Offline Changepoint Detection With Gaussian Processes

Janneke Verbeek[1] **Tom Heskes**[1] **Yuliya Shapovalova**[1]

[1]Radboud University Nijmegen, Institute for Computing and Information Sciences, Nijmegen, The Netherlands

Abstract

This work proposes Segmenting changepoint Gaussian process regression (SegCPGP), an offline changepoint detection method that integrates Gaussian process regression with the changepoint kernel, the likelihood ratio test and binary search. We use the spectral mixture kernel to detect various types of changes without prior knowledge of their type. SegCPGP outperforms state-of-the-art methods when detecting various change types in synthetic datasets; in real world changepoint detection datasets, it performs on par with its competitors. While its hypothesis test shows slight miscalibration, we find SegCPGP remains reasonably reliable.

1 INTRODUCTION

Changepoint detection (CPD) refers to the problem of finding and characterizing changes in data generating processes, such as changes in the mean, variance, trend, periodicity, or other properties of the data. Applications of change point detection algorithms include climate data [Reeves et al., 2007], quality control, [Lai, 1995] EEG analysis, network analysis [Tartakovsky et al., 2012] and finance [Andreou and Ghysels, 2009].

Changepoint detection is an extensively studied problem [Truong et al., 2020, Aminikhanghahi and Cook, 2017, Reeves et al., 2007, Van den Burg and Williams, 2020, Aue and Horváth, 2013]; available methods can be divided in *online* methods, which detect changepoints as new data arrives, and *offline* methods, which analyze the entire dataset at once to identify changepoints.

In online changepoint detection, CPD algorithms need to be efficient enough to process a potentially never-ending stream of data. Processing only one window of data at a time is a common strategy for these algorithms [Keogh et al., 2001, Chen et al., 2022]. For example, a popular Bayesian online changepoint detection (BOCPD) method estimates the run length, which represents the number of time steps since the last changepoint and essentially dynamically detects shifts in the data as new observations arrive [Adams and MacKay, 2007]. Several variations on BOCPD, for instance robust versions [Altamirano et al., 2023, Knoblauch et al., 2018], and model selection Knoblauch and Damoulas [2018]) have been proposed as extensions.

Numerous other offline changepoint detection methods exist [Killick et al., 2012, Auger and Lawrence, 1989, Haynes et al., 2017, Zou et al., 2014, Celisse et al., 2018] — for a comprehensive overview, see [Truong et al., 2020]. An offline method of particular interest to this paper is binary segmentation [Scott and Knott, 1974, Vostrikova, 1981], which recursively partitions the signal by selecting split points that optimize a specific metric, such as likelihood or information criterion. Some variations of this algorithm exist [Fryzlewicz, 2014, Olshen et al., 2004].

Many CPD methods are designed for specific changes (e.g., detecting mean or variance shifts in time series). Gaussian processes (GPs) provide a flexible framework where different types of changes may be incorporated at the same time. CPD methods based on Gaussian processes (GPs) have been widely studied in online setting [Caldarelli et al., 2022, Garnett et al., 2009, Saatçi et al., 2010], but their application in the offline setting remains underexplored. In offline methods where GPs are used, the focus has primarily been on detecting mean shifts [Keshavarz et al., 2018, Lebarbier, 2005].

A Gaussian process is fully determined by its mean and covariance function, also known as the *kernel*, making their selection a crucial step in its application. The choice of kernel reflects prior beliefs about the types of functions the GP should model. In the context of CPD, this is especially important when little is known about the data or the nature of the changes to be detected. Thus, the selection of a suitable

kernel may prove crucial to the overall performance of a GP-based CPD method.

Consequently, our research aims to answer the question: can we devise an offline, Gaussian process based changepoint detection method without the need to devote much attention to kernel selection? In the next section, we will proceed with a more detailed discussion of available Gaussian process-based changepoint detection methods.

2 RELATED WORK

Gaussian processes (GPs) are flexible, nonparametric models that are capable of modeling spatiotemporal correlations. GPs have found ample application in changepoint detection, particularly in the online setting. GPTS-CP [Saatçi et al., 2010] models temporal correlations in the BOCPD framework, using GPs as an underlying predictive model. However, the BOCPD framework can be highly sensitive to the choice of hyperparameters which can hinder its performance in real-world setting.

