in $\mathbf{y} = \{y_i\}_{i=1}^n$ is to be modelled from a latent process $\mathbf{f} = \{f(x_i)\}_{i=1}^n$ and a noise vector $\boldsymbol{\varepsilon} = \{\varepsilon_i\}_{i=1}^n$ such that

$$y_i = f_i + \varepsilon_i \qquad \text{for } i = 1,\cdots,n \quad (5)$$

where

$$\begin{bmatrix}\mathbf{f}\\ \boldsymbol{\varepsilon}\end{bmatrix} \sim St\left(\mathbf{0}_{2n}, \begin{pmatrix}\mathbf{K} & 0\\ 0 & \sigma_n^2\mathbf{I}_n\end{pmatrix}, \nu\right). \quad (6)$$

From the properties of the Student-t distribution, $\mathbf{y} = \mathbf{f}+\boldsymbol{\epsilon} \sim \mathcal{TP}(\mathbf{0}_n, \mathbf{K}+{\sigma_n}^2\mathbf{I}_n, \nu)$. Therefore, we obtain a tractable distribution for $\mathbf{y}$, simply incorporating the noise variance into the kernel. Note that $\mathbf{f}$ and $\boldsymbol{\varepsilon}$ in Equation (6) are not independent since the scaling parameter ν has an effect on both the covariances of $\mathbf{f}$ and $\boldsymbol{\varepsilon}$.

Tang et al. (2016) gives a probabilistic interpretation to this noise incorporation. Equation (6) can be shown to be equivalent to a noise model following

$$p(\boldsymbol{\varepsilon}|\mathbf{f}) \sim St\left(\boldsymbol{\varepsilon}\middle|\mathbf{0}, \frac{\sigma_n^2}{\nu+n}\left(\nu+\mathbf{f}^\top\mathbf{K}^{-1}\mathbf{f}\right)\mathbf{I}, \nu+n\right). \quad (7)$$

Thus, the variance of the noise model adjusts to the data fit term $\mathbf{f}^\top\mathbf{K}^{-1}\mathbf{f}$ present in the noise-free model marginal log likelihood. This means that when the noise-free model fits the data well, the added noise will have a smaller variance, and vice versa.

Relation to GPs A TP can be seen as a generalization of a GP. As the parameter ν approaches infinity, the TP converges to the GP in the following sense: If we have $f \sim \mathcal{TP}(\mu,k,\nu)$, where μ represents the mean function and k denotes the covariance function, then the distribution of f tends towards $\mathcal{GP}(\mu,k)$ as ν tends to infinity (Shah et al., 2014, Lemma 2). A TP is in fact the most general elliptical process with an analytically-representable density (Shah et al., 2014). Furthermore, Tang et al. (2016) argue that a TP with noise incorporated in the kernel as in Equations (6) and (7) tends to a GP with i.i.d Gaussian noise as $\nu \to \infty$.

2.3. BOCPD with TP-based UPM

We propose a BOCPD extension where the UPM is based on a TP process with Student-t noise. We first introduce a TP auto-regressive model of order p where at time t, the past p values $\mathbf{x}_{t-p:t-1}$ are taken as input and x_t as the ob-

Algorithm 1 BOCPD run length estimation

$(\Xi_0, \nabla_h \Xi_0, \nabla_\theta \Xi_0) \leftarrow (1, 0, 0)$ ▷ Initialize the recursion, set hazard and UPM derivatives to 0.
Compute the eigenfunctions evaluation $\mathbf{\Phi}$
Define $\tilde{\Xi}_t$ as $\Xi_t[2 : t+1]$
for $t = 1$ **to** T **do**
  $(\boldsymbol{\pi}_t, \nabla_\theta \boldsymbol{\pi}_t) \leftarrow \texttt{UPM\_predictive}(\mathbf{x}_t, t, \mathbf{\Phi})$
  $\mathbf{h} \leftarrow H(1:t)$
  Update growth probabilities:
  $\tilde{\Xi}_t \leftarrow \Xi_{t-1} \odot \boldsymbol{\pi}_t \odot (1 - \mathbf{h})$
  $\nabla_\theta \tilde{\Xi}_t \leftarrow (1 - \mathbf{h}) \odot (\nabla_\theta \Xi_{t-1} \odot \boldsymbol{\pi}_t + \nabla_\theta \boldsymbol{\pi}_t \odot \Xi_{t-1})$
  $\nabla_h \tilde{\Xi}_t \leftarrow \boldsymbol{\pi}_t \odot (\nabla_h \Xi_{t-1} \odot (1 - \mathbf{h}) - \Xi_{t-1} \odot \nabla_h \mathbf{h})$
  Update change point probabilities:
  $\Xi_t[1] \leftarrow \sum \Xi_{t-1} \odot \boldsymbol{\pi}_t \odot \mathbf{h}$
  $\nabla_\theta \Xi_t[1] \leftarrow \sum \mathbf{h} \odot (\nabla_\theta \Xi_{t-1} \odot \boldsymbol{\pi}_t + \nabla_\theta \boldsymbol{\pi}_t \odot \Xi_{t-1})$
  $\nabla_h \Xi_t[1] \leftarrow \sum \boldsymbol{\pi}_t \odot (\nabla_h \Xi_{t-1} \odot \mathbf{h} + \Xi_{t-1} \odot \nabla_h \mathbf{h})$
  Perform prediction:
  $p(r_t | \mathbf{x}_{1:t-1}) \leftarrow$ normalized Ξ_t
end for
$p(\mathbf{x}_{1:T}) = \sum \Xi_T$ ▷ Compute the evidence
$\nabla p(\mathbf{x}_{1:T}) = (\sum \nabla_h \Xi_T, \nabla_\theta \Xi_T)$
return $(p(\mathbf{x}_{1:T}), \nabla p(\mathbf{x}_{1:T}))$

servation, i.e.

$$x_t = f(\mathbf{x}_{t-p:t-1}) + \varepsilon_t \tag{8}$$

where $f \sim \mathcal{TP}(0, k, \nu)$ and ε is a dependent Student-t noise with scale parameter σ_n described in Equation (6).

Interestingly, by Lemma 2.2, we can marginalize out the latent f, to yield an marginal predictive distribution. This yields an auto-regressive TP-based UPM of the form

$$p(x_t | \mathbf{x}_{t-r:t-1}) \sim St(x_t | m_{t,r}, v_{t,r}, \nu + r - 1) \tag{9}$$

where

$$\begin{aligned} m_{t,r} &= \mathbf{k}_*^\top \tilde{\mathbf{K}}^{-1} \mathbf{x}_{t,r} \\ v_{t,r} &= \alpha_{t,r} \left(k(x_t, x_t) - \mathbf{k}_*^\top \tilde{\mathbf{K}}^{-1} \mathbf{k}_* \right) \\ \alpha_{t,r} &= \frac{v - 2 + \beta_{t,r}}{v - 3 + r} \\ \beta_{t,r} &= \mathbf{x}_{t,r}^\top \tilde{\mathbf{K}}^{-1} \mathbf{x}_{t,r}. \end{aligned} \tag{10}$$

Here $\mathbf{x}_{t,r} = \mathbf{x}_{t-r+1:t-1}$, $\tilde{\mathbf{K}}_{i,j} = k(\mathbf{x}_{i-p+1:i}, \mathbf{x}_{j-p+1:j}) + \sigma_n^2 \delta_{i,j}$ for $i, j = t-r, \cdots, t-2$, $\delta_{i,j}$ denotes the Kronecker delta and $\mathbf{k}_*$ is an $(r-1)$-dimensional vector with the ith entry being $k(\mathbf{x}_{t-p:t-1}, \mathbf{x}_{i-p+1:i})$ for $i = t-r, \cdots, t-2$.

The predictive mean $m_{t,r}$ has the same form as for a GP (assuming the same kernel and hyperparameters). However, due to the differing marginal likelihood between TP and GP, the predictive mean differs after learning the hyperparameters. Unlike a GP, the TP model exhibits more adaptive predictive volatility based on the training observations. The parameter $\beta_{t,r}$ explicitly depends on $\mathbf{x}_{t,r}$. When $\beta_{t,r}$ exceeds $(r-1)$, TP's predictive variance surpasses that of a GP, and vice versa. In fact, assuming $\mathbf{x}_{t,r}$ is drawn from a GP prior $\mathcal{N}(0, \tilde{\mathbf{K}})$, $\beta_{t,r}$ follows a Chi-squared distribution with mean $(r-1)$. Consequently, if observations have similar variance as expected under a GP prior, TP's covariance is comparable to that of a GP. However, significantly larger or smaller variability in the observations leads to higher or lower posterior uncertainty in TP, respectively.