As an alternative to BOCPD, Adaptive Gaussian process change point detection (ADAGA) [Caldarelli et al., 2022] is an online changepoint detection method based on statistical hypothesis testing. ADAGA detects changepoints via a window sliding method and tests whether the function values in the subwindow come from the same observational model as the rest of the window. The authors derive theoretical bounds for the probability of Type I and Type II errors in their changepoint detection heuristic. Nevertheless, ADAGA still relies on a prior specification of the kernel for different types of changes.

Garnett et al. [2009] exploited the kernel structure of GPs for CPD in the online setting, inspired by work on general linear models [Ruanaidh et al., 1994]. By using block-diagonal covariance matrices, their approach captures abrupt transitions between regimes governed by different kernels. In this case, the location of the changepoint can then be treated as a kernel parameter. In contrast, the changepoint kernel [Lloyd et al., 2014] parametrizes changepoints via steepness as well as location. This kernel has been proposed in an automatic statistician-type of framework for modeling complex time series behavior, but, to our knowledge, has not been explored in the context of CPD.

The likelihood ratio test has been used in the context of CPD more frequently, for instance in Caldarelli et al. [2022]. For a general overview, see Aminikhanghahi and Cook [2017], Truong et al. [2020].

Contributions We propose SegCPGP, a flexible offline changepoint detection method based on Gaussian processes that makes no assumptions about the type or nature of changes that might occur in the data. SegCPGP builds upon several components. First, we utilize the changepoint kernel, allowing for both steep and smooth transitions. Second, we use the likelihood ratio test with binary segmentation [Scott and Knott, 1974] for sequential detection of multiple changepoints in the data. Finally, we propose incorporating the spectral mixture kernel [Wilson and Adams, 2013] within the changepoint kernel framework, allowing for flexibility beyond mean/variance changes and eliminating the need to specify the nature of changes a priori. Code for SegCPGP is publicly available[1].

3 BACKGROUND

This section is structured as follows. We begin with an overview of Gaussian processes and Gaussian process regression. Next, we introduce two specific kernels that form the basis of our approach: the spectral mixture kernel, which aims to alleviate the challenge of kernel selection, and the changepoint kernel.

3.1 GAUSSIAN PROCESSES

Gaussian Process (GP) A Gaussian process is a collection of random variables, any finite subset of which has a multivariate Gaussian distribution (see for an extensive introduction Williams and Rasmussen [2006]). It is fully defined by its mean function $\mu(t)$ and covariance function $k(t, t')$. For a finite set of input points $t = \{t_1, t_2, \ldots, t_n\}$, a Gaussian process is denoted as

$$f(t) \sim \text{GP}(\mu(t), k(t, t')).$$

In this paper, without loss of generality, we assume that $\mu(t) = 0$, but the proposed approach can be straightforwardly extended to specific mean functions.

Gaussian Process Regression (GPR): Gaussian process regression is a non-parametric Bayesian approach that assigns a Gaussian process prior on the functional relationship between input and output variables. A Gaussian process regression is defined as

$$y(t) = f(t) + \epsilon(t), \tag{1}$$

where $\epsilon(t) \sim N(0, \sigma_\epsilon^2)$ is Gaussian noise. Given a set of observed input-output pairs $D = \{(t_1, y_1), (t_2, y_2), \ldots, (t_n, y_n)\}$, the goal is to estimate the function $f(t)$ and make predictions for new, unseen inputs t^*. In the case of the Gaussian likelihood, the posterior distribution over $f(t^*)$ is available in the closed form [Williams and Rasmussen, 2006]

$$p(f(t^*)|D) = \mathcal{N}(f(t^*)|\mu^*, \Sigma^*), \tag{2}$$

[1] https://github.com/JVerbeek/segcpgp/

where μ^* is the predictive mean and Σ^* is the predictive covariance. The mean and covariance of the posterior distribution are given by

$$\mu^* = k(t^*, t)[K + \sigma_\epsilon^2 I]^{-1} y, \tag{3}$$

$$\Sigma^* = k(t^*, t^*) - k(t^*, t)[K + \sigma_\epsilon^2 I]^{-1} k(t, t^*) \tag{4}$$

where $k(t^*, t)$ is the covariance matrix between the test inputs t^* and the training inputs t, $k(t^*, t^*)$ is the covariance matrix between the test inputs, $K = k(t, t)$ is the covariance matrix for the training inputs, y is the vector of observed outputs, and σ_ϵ^2 is the noise variance.