2.4. Hilbert Space Approximate Student-t Process

The TP UPM in Equation (9) inherits the same cubic computational cost of GPs, which is prohibitive for most applications. We propose a reduced-rank implementation of the Student-t Process UPM based on the novel *Hilbert space* method for reduced-rank kernel approximation (Solin & Särkkä, 2020). Solin & Särkkä (2020) obtain approximate eigendecompositions of stationary covariance functions in terms of an eigenfunction expansion of the Laplace operator in a compact subset of $\mathbb{R}^d$.

The *Hilbert space reduced-rank* method provides a different advantage in our case compared to other reduced-rank approximations :

(i) The Laplace-based feature vectors are independent of the particular choice of kernel, including the kernel hyperparameters. Gradient computation is thus facilitated, which in turn speeds up the learning phase. We refer to Section 4 for more details.

(ii) The decay of the expansion coordinates is fast. Hence, a good approximation can be obtained with relatively few basis points. As an example, Solin & Särkkä (2020) obtains a good approximation to univariate RBF kernels with only 12 eigenfunctions. They argue that adding more eigenfunctions has negligible effect on the approximation accuracy.

Hilbert Space Reduced-Rank Kernel For the univariate case with observations within a closed interval $\Omega = [-L, L] \subset \mathbb{R}$, where L is some positive real number, we can approximate k with hyperparameters θ with a kernel representation

$$k_\theta(x, x') = \sum_{j=1}^{\infty} S_\theta(\sqrt{\lambda_j}) \phi_j(x) \phi_j(x') \quad \forall x, x' \in \Omega \tag{11}$$

where S is the spectral density of the stationary kernel k, $\{\phi_j\}_{j=1}^\infty$ and $\{\lambda_j\}_{j=1}^\infty$ are the sets of eigenfunctions and eigenvalues of the Laplace operator ∇^2 in Ω. A short review of the technical details surrounding the approximation derivation is provided in Appendix B.

In particular, for a Gaussian kernel $k(x - x') = \sigma^2 \exp\left(-(x - x')^2 / 2\,\ell\right)$ with scaling parameter σ and

length-scale parameter ℓ, the corresponding spectral density is defined as $S_\theta(w) = \sigma\sqrt{2\pi\ell^2}\exp(-\frac{\pi^2\ell^2 w^3}{2})$.

The eigenvalues $\{\lambda_j\}_{j=1}^{\infty}$ and eigenfunctions $\{\phi_j\}_{j=1}^{\infty}$ follow

$$\lambda_j = \left(\frac{\pi j}{2L}\right)^2, \tag{12}$$

$$\phi_j(x) = \frac{1}{\sqrt{L}}\sin\left(\sqrt{\lambda_j}(x+L)\right). \tag{13}$$

The multivariate formulation of the decomposition is provided in Appendix B. Note that the total number of basis function grows exponentially with the number of basis functions per dimension, a problem inherent in most choices of basis-function expansion.

The eigenvalues λ_j are monotonically increasing with j, and for a bounded kernel, the spectral density $S(\cdot)$ tends to zero quickly at higher frequencies. Thus, a good approximation is obtained by truncating the expansion in Equation (11) to the first m terms. We can form an approximate eigendecomposition of the covariance matrix

$$\mathbf{K} \approx \mathbf{\Phi}^\top \mathbf{\Lambda} \mathbf{\Phi} \tag{14}$$

where $\mathbf{\Lambda}$ is a diagonal matrix with entries $\{S_\theta(\sqrt{\lambda_j})\}_{j=1}^m$ and $\mathbf{\Phi}$ is a matrix of eigenfunction evaluations such that $\mathbf{\Phi}_{i,j} = \phi_i(x_j)$. The quality on the approximation also relies on the choice of closed interval $[-L, L]$. Solin & Särkkä (2020) simply normalized the data and adjust L accordingly.

2.5. BOCPD with Hilbert Space Approximate TP UPM

Using the reduced-rank Hilbert method, the predictive distribution of Equation (9) becomes $p(x_t|\mathbf{x}_{t-r:t-1}) \sim St(x_t|m_{t,r}, v_{t,r}, \nu + r - 1)$. Further, using the Woodbury matrix inversion formula

$$\begin{aligned} m_{t,r} &= \phi(x_{t-1})^\top \mathbf{Q}_{t,r}\mathbf{\Phi}_{t,r}\mathbf{x}_{t,r} \\ v_{t,r} &= \alpha_{t,r}\left(\phi(x_{t-1})^\top \mathbf{Q}_{t,r}\phi(x_{t-1})\right) \\ \alpha_{t,r} &= \frac{(\nu-2)\sigma_n^2 + \beta_{t,r}}{\nu - 3 + r} \\ \beta_{t,r} &= ||\mathbf{x}_{t,r}||_2^2 - \mathbf{x}_{t,r}\mathbf{\Phi}_{t,r}^\top \mathbf{Q}_{t,r}\mathbf{\Phi}_{t,r}\mathbf{x}_{t,r} \end{aligned} \tag{15}$$

with $\mathbf{x}_{t,r} = \mathbf{x}_{t-r+1:t-1}$. Here, $\mathbf{\Phi}_{t,r}$ is a $m \times (r-1)$ matrix of eigenfunctions with i, j entry $\phi_i(\mathbf{x}_j)$ for $i = 1, \cdots, m$ and $j = t-r, \cdots, t-2$, and $\mathbf{Q}_{t,r}$ is the $m \times m$ precision matrix such that

$$\mathbf{Q}_{t,r} = \left(\mathbf{\Phi}_{t,r}\mathbf{\Phi}_{t,r}^\top + \sigma_n^2\mathbf{\Lambda}^{-1}\right)^{-1}. \tag{16}$$

Algorithm 2 HSSPAR-CP UPM implementation

Function `UPM_predictive`$(\mathbf{x}_t, t, \mathbf{\Phi})$:
Inputs(from previous iteration): $r_{\max} = |r_{t-1}|$, $\mathbf{Q}_{t-1,r_{\max}}$, $\mathbf{u}_{t-1} := \mathbf{\Phi}_{t-1,r_{\max}}\mathbf{x}_{t-1,r_{\max}}$
$\mathbf{Q} \leftarrow$ `h_update`$(\mathbf{Q}_{t-1,r_{\max}}, \phi(x_{t-2}))$ ▷ Eq.(19)
$\mathbf{u} \leftarrow \mathbf{u}_{t-1} + x_{t-1}\phi(x_{t-2})$
if $r_{\max} + 1 > R_{\max}$ **then**
  $\mathbf{Q} \leftarrow$ `h_downdate`$(\mathbf{Q}, \phi(x_{t-r_{\max}}))$ ▷ Eq.(20)
  $\mathbf{u} \leftarrow \mathbf{u} - x_{t-r_{\max}}\phi(x_{t-r_{\max}-1})$
end if
$r_{\max} \leftarrow \min(R_{\max}, r_{\max}+1)$
$\mathbf{Q}_{t,r_{\max}} \leftarrow \mathbf{Q}$
$\mathbf{u}_t \leftarrow \mathbf{u}$
for $r = r_{\max}$ **to** $t = 1$ **do**
  $m_{t,r} \leftarrow \phi(x_{t-1})^\top \mathbf{Q}\mathbf{u}$
  $\beta_{t,r} \leftarrow ||\mathbf{x}_{t,r}||_2^2 - \mathbf{u}^\top \mathbf{Q}\mathbf{u}$
  Compute $\alpha_{t,r}$ and $v_{t,r}$ ▷ Eq (15)
  $\pi_{t,r} \leftarrow p(x_t|\mathbf{x}_{t-r:t-1})$ ▷ Eq.(9)
  Compute $\nabla m_{t,r}, \nabla\beta_{t,r}, \nabla\alpha_{t,r}$ and $\nabla v_{t,r}$
  Compute $\nabla\pi_{t,r}$ with chain rule
  $\mathbf{Q} \leftarrow$ `v_update`$(\mathbf{Q}, \phi(x_{t-r}))$ ▷ Eq.(18)
  $\mathbf{u} \leftarrow \mathbf{u} - x_{t-r}\phi(x_{t-r-1})$
end for
return $(\boldsymbol{\pi}_t, \nabla\boldsymbol{\pi}_t)$.

3. Implementation Details

As noted in Equation (12), the basis function in the reduced-rank approximation does not depend on covariance function hyperparameters. Thus the eigenfunctions can be evaluated once and stored in a cached $T \times m$ matrix $\mathbf{\Phi}$ through the learning process, with $O(Tm)$ space complexity. If the number of observations T is so large that storing is not feasible, evaluation can be carried out in blocks or only when necessary.

Pruning the run length distribution In a naive implementation, all the run lengths are retained and the posterior $p(r_t|\mathbf{x}_{1:t})$ for $r_t = \{1, \cdots, t\}$ forms a vector of size t at every update step. In practice the run length distribution is highly peaked. A modification of the algorithm is to *prune* out the run length probability estimates with a total mass below a certain threshold, i.e $\leq 1/R_{\max}$; or to only consider the $R_{\max}$ most probable values, i.e $|r_t| \leq R_{\max}$ (Adams & MacKay, 2007; Saatçi et al., 2010; Knoblauch & Damoulas, 2018). This yields a running complexity of $O(TR_{\max}^2 m^2)$ for the reduced-rank TP-based UPM.

Vertical Rank-One Update We can improve the implementation further, by performing a rank-one update of the precision matrix $\mathbf{Q}_{t,r}$ across run lengths. Indeed, at time t, the product $\mathbf{\Phi}_{t,r}\mathbf{\Phi}_{t,r}^\top$ in Equation (16) can be updated across run lengths as

$$\mathbf{\Phi}_{t,r}\mathbf{\Phi}_{t,r}^\top = \mathbf{\Phi}_{t,r-1}\mathbf{\Phi}_{t,r-1}^\top + \phi(x_{t-r})\phi(x_{t-r})^\top. \tag{17}$$

Thus, knowing $\mathbf{Q}_{t,r_{\max}}$, where $r_{\max}$ stands for the maximum run length size at time t, we can use the Sherman-Morrison

inversion formula to obtain the following recursion

$$\mathbf{Q}_{t,r-1} = \left(\mathbf{I}_m + \frac{\mathbf{Q}_{t,r}\phi(x_{t-r})\phi(x_{t-r})^\top}{1-\phi(x_{t-r})^\top\mathbf{Q}_{t,r}\phi(x_{t-r})}\right)\mathbf{Q}_{t,r}. \tag{18}$$

Equation (18) can be implemented as an outer product of two matrix-vector products. To make the evaluation fast, we used the specialized BLAS routines for rank-one update (i.e. the scipy method `linalg.blas.dger` for Python). This rank-one update of $\mathbf{Q}_{t,r}$, together with an efficient update of the product $\mathbf{\Phi}_{t,r}\mathbf{x}_{t,r}$ in Equation (15) yields a running complexity of $O(TR_{\max}m^2)$.

Horizontal Rank-One Update We can also perform a *horizontal* update of the precision matrices across time t. Let $\mathbf{Q}_{t,r_{\max}}$ denote the precision matrix associated with the largest run length at time t (i.e. for $r_t = |r_t|$). Using Equation (17), we obtain

$$\mathbf{Q}_{t,r_{\max}} = \left(\mathbf{I}_m - \frac{\mathbf{Q}_{t-1,r_{\max}}\phi(x_{t-2})\phi(x_{t-2})^\top}{1+\phi(x_{t-2})^\top\mathbf{Q}_{t-1,r_{\max}}\phi(x_{t-2})}\right)\mathbf{Q}_{t-1,r_{\max}}. \tag{19}$$

To maintain consistency with *pruning*, an additional rank-one downdate is necessary when $|r_{t-1}|+1 > R_{\max}$, to remove the information carried by $\phi(x_{t-|r_{t-1}|})$, as given below

$$\mathbf{Q}_{t,r_{\max}} = \left(\mathbf{I}_m + \frac{\mathbf{Q}_{t,r_{\max}}\phi(x_{t-|r_{t-1}|})\phi(x_{t-|r_{t-1}|})^\top}{1-\phi(x_{t-|r_{t-1}|})^\top\mathbf{Q}_{t,r_{\max}}\phi(x_{t-|r_{t-1}|})}\right)\mathbf{Q}_{t,r_{\max}}. \tag{20}$$

Maximum A-Posteriori (MAP) Segmentation For the identification of change points, we used a variation of the *MAP segmentation* algorithm proposed by Knoblauch & Damoulas (2018). We compute MAP_t, an estimator of the density of the run length MAP estimate before t with the recursion

$$\text{MAP}_t = \max_r\{p(r_t = r|\mathbf{x}_{1:t})\text{MAP}_{t-r-1}\}. \tag{21}$$

For r_t^*, the maximizer of Equation (21) at time t, the MAP segmentation is $S_t = S_{t-r_t^*-1}\cup\{(t-r_t^*)\}$, $S_0 = \emptyset$, where $t' \in S_t$ means a CP occurs at $t' \geq t$.

4. Hyperparameter Learning

Following Saatçi et al. (2010), the hyperparameters $\Theta := (\theta, \nu, \sigma_n)$ where θ refers to the kernel hyperparameters, are learned by minimizing the marginal negative log likelihood

$$\log p(\mathbf{x}_{1:T}|\Theta) = -\sum_{i=1}^{T}\log p(x_t|\mathbf{x}_{1:t-1},\Theta). \tag{22}$$

Saatçi et al. (2010) optimize the hyperparameters on a test subset $\{\mathbf{x}_{1:T'}\}$ by running the BOCPD multiple times to find $\tilde{\Theta} = \arg\min_\Theta\{\log p(\mathbf{x}_{1:T'}|\Theta)\}$. The gradient of the log likelihood is obtained from the gradient of the one-step-ahead predictor gradients $\nabla p(x_t|\mathbf{x}_{1:t-1})$. The terms $\nabla p(x_t|\mathbf{x}_{1:t-1})$ are themselves computed by iteratively calculating the gradient of the UPM, $\nabla p(x_t|\mathbf{x}_{t-r:t-1})$, the gradient of the hazard rate $\nabla p(r_t|r_{t-1})$ and then propagating forward using the chain rule (Saatçi et al., 2010). These computations are consistent with hyperparameter learning in other on-line GP methods (Ranganathan et al., 2011).

For GP-based UPM, the computation and forward propagation of the gradient is particularly expensive and accounts for most of the training time. In our case, computation of the UPM gradient is easier since the feature vectors $\mathbf{\Phi}_{t,r}$ are independent of the hyperparameters Θ. We are left from Equation (15) with

$$\begin{aligned}
\nabla_\Theta m_{t,r} &= \phi(x_{t-1})^\top\nabla_\Theta\mathbf{Q}_{t,r}\mathbf{\Phi}_{t,r}\mathbf{x}_{t,r}\\
\nabla_\Theta v_{t,r} &= \frac{v_{t,r}}{\alpha_{t,r}}\nabla\alpha_{t,r} + \alpha_{t,r}\left(\phi(x_{t-1})^\top\nabla_\Theta\mathbf{Q}_{t,r}\phi(x_{t-1})\right)\\
\nabla_\Theta\beta_{t,r} &= -\mathbf{x}_{t,r}\mathbf{\Phi}_{t,r}^\top\nabla_\Theta\mathbf{Q}_{t,r}\mathbf{\Phi}_{t,r}\mathbf{x}_{t,r}
\end{aligned} \tag{23}$$

where

$$\begin{aligned}
\nabla_\theta\mathbf{Q}_{t,r} &= \sigma_n^2\mathbf{Q}_{t,r}\left(\mathbf{\Lambda}^{-2}\nabla_\theta\mathbf{\Lambda}\right)\mathbf{Q}_{t,r}\\
\nabla_\nu\mathbf{Q}_{t,r} &= 0\\
\nabla_{\sigma_n}\mathbf{Q}_{t,r} &= \frac{1}{2\sigma_n}\mathbf{Q}_{t,r}\mathbf{\Lambda}^{-1}\mathbf{Q}_{t,r}.
\end{aligned} \tag{24}$$

The term $\mathbf{\Lambda}^{-2}\nabla_\theta\mathbf{\Lambda}$ in Equation (24) is independent of the observations and thus can be computed once at the beginning of each optimizing step and reused throughout the BOCPD iterations. Equation (23) and (24) provide a simple computational routine for the gradient, once the precision matrix $\mathbf{Q}_{t,r}$ update is obtained. The gradient of the UPM, $\nabla p(x_t|\mathbf{x}_{t-r:t-1})$ is then derived from the gradient of UPM parameters.

5. Experiments

We compare our scheme to the two GP-based UPM variants introduced in Saatçi et al. (2010), namely ARGP and GPTS. We also include as baseline the normal i.i.d UMP (TIM) of Adams & MacKay (2007). We use the acronyms HSSPAR-CP to refer to our reduced-rank Student-t process-based UPM as in Equation (15), and HSGPAR-CP for an equivalent reduced-rank GP-based

Table 1: Results of predictive performance on Nile Data, Well Log Data, Bee Waggle Data and Whistler Snowfall Data. The results are provided with 95% error bars[1] and the p-value testing the null hypothesis that methods are equivalent to the best performing method, according to NLL, using one sided t-test. (·)-CP refers to the BOCPD variant of the respective method.