The kernel hyperparameters of the GP prior and the variance of the noise, denoted together by θ, are inferred by maximizing the marginal log-likelihood, given by

$$\log p(y|t, \theta) = -\frac{1}{2} y^T [K + \sigma_\epsilon^2 I]^{-1} y - \frac{1}{2} \log |K + \sigma_\epsilon^2 I| - \frac{N}{2} \log 2\pi. \tag{5}$$

Equations (1) through (5) represent the mathematical formulation of Gaussian Process Regression, allowing for the estimation of the posterior distribution over the function values and providing predictions with associated uncertainties.

3.2 SPECTRAL MIXTURE KERNEL

In Equations (3)-(4), the covariance function, also known as a kernel, plays a crucial role in modeling the similarity or correlation between different inputs. The structural form of the kernel directly determines which kinds of functions can be drawn from a Gaussian process prior [Williams and Rasmussen, 2006]. Notable examples of kernel functions include: the squared exponential kernel, for modeling smooth functions without discontinuities or abrupt changes; the Matérn family kernels, which allow for modeling some degree of roughness or discontinuities; and the periodic kernel, which allows for modeling repeating patterns in time series, such as seasonality. For an illustration of these kernel functions, see Appendix A.

If there is no prior knowledge about the most suitable kernel function for a given task, the appropriate structural form can be determined through kernel search [Duvenaud et al., 2013] and kernel learning [Bach, 2008]. Kernel search involves exploring the space of possible kernels, which can be computationally expensive. In contrast, kernel learning offers a more efficient alternative, potentially reducing the computational complexity of GPR from cubic to linear [Wilson et al., 2016]. Kernel learning in the context of changepoint detection, however, would likely require training data with labeled changepoints, which may not always be available in the context of CP detection.

Kernel selection may be sidestepped by using kernels that are sufficiently expressive, such as the spectral mixture (SM) kernel Wilson and Adams [2013]. The SM kernel can in theory approximate any stationary covariance kernel as a mixture of Gaussians in the frequency domain. We will further discuss this kernel in the remainder of the section and later apply it in the context of change point detection problem.

According to Bochner's theorem, any stationary covariance function $k(\cdot)$ can be expressed as an integral of the form

$$k(\tau) = \int_{\mathbb{R}^P} e^{2\pi i s^\top \tau} \psi(dt), \tag{6}$$

where we use $\tau = t - t'$ as a notational shorthand similarly to Wilson and Adams [2013] and ψ is a positive finite measure. If $\psi(ds)$ has a spectral density $S(s)$, then $k(\tau)$ and $S(s)$ are Fourier duals

$$k(\tau) = \int_{\mathbb{R}^P} S(s) e^{2\pi i s^\top \tau} ds,$$

$$S(s) = \int_{\mathbb{R}^P} k(\tau) e^{-2\pi i s^\top \tau} d\tau.$$

The spectral density $S(s)$ can be approximated via a Gaussian mixture model (GMM). A GMM models the data as a mixture of Q Gaussian densities with means $\mu_1, \ldots, \mu_Q$ and variances $\sigma_1^2, \ldots, \sigma_Q^2$ so that $k(\tau)$ has the form

$$k(\tau) = \sum_{q=1}^{Q} w_q \exp(-2\pi^2 \tau^2 \sigma_q^2) \cos(2\pi\tau\mu_q). \tag{7}$$

The weights w_q specify the relative contribution of each component, and do not necessarily sum to 1 as in a GMM. For a single Gaussian component, the mean μ can be interpreted as the frequency captured by the component. The inverse of the standard deviation σ represents the lengthscale, which determines how smooth or wiggly the function is. A large lengthscale for a spectral mixture component leads to functions that are almost constant, while a small lengthscale may result in a more periodic function. Note that since τ is the difference between t and t', the quantity $-2\pi^2\tau^2\sigma^2$ corresponds to a squared Euclidean norm scaled by lengthscale. Provided enough Gaussian mixture components are used, any stationary covariance function can be approximated in this way [Wilson and Adams, 2013].

To the best of our knowledge, the spectral mixture kernel has not been used in the context of changepoint detection. Due to its versatility, we apply the spectral mixture kernel in the context of multiple changepoint detection to detect different types of changepoints with a single kernel. Kernel selection is therefore largely bypassed. We use the SM kernel implementation of Leeftink and Hinne [2020]. To initialize the SM kernel hyperparameters, a Lomb-Scargle periodogram is used to approximate the empirical spectrum (as in Leeftink and Hinne [2020]); subsequently, a GMM is fit to this spectrum.