Method	Negative Log Likelihood	p-value	MSE	p-value	time(s)
Nile Data (200 training points, 463 Test points)					
HSGPAR-CP	1.1480 (± 0.0564)	0.072	0.5756 (±0.0977)	0.480	43.18
HSSPAR-CP	**1.0984** (± 0.0653)	N/A	0.5783 (±0.0995)	N/A	44.52
GPTS	1.2313 (±0.0449)	< 0.001	0.6050 (±0.0942)	0.301	2.74
GPTS-CP	1.1468 (±0.0533)	0.067	**0.5381** (±0.0890)	0.208	5.20
GPAR	1.1729 (±0.0527)	0.020	0.5587 (±0.0978)	0.355	142.66
GPAR-CP	1.1481 (±0.0587)	0.079	0.5792 (±0.0964)	0.493	267.17
TIM	1.1769 (±0.0852)	0.065	0.6644 (±0.1029)	0.081	N/A
Well Log Data (1000 training points, 3047 Test points)					
HSGPAR-CP	0.1927 (±0.0343)	0.390	**0.1165** (±0.0109)	0.312	528.75
HSSPAR-CP	**0.1875** (±0.0321)	N/A	0.1194 (±0.0123)	N/A	659.20
GPTS	0.5557 (±0.0480)	< 0.001	0.1575 (±0.0199)	0.007	17.88
GPTS-CP	0.2489 (±0.0446)	< 0.001	0.1201 (± 0.0115)	0.460	78.24
GPAR	0.3001 (±0.0383)	< 0.001	0.1704 (±0.0380)	0.023	11,596.64
GPAR-CP	0.1926 (±0.0342)	0.392	0.1166 (±0.0110)	0.316	13,610.75
TIM	0.2562 (±0.0287)	0.003	0.1921 (±0.0275)	0.002	N/A
Bee Waggle Data (250 training points, 806 Test points)					
HSGPAR-CP	-0.9249 (±0.1574)	0.006	0.8623 (±0.1670)	0.034	225.63
HSSPAR-CP	**-1.2291** (±0.1099)	N/A	**0.6646** (±0.1071)	N/A	315.41
GPTS	1.2786 (±0.2440)	< 0.001	1.6688 (±0.2321)	< 0.001	13.58
GPTS-CP	0.0766 (±0.1737)	< 0.001	1.1911 (±0.1856)	< 0.001	20.91
GPAR	-0.4948 (±0.2976)	< 0.001	0.7757 (±0.1115)	0.054	412.66
GPAR-CP	-1.0430 (±0.1175)	0.013	0.7238 (±0.1275)	0.202	485.46
TIM	1.3853 (±0.1106)	< 0.001	1.3670 (±0.1943)	< 0.001	N/A
Whistler Snowfall Data (1000 training points, 13380 Test points)					
HSGPAR-CP	-0.0278 (±0.0531)	< 0.001	1.3040 (±0.0962)	< 0.001	605.64
HSSPAR-CP	**-0.52425** (±0.0393)	N/A	**0.9785** (±0.0900)	N/A	591.06
GPTS	1.2965 (±0.0495)	< 0.001	1.1828 (±0.0774)	0.002	18.15
GPTS-CP	0.6143 (±0.0693)	< 0.001	1.1701 (±0.0807)	0.003	59.10
GPAR	1.1708 (±0.1453)	< 0.001	1.1195 (±0.1013)	0.021	12,150.95
GPAR-CP	-0.1890 (±0.0433)	< 0.001	1.1959 (±0.0994)	0.004	14,493.47
TIM	0.3374 (±0.0264)	< 0.001	0.9912 (±0.0769)	0.381	N/A

UPM. We test the algorithms on four real data sets (3 in 1D and 1 in 2D). The average one-step-ahead negative log likelihood (NLL) and the mean squared error relative to the predictive mean (MSE) are used as evaluation metrics. Results are presented in Table 1.

5.1. Settings

We use a hazard function with a trainable constant hazard rate h initialized at 100, which yields a conditional prior $p(r_t|r_{t-1})$ with probability of a change point equal to 0.01. Following Saatçi et al. (2010), we used a rational quadratic kernel for GPTS and a Gaussian kernel for the auto-regressive variants. The GPTS execution time is improved by assuming uniform discrete observation time and exploiting the *Toeplitz* structure of the covariance function (Saatçi et al., 2010). For the implementation of GPAR and GAPRCP algorithms, we adopt the approach outlined by Saatçi et al. (2010), incorporating horizontal and vertical rank-one Cholesky updates and pruning techniques. The computational complexity of GAPRCP, denoted using our notation, is $O(TR_{\text{max}}^3)$. In addition, we also incorporate the original time-independent CP model (TIM) introduced by Adams & MacKay (2007) as a baseline, which assumes the data to be i.i.d. normal.

For HSSPAR, the trainable hyperparameters consist of the UPM parameters $\Theta := (\theta, \nu, \sigma_n)$ where θ refers to the kernel hyperparameters. Our implementations of HSSPAR and HSGPAR use the Hilbert space reduced-rank kernel derived from Gaussian kernels with the number of basis functions m ranging from 5 to 15. For auto-regressive UPM (GPAR and HSSPAR variants), we use lag parameter $p = 1, 2, 3$. We observed that for larger p, the computational advantage of HSSPAR reduces, since as discussed earlier, the number of multivariate basis functions increases exponentially with dimension. Other authors make similar observations for multivariate HSGP regression (Riutort-Mayol et al., 2022).

[1] Here error bars are $\pm 1.96\times$ standard error.

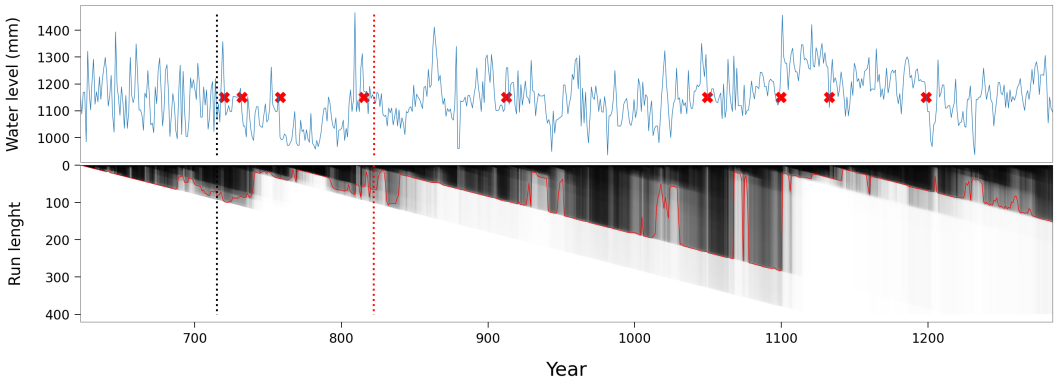

Figure 1: Results for the Nile record data with HSSPAR-CP. **Top:** The vertical dashed red line represents the boundary between train and test sets. The vertical dashed black line marks the installation of the nilometer in 715. The small red crosses represents alert locations obtained from MAP segmentation. **Bottom:** The run length CDF (black) and its median (red).

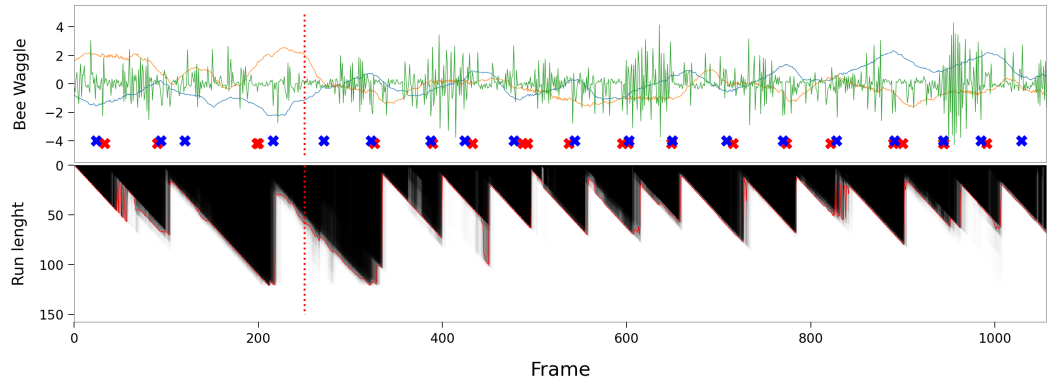

Figure 2: Results for the Bee Waggle Dance data with HSSPAR-CP. **Top:** The time-series are the bee's x-location (blue), y-location (orange) and angular difference (green). The vertical dashed red line represents the boundary between train and test sets. The small red crosses represents alert locations obtained from MAP segmentation. The small blue crosses represents the known true change point. **Bottom:** The run length CDF (black) and its median (red).

5.2. Nile Data

The *Nile* data set records the lowest annual water levels of the Nile river during the period 622-1284. The data has been used for change point detection in Garnett et al. (2009) and Saatçi et al. (2010). Following Saatçi et al. (2010), we learn the hyperparameters on the first 200 entries and evaluate the performance on the remaining period 822-1284. A structural change in the data is known to occur in year 715 due to an upgrade in ancient sensor technology to the nilometer. Results are given in Table 1. The run length posterior for HSSPAR is displayed in Figure 2. We can see by comparing HSGPAR-CP to GPAR-CP that the reduced-rank approximation does not alter the performance significantly. HSSPAR-CP outperforms both GPTS-CP and GSPAR-CP in terms of NLL. The error bars tend to be larger than desired, but this is something that was also observed in Saatçi et al. (2010), and attributable to the small test size (463 points). In Figure 2 we can also see that HSSPAR correctly captures the known change point at the year 715. While Saatçi et al. (2010) identified 18 CPs, our algorithm is more robust in that it only detects 9 CPs.

5.3. Well Log Data

The *Well Log* data set contains 4050 measurements of radioactivity taken during the drilling of a well. These data have been studied in the context of change point detection by Ruanaidh & Fitzgerald (2012) and by Fearnhead & Clifford (2003). The data set contains many outliers. Some authors, e.g. (Adams & MacKay, 2007; Levy-Leduc & Harchaoui, 2007) remove these before running the change point algorithms; however, outliers are retained by other authors, e.g. Fearnhead & Rigaill (2019) and Knoblauch et al. (2018). In our case, we use the data unfiltered. Results in Table 1 show slightly better performance for HSSPAR-CP compared to HSGPAR-CP and GPAR-CP even though this advantage might lack statistical significance. For this

data set, we see the effectiveness of the reduced-rank formulation when the training set becomes relatively large (≥ 1000). The fitting of HSSPAR-CP and HSGPAR-CP is $> 20\times$ faster than that of GPAR-CP in our experiment. In terms of alerted change points, on the unfiltered data, HSSPAR-CP identifies 25 CPs compared to 44 for GPAR-CP in Saatçi et al. (2010). Notably, when the data is filtered, the number of CPs reduces to 22 for HSSPAR-CP, indicating that applying filtering results in only 3 additional CPs. In Appendix C, we provide visualizations of the run length posterior for HSSPAR-CP on both filtered and unfiltered data.

5.4. Bee Waggle Dance Data

The waggle dance is bees' method of communicating the location of forage (direction, distance and profitability of food source) to each other. Entomologists have been interested in identifying change points in different stages in the bee dance. The *Bee Waggle Dance* data set contains the bee's x-coordinate position, y-coordinate position and head angle at each frame of 6 video sequences of bee waggle dances. Following Saatçi et al. (2010), we examine the first video sequence only, and consider angle differences for the angle sequence. HSSPAR-CP outperforms in terms of NLL and MSE. Figure 1 shows the run length posterior and change point alerts for HSSPAR-CP. The HSSPAR-CP model correctly identifies 16 of the 19 known CPs.

5.5. Snowfall Data

The *Snowfall* data report the historical daily snowfall level in Whistler BC (Canada) from 1972 to 2008. We train the model on the first 1000 entries of the data (corresponding to approximately three years) and test on the 12,880 remaining points. The HSSPAR-CP model performs significantly better in terms of both NLL and MSE compared to its competitors. Fitting of HSSPAR-CP is also $> 20\times$ faster than that of GPAR-CP.

The Student-t UPM outperforms other GP-based CP algorithms in terms of NLL in all experiments. We attribute this performance to the generalization property (compared to a GP) and to the fatter predictive distribution of a TP. The reduced-rank approximation yields significantly faster training while maintaining good performance for applications with larger training sets, i.e. *Well Log* and *Snowfall*.

6. Conclusion

We introduce a Bayesian online change point detection framework that combines a Student-t process with dependent Student-t noise as a time-series model, and *Hilbert space* reduced-rank kernel approximation for mitigating computation complexity. We illustrate the use of our scheme on a diverse set of real world examples. Our method compares favorably to other GP-based alternatives in terms of both prediction and hyperparameter learning time.

Acknowledgements

We also thank the anonymous reviewers for useful comments during the review process. Jeremy Sellier was supported under an EPSRC CASE studentship award in conjunction with Shell Research Ltd.

References

Adams, R. P. and MacKay, D. J. Bayesian online changepoint detection. *arXiv:0710.3742*, 2007.

Agudelo-España, D., Gómez-González, S., Bauer, S., Schölkopf, B., and Peters, J. Bayesian online detection and prediction of change points. *CoRR*, abs/1902.04524, 2019.

Archambeau, C. and Bach, F. Multiple Gaussian process models. *arxiv:1110.5238*, 2011.

Aroian, L. A. and Levene, H. The effectiveness of quality control charts. *Journal of the American Statistical Association*, 45(252):520–529, 1950.

Barry, D. and Hartigan, J. A. A Bayesian analysis for change point problems. *Journal of the American Statistical Association*, 88(421):309–319, 1993.

Bochner, S. Vorlesungen über fouriersche integrale. *Akademische Verlagsgesellschaf*, 1932.

Caron, F., Doucet, A., and Gottardo, R. On-line changepoint detection and parameter estimation with application to genomic data. *Statistics and Computing*, 22(2): 579–595, 2012.

Chib, S. Estimation and comparison of multiple changepoint models. *Journal of Econometrics*, 86(2):221–241, 1998.

Dawid, A. P. Some matrix-variate distribution theory: Notational considerations and a Bayesian application. *Biometrika*, 68:265–274, 1981.

Fearnhead, P. and Clifford, P. On-line inference for hidden Markov models via particle filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(4):887–899, 2003.

Fearnhead, P. and Liu, Z. On-line inference for multiple change-point problems. *Journal of the Royal Statistical Society Series B*, 69:589–605, 2007.

Fearnhead, P. and Rigaill, G. Changepoint detection in the presence of outliers. *Journal of the American Statistical Association*, 114(525):169–183, 2019.

Garnett, R., Osborne, M. A., and Roberts, S. J. Sequential Bayesian prediction in the presence of change-points. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML'09, pp. 345–352, 2009.

Green, P. J. Reversible-jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

Knoblauch, J. and Damoulas, T. Spatio-temporal Bayesian on-line change-point detection with model selection. In *Proceedings of the 35th International Conference on Machine Learning*, ICML'18, pp. 2718–2727, 2018.

Knoblauch, J., Jewson, J. E., and Damoulas, T. Doubly robust Bayesian inference for non-stationary streaming data with beta-divergences. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Koop, G. and Potter, S. Forecasting and estimating multiple change-point models with an unknown number of change points. *Rev. Econ. Stud.*, 74, 12 2004.

Kotz, S. and Nadarajah, S. *Multivariate t Distributions and Their Applications*. Cambridge University Press, 2004.

Kummerfeld, E. and Danks, D. Tracking time-varying graphical structure. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Levy-Leduc, C. and Harchaoui, Z. Catching change points with Lasso. In *Advances in Neural Information Processing Systems*, volume 20, 2007.

Li, X. and Ma, J. Non-central Student-t mixture of Student-t processes for robust regression and prediction. In *Intelligent Computing Theories and Application*, pp. 499–511, 2021.

Manogaran, G. and Lopez, D. Spatial cumulative sum algorithm with big data analytics for climate change detection. *Computers Electrical Engineering*, 65:207–221, 2018.

Murray-Smith, R. and Girard, A. Gaussian process priors with arma noise models. *Irish Signals and Systems Conference*, pp. 147–152, 2001.

Panda, S. and Nayak, A. Automatic speech segmentation in syllable-centric speech recognition system. *Int J Speech Technol 19*, 9(18), 12 2016.

Polunchenko, A. S., Tartakovsky, A. G., and Mukhopadhyay, N. Nearly-optimal change-point detection with an application to cybersecurity. *Sequential Analysis*, 31:409 – 435, 2012.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pp. 1177–1184, 2007.

Ranganathan, A., Yang, M.-H., and Ho, J. Online sparse Gaussian process regression and its applications. *IEEE Transactions on Image Processing*, 20(2):391–404, 2011.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. MIT Press, 2005.

Riutort-Mayol, G., Bürkner, P.-C., Andersen, M., Solin, A., and Vehtari, A. Practical Hilbert space approximate Bayesian Gaussian processes for probabilistic programming. *Statistics and Computing*, 33, 2022.

Ruanaidh, J. and Fitzgerald, W. *Numerical Bayesian Methods Applied to Signal Processing*. Statistics and Computing. Springer, 2012.

Rudin, W. Courier Dover Publications, 2017.

Saatçi, Y., Turner, R., and Rasmussen, C. E. Gaussian process change point models. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pp. 927–934, 2010.

Shah, A., Wilson, A., and Ghahramani, Z. Student-t processes as alternatives to Gaussian processes. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, ICML'14, pp. 877–885, 2014.

Solin, A. and Särkkä, S. Hilbert space methods for reduced-rank Gaussian process regression. *Statistics and Computing*, 30(2):419–446, mar 2020.