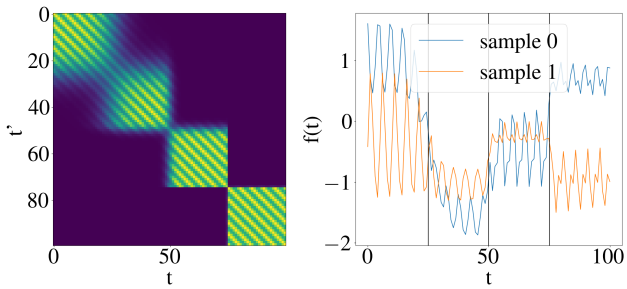


Figure 1: Left: kernel structure of the changepoint kernel with periodic base kernels for three changepoints at locations 25, 50 and 75. The steepness of the changes is 0.3, 1 and 10 at index 25, 50 and 75, respectively. Right: two samples from the kernel displayed on the left.

3.3 CHANGEPOINT KERNEL

The changepoint (CP) kernel was first proposed by Lloyd et al. [2014] in an automatic statistician-type framework. The kernel specifies a structural change in a signal, in particular (possibly) a smooth transition between two base kernels. In the following section, we will give a definition of the CP kernel.

Let $k_1(t,t')$ and $k_2(t,t')$ be base kernels (such as RBF/linear/local periodic/spectral mixture). Then the change point kernel is defined as

$$\text{cov}(f(t), f(t')) = k_1(t,t')\bar{\psi}(t,t') + k_2(t,t')\psi(t,t'), \quad (8)$$

where $\psi(t,t') = \psi(t)\psi(t')$ and $\bar{\psi}(t,t') = (1-\psi(t))(1-\psi(t'))$. The sigmoid $\psi(t)$ is parametrized by the location (t_0) and steepness (s) parameters, $\psi(t) = 1/(1+\exp(-s(t-t_0)))$. Besides inferring kernel parameters such as variance, period or lengthscale defined previously, we can also infer the location of the change point t_0 and steepness of the change s.

As an example, Figure 1 shows a changepoint kernel with changepoints at several locations, as well as a sample from that kernel. The steepness of each of these changepoints is different, leading to smoother or more abrupt transitions from regime to regime.

4 METHODOLOGY

In this section, we define an algorithm based on Gaussian process regression with change point kernel and binary search to detect multiple changepoints.

4.1 MODEL SELECTION

To determine whether a dataset contains a changepoint, we propose to compare two models with a likelihood-ratio test (LRT): a GPR with a single kernel and one with a changepoint kernel.

Let the Gaussian process regression with a single kernel be the *single GPR*

$$y(t) = f(t) + \epsilon(t),$$
$$f(t) \sim \text{GP}(0, k(t,t')).$$

Furthermore, let the Gaussian process regression that employs the changepoint kernel be the *changepoint GPR*

$$y(t) = f(t) + \epsilon(t),$$
$$f(t) \sim \text{GP}(0, k_1(t,t')\bar{\psi}(t,t') + k_2(t,t')\psi(t,t')),$$

where the Gaussian process is defined by the change point kernel as in (8). The (log-)likelihoods for both these models can be computed using (5).

The likelihood ratio test statistic $\mathcal{R}$ is given by

$$\mathcal{R} = -2(\log p(y|t,\theta_0) - \log p(y|t,\theta_1)), \quad (9)$$

where θ_0 are hyperparameters of the *single GPR* model and θ_1 are hyperparameters of the *changepoint GPR*. If the models are composite (or nested) — that is, the parameter space of the null model is in the interior of the parameter space of the alternative model — then in theory $\mathcal{R}$ follows a χ^2_d-distribution under the null hypothesis, where d is the difference in dimensionality between the two models [Wilks, 1938]. The p-value is then obtained as the density of the χ^2_d distribution larger than $\mathcal{R}$.

We are interested in applying the likelihood ratio test between the single and changepoint GPR; we thus need to reduce the alternative model to the null model. Placing the constraint $s = \infty$ on the steepness parameter of the changepoint kernel with base kernels $k_1(t,t')$ and $k_2(t,t')$, reduces the changepoint kernel to $k_1(t,t')$ or $k_2(t,t')$ respectively (for a detailed elaboration, see Appendix B). Thus, when we set the single GPR's kernel equal to k_1 or k_2, we arrive at the desired model selection.

Setting $s = \infty$ means that the null and alternative models are no longer composite. Since ∞ lies on the boundary of the admissible values for s, the null model does not lie in the interior of the alternative model's parameter space. Therefore, in practice, the distribution of $\mathcal{R}$ may (slightly) deviate from χ^2_d. We will further discuss this in the Experiments section.

4.2 SEGCPGP