Solin, A. and Särkkä, S. State-space methods for efficient inference in Student-t process regression. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, ICML'15, pp. 885–893, 2015.

Svensson, A. and Schön, T. B. A flexible state–space model for learning nonlinear dynamical systems. *Automatica*, 80:189–199, 2017.

Svensson, A., Solin, A., Särkkä, S., and Schön, T. B. Computationally-efficient Bayesian learning of Gaussian process state space models. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, AISTATS, pp. 213–221, 2016.

Tang, Q., Wang, Y., and Xia, S.-T. Student-t process regression with dependent Student-t noise. In *Proceedings of the 22th European Conference on Artificial Intelligence*, ECAI'16, pp. 82–89, 2016.

Tang, Q., Niu, L., Wang, Y., Dai, T., An, W., Cai, J., and Xia, S.-T. Student-t process regression with Student-t likelihood. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 2822–2828, 2017.

Tracey, B. D. and Wolpert, D. Upgrading from Gaussian processes to Student-t processes. *AIAA Non-Deterministic Approaches Conference*, 2018.

Turner, R., Saatçi, Y., and Rasmussen, C. E. Adaptive sequential Bayesian change-point detection. In *Advances in Neural Information Processing Systems*, Temporal segmentation Workshop on the Meaning, 2009.

Williams, C. and Seeger, M. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, pp. 682–688, 2001.

Wilson, R., Nassar, M., and Gold, J. Bayesian online learning of the hazard rate in change-points problems. *Neural computation*, 22:2452–76, 2010.

Xuan, X. and Murphy, K. Modeling changing dependency structure in multivariate time series. In *Proceedings of the 24th International Conference on Machine Learning*, ICML'07, pp. 1055–1062, 2007.

Yu, S., Tresp, V., and Yu, K. Robust multi-task learning with t-processes. In *Proceedings of the 24th International Conference on Machine Learning*, ICML'07, pp. 1103–1110, 2007.

Zhang, Y. and Yeung, D. Multi-task learning using generalized t process. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, AISTATS, pp. 964–971, 2010.

A. Student-t Process Construction Details

In this section, we present a summary of the Student-t process derivation introduced by Shah et al. (2014).

We can construct a Student-t process by placing an inverse Wishart process prior on the kernel of a Gaussian process. The Wishart distribution is a probability distribution over $\Pi(n)$, the set of real-valued, $n \times n$, symmetric, positive definite matrices.

Definition A.1. A random matrix $\Sigma \in \Pi(n)$ is *inverse Wishart* distributed with parameters $\nu \in \mathbb{R}^+$, $\mathbf{K} \in \Pi(n)$ and we write $\Sigma \sim \mathcal{IW}(\nu, \mathbf{K})$ if its density is given by

$$p(\Sigma|v, \mathbf{K}) \propto |\Sigma|^{-\frac{v+2n}{2}} \exp\left(-\frac{1}{2} tr(\mathbf{K}\boldsymbol{\Sigma}^{-1})\right). \tag{25}$$

Dawid (1981) shows that the inverse Wishart distribution is consistent under marginalization. Thus we can defined a Wishart process for some input space $\mathcal{X}$ and a positive definite function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

Definition A.2. The process σ is a inverse Wishart process (IWP) on $\mathcal{X}$ with parameter v and kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbf{R}$ if for any finite collection $x_1, \cdots, x_n \in \mathcal{X}$, $\sigma(x_1, \cdots, x_n) \sim \mathcal{IW}(\nu, \mathbf{K})$ where $\mathbf{K} \in \Pi(n)$ is the Gram matrix with i, j entries $k(x_i, x_j)$ for $i, j = 1, \ldots, n$. We write $\sigma \sim \mathcal{IWP}(\nu, k)$.

For some kernel function k parametrized by θ and a mean function $\mu : \mathcal{X} \to \mathbb{R}$, Shah et al. (2014) propose deriving the Student-t process f as a hierarchical model such that

$$\begin{aligned} \sigma &\sim \mathcal{IWP}(\nu, k_\theta) \\ f|\sigma &\sim \mathcal{GP}(\mu, (\nu-2)\sigma). \end{aligned} \tag{26}$$

For any collection $\mathbf{f} = (f(x_1), \cdots, f(x_n))^\top$ with $\boldsymbol{\mu} = (\mu(x_1), \cdots, \mu(x_n))^\top$ and $\boldsymbol{\Sigma} = \sigma(x_1, \cdots, x_n)$, we see that

$$\begin{aligned} p(\mathbf{f}|\boldsymbol{\mu}, \mathbf{K}, v) &= \int p(\mathbf{f}|\boldsymbol{\Sigma}) p(\boldsymbol{\Sigma}|v, K) d\boldsymbol{\Sigma} \\ &\propto \frac{\exp\left(-\frac{1}{2} tr\left(\boldsymbol{K} + \frac{(\mathbf{f}-\boldsymbol{\mu})(\mathbf{f}-\boldsymbol{\mu})^\top}{\nu-2}\right)\right)}{|\boldsymbol{\Sigma}|^{(\nu+2n+1)/2}} \\ &\propto \left(1 + \frac{1}{v-2}(\mathbf{y}-\boldsymbol{\mu})^\top \mathbf{K}^{-1}(\mathbf{y}-\boldsymbol{\mu})\right)^{-\frac{v+n}{2}} \end{aligned} \tag{27}$$

which is a multivariate Student-t distribution $St(\boldsymbol{\mu}, \mathbf{K}, \nu)$. Since the multivariate Student-t distribution is consistent under marginalisation, Shah et al. (2014) conclude that Equation (27) is the finite-dimensional distribution of a well defined stochastic process f. We write $f \sim \mathcal{TP}(\nu, k)$.

B. Hilbert Space Methods for Reduced-Rank Kernels

In this section, we present a summary of the mathematical details of the Hilbert space based reduced-rank kernels introduced by Solin & Särkkä (2020).

Bochner representation Hilbert space methods for reduced-rank kernels are constructed via the Bochner's theorem (Bochner, 1932; Rudin, 2017), which states that any bounded, continuous and shift-invariant kernel $k(\mathbf{x}, \mathbf{x}') := k(\boldsymbol{\tau})$ with $\boldsymbol{\tau} = \mathbf{x} - \mathbf{x}'$, is the inverse Fourier transform of a bounded positive measure.

Theorem B.1. *(Bochner) An integrable function $k : \mathbb{R}^d \to \mathbb{R}$ is the covariance function of a weakly stationary mean square continuous random process on $\mathbb{R}^d$ if and only if it can be represented as*

$$k(\boldsymbol{\tau}) = \int_{\mathbb{R}^d} \exp(iw^\top \boldsymbol{\tau}) d\mu(w) \tag{28}$$

where $\mu(w)$ is a positive definite measure.

If the measure $\mu(w)$ admits a *spectral density* $S(w)$, we can furthermore express the following Fourier identities

$$\begin{aligned} k(\boldsymbol{\tau}) &= \frac{1}{(2\pi)^d} \int \exp(iw^\top \boldsymbol{\tau}) S(w) dw \\ S(w) &= \int k(\boldsymbol{\tau}) \exp(-iw^\top \boldsymbol{\tau}) d\boldsymbol{\tau}. \end{aligned} \tag{29}$$

In the isotropic case where the covariance function only depends on the Euclidian norm $||\boldsymbol{\tau}||$ such that $k(\boldsymbol{\tau}) = k(||\boldsymbol{\tau}||)$, the spectral density is also only dependent on the norm of w i.e. $S(w) = S(||w||)$.

Covariance operator as a pseudo-differential operator We can define a covariance operator $\mathcal{K}$ associated with each covariance function k as

$$\mathcal{K} f = \int k(\cdot, \mathbf{x}') f(\mathbf{x}') d\mathbf{x}' \tag{30}$$

for any regular functions f. When k is stationary then $\mathcal{K}$ is translation invariant. Thus, we can express the Fourier transform of $\mathcal{K}$ as a transfer function, which is the spectral density $S(\cdot)$ itself. Indeed, one can verify that $\mathcal{F}[\mathcal{K} f](w) = S(w)\mathcal{F}[f](w)$ where $\mathcal{F}[\cdot]$ denotes the Fourier transform of its argument.

We consider the isotropic case $S(w) = S(||w||)$. We further assume that $S(\cdot)$ is regular enough to be represented as a polynomial expansion i.e.

$$S(||w||) = a_0 + a_1 ||w||^2 + a_2 (||w||^2)^2 + a_3 (||w||^2)^3 + \cdots \tag{31}$$

Recall that the transfer function of the Laplace operator ∇^2 is $-||w||^2$ i.e. $\mathcal{F}[\nabla^2 f](w) = -||w||^2 \mathcal{F}[f](w)$. Thus from Equation (31), we have

$$\begin{aligned} \mathcal{F}[\mathcal{K} f](w) &= S(||w||)\mathcal{F}[f](w) \\ &= \left[a_0 + a_1 ||w||^2 + a_2 (||w||^2)^2 + a_3 (||w||^2)^3 + \cdots\right] \mathcal{F}[f](w) \\ &= a_0 \mathcal{F}[f](w) - a_1 \mathcal{F}[\nabla^2 f](w) - a_2 \mathcal{F}[(\nabla^2)^2 f](w) - a_3 \mathcal{F}[(\nabla^2)^3 f](w) + \cdots \\ &= \mathcal{F}[a_0 - a_1 \nabla^2 f - a_2 (\nabla^2)^2 f - a_3 (\nabla^2)^3 f + \cdots](w). \end{aligned} \tag{32}$$

From the equality (32), we get the following representation of $\mathcal{K}$, which defines a pseudo-differential operator as a series of Laplace operator

$$\mathcal{K} = a_0 - a_1 \nabla^2 - a_2 (\nabla^2)^2 - a_3 (\nabla^2)^3 + \cdots . \tag{33}$$

Hilbert space approximation of $\mathcal{K}$ We now form a Hilbert space approximation for the pseudo-differential operator defined in Equation (33). Consider the eigenvalue problem for the Laplace operator ∇^2 in the compact subset $\Omega \subset \mathbb{R}^d$ and with Dirichlet boundary conditions

$$\begin{aligned} -\nabla^2 \phi_j(\mathbf{x}) &= \lambda_j \phi_j(\mathbf{x}) \quad \text{if} \quad \mathbf{x} \in \Omega, \\ \phi_j(\mathbf{x}) &= 0 \qquad\qquad \text{if} \quad \mathbf{x} \in \partial\Omega \end{aligned} \tag{34}$$

where $\{\phi_j\}_{j=1}^{\infty}$ and $\{\lambda_j\}_{j=1}^{\infty}$ are the set of eigenvalues and eigenfunctions of the Laplacian operator. Because $-\nabla^2$ is a positive definite Hermitian operator, the set of eigenfunction $\{\phi_j\}_{j=1}^{\infty}$ is orthonormal with respect to the inner product

$$\langle f, g\rangle = \int_{\Omega} f(\mathbf{x})g(\mathbf{x})d\mathbf{x} \tag{35}$$

that is

$$\int_{\Omega} \phi_i(\mathbf{x})\phi_j(\mathbf{x})d\mathbf{x} = \delta_{i,j} \tag{36}$$

and all eigenvalues $\{\lambda_j\}$ are real and positive.

The Laplace operator can be assigned a formal kernel

$$l(\mathbf{x}, \mathbf{x}') = \sum_j \lambda_j \phi_j(\mathbf{x})\phi_j(\mathbf{x}') \tag{37}$$

in a sense that

$$\begin{aligned} -\nabla^2 f(\mathbf{x}) &= \sum_j \lambda_j \langle f, \phi_j\rangle \phi_j(\mathbf{x}) \qquad \textit{(spectral decomposition)} \\ &= \int_{\Omega} l(\mathbf{x}, \mathbf{x}')f(\mathbf{x}')d\mathbf{x}'. \end{aligned}$$

Similarly, we can define the kernel of the power representation the Laplace operator as

$$l^s(\mathbf{x}, \mathbf{x}') = \sum_j \lambda_j^s \phi_j(\mathbf{x})\phi_j(\mathbf{x}') \tag{38}$$

for $s = 1, 2, \cdots$, in a sense that due the orthonormality of the basis

$$-(\nabla^2)^s f(\mathbf{x}) = \int_{\Omega} l^s(\mathbf{x}, \mathbf{x}')f(\mathbf{x}')d\mathbf{x}'.$$

This implies that we also have

$$\begin{aligned} &\left[a_0 + a_1(-\nabla^2) + a_2(-\nabla^2)^2 + \cdots\right] f(\mathbf{x}) \\ &= \int_{\Omega} \left[a_0 + l^1\nabla^2 + l^2(\nabla^2)^2 + \cdots\right] f(\mathbf{x}')d\mathbf{x}'. \end{aligned} \tag{39}$$

The left hand side is $\mathcal{K}f$ as defined in Equation (33). Thus from Equation (30), we conclude that

$$\begin{aligned} k(\mathbf{x}, \mathbf{x}') &\approx a_0 + a_1 l^1(\mathbf{x}, \mathbf{x}') + a_2 l^2(\mathbf{x}, \mathbf{x}') + \cdots \\ &= \sum_j \left[a_0 + a_1\lambda_j + a_2\lambda_j^2 + \cdots\right] \phi_j(\mathbf{x})\phi_j(\mathbf{x}'). \end{aligned} \tag{40}$$

By letting $||w||^2 = \lambda_j$ the spectral density in Equation (31) becomes

$$S(||w||) = a_0 + a_1\lambda_j + a_2\lambda_j^2 + a_3\lambda_j^3 + \cdots .$$

and substituting in Equation (40) leads to the final approximation

$$k(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{\infty} S(\sqrt{\lambda_j})\phi_j(\mathbf{x})\phi_j(\mathbf{x}') \tag{41}$$

As discussed in the main text, the eigenvalues λ_j are monotonically increasing with j and for bounded kernel the spectral density goes to zero with higher frequencies. If we truncate the sum to the first m terms, the approximate covariance becomes

$$k(\mathbf{x}, \mathbf{x}') \approx \sum_{j=1}^{m} S(\sqrt{\lambda_j})\phi_j(\mathbf{x})\phi_j(\mathbf{x}'). \tag{42}$$

In one dimension For one dimension within a closed interval $\Omega = [-L, L] \subset \mathbb{R}$ where L is some positive real number, the solution to the Laplacian eigenvalue problem in Equation (34) is independent of the specific choice of covariance function and is given by

$$\begin{aligned} \phi_j(x) &= \frac{1}{\sqrt{L}} \sin\left(\frac{\pi j(x+L)}{2L}\right), \\ \lambda_j &= \left(\frac{\pi j}{2L}\right)^2. \end{aligned} \tag{43}$$

for $j = 1, \cdots, m$ where m denotes the number of basis functions.

In d dimensions In the d-dimensional case, we consider rectangular domain $\Omega = [-L_1, L_1] \times \cdots \times [-L_d, L_d]$ with Dirichlet boundary conditions. The number of eigenfunctions and eigenvalues in the approximation is equal to the number of d-tuples, that is, possible combinations of univariate eigenfunctions over all dimensions.

Every k-th dimension has a number of univariate basis functions equal to m_k with indices ranging from $1, \cdots, m_k$. Let $\mathbf{S} \in \mathbb{N}^{m^* \times d}$ be the matrix of all these d-tuples indices with $m^* = \prod_{k=1}^{d} m_k$.

Each multivariate eigenfunction $\phi_j^* : \Omega \to \mathbb{R}$ corresponds to the product of the univariate eigenfunctions whose indices corresponds to the j-th element of the d-tuples $\mathbf{S}_{j,\cdot}$, and each multivariate eigenvalue $\boldsymbol{\lambda}_j^*$ is a d-vector with elements that are the univariate eigenvalues whose indices corresponds to the j-th elements of the d-tuples $\mathbf{S}_{j,\cdot}$. Thus for $\mathbf{x} = (x_1, \cdots, x_d) \in \Omega$ and $j = 1, \cdots, m^*$, we have

$$\begin{aligned} \phi_j^*(\mathbf{x}) &= \prod_{k=1}^{d} \phi_{\mathbf{S}_{j,k}}(x_k) = \prod_{k=1}^{d} \frac{1}{\sqrt{L_k}} \sin\left(\frac{\pi j(\mathbf{S}_{j,k} + L_k)}{2L_k}\right), \\ \boldsymbol{\lambda}_j^* &= \{\lambda_{\mathbf{S}_{j,k}}\}_{k=1}^{d} = \left\{\left(\frac{\pi \mathbf{S}_{j,d}}{2L}\right)^2\right\}_{k=1}^{d} \end{aligned} \tag{44}$$

for $j = 1, \cdots, m^*$.

The approximate covariance function is then

$$k(\mathbf{x}, \mathbf{x}') \approx \sum_{j=1}^{m} S\left(\sqrt{\boldsymbol{\lambda}_j^*}\right) \phi_j^*(\mathbf{x}) \phi_j^*(\mathbf{x}') \tag{45}$$

where S denotes the d-dimensional spectral density of the covariance functions with argument $\sqrt{\boldsymbol{\lambda}_j^*}$ that denotes the element-wise square root of the vector $\boldsymbol{\lambda}_j^*$.

C. Supplementary Figure for the Well Log Data Set

The figures below depict the run length posterior and change point alerts for HSSPAR-CP on the Well Log data set. Figure 3 presents results obtained using unfiltered data, similar to the experiment described in Table 1 of the main text. To explore the effects of outlier processing, Figure 4 showcases results obtained from a filtered version of the Well Log data set.

Furthermore, Figure 5 and Figure 6 display similar outcomes as Figure 3, but with a reduced measurement range of 400 to 1200 and 1600 to 2700 (time units), respectively. Consequently, Figure 3 allows for a direct comparison to Figure 2 in Adams & MacKay (2007).

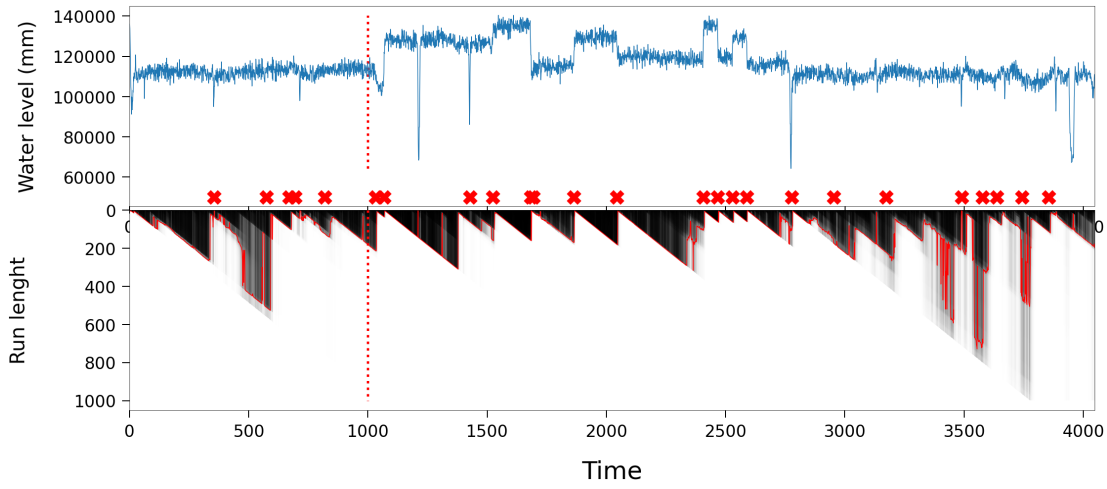

Figure 3: Results for the unfiltered Well Log data with HSSPAR-CP. **Top:** The vertical dashed red line represents the boundary between train and test sets. The small red crosses represents alert locations obtained from MAP segmentation. **Bottom:** The run length CDF (black) and its median (red).

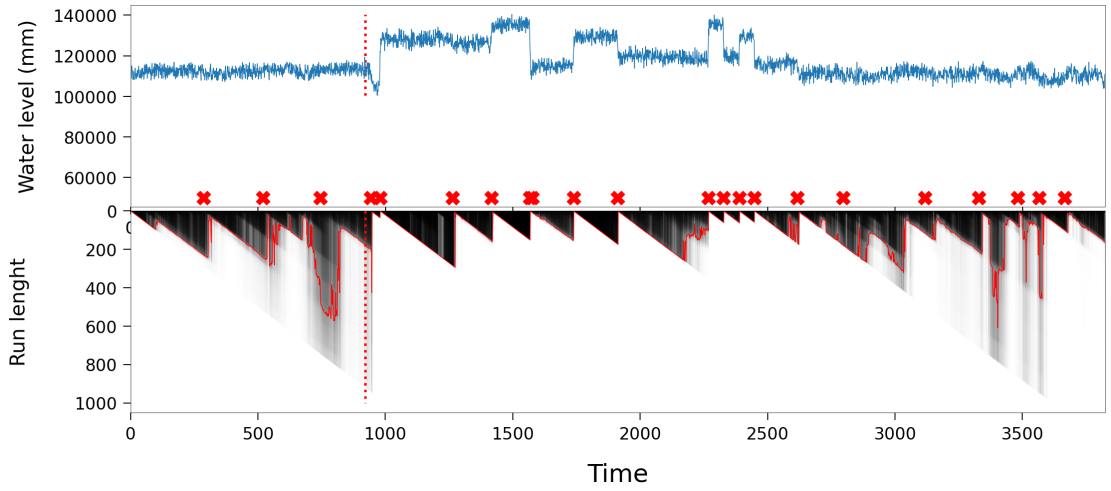

Figure 4: Results for the filtered Well Log data with HSSPAR-CP. **Top:** The vertical dashed red line represents the boundary between train and test sets. The small red crosses represents alert locations obtained from MAP segmentation. **Bottom:** The run length CDF (black) and its median (red).

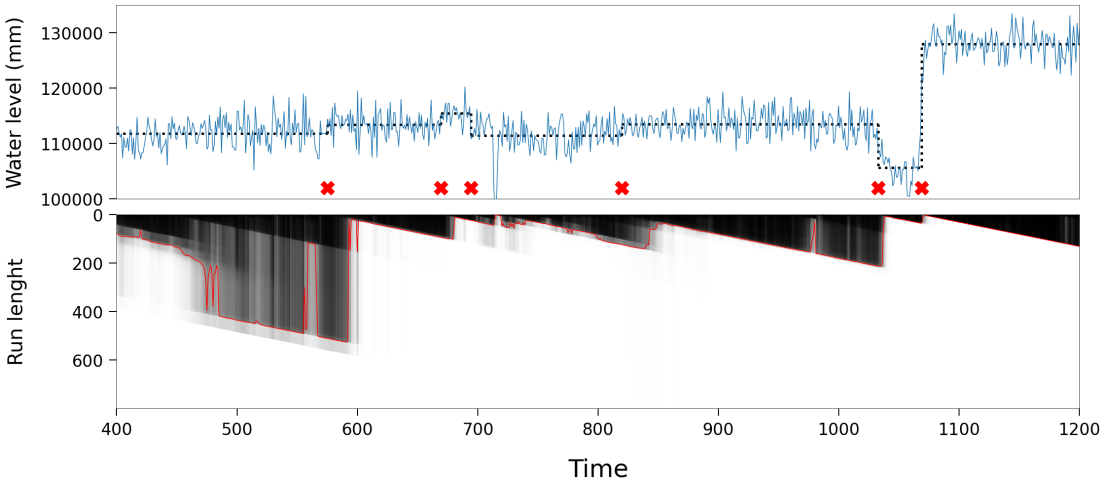

Figure 5: Results for the unfiltered Well Log data with HSSPAR-CP, considering measurements ranging from 400 to 1200 (in time units). **Top:** Alert locations obtained from MAP segmentation are represented by small red crosses. The horizontal dashed black line indicates the mean of observations between change points. **Bottom:** The run length CDF (black) and its median (red).

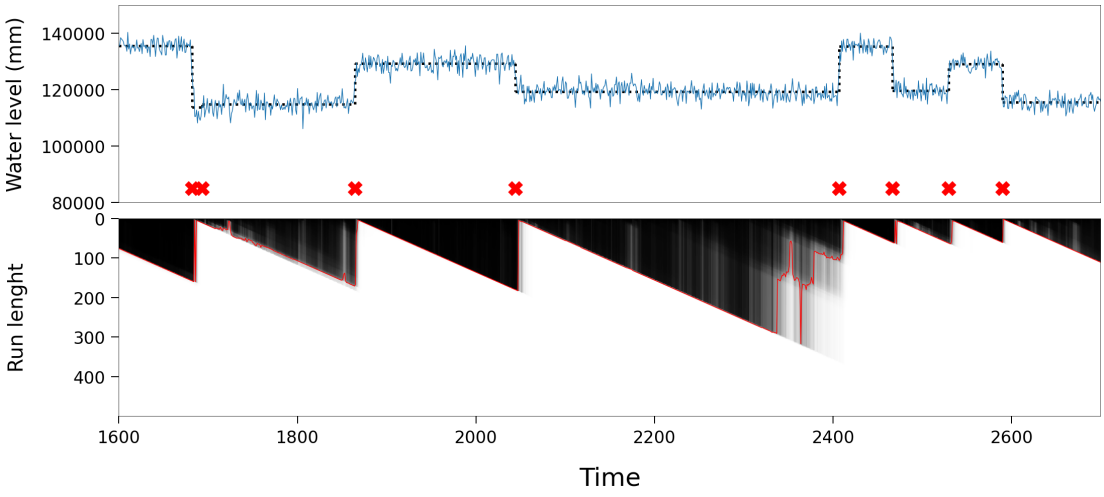

Figure 6: Results for the unfiltered Well Log data with HSSPAR-CP, considering measurements ranging from 1600 to 2700 (in time units). **Top:** Alert locations obtained from MAP segmentation are represented by small red crosses. The horizontal dashed black line indicates the mean of observations between change points. **Bottom:** The run length CDF (black) and its median (red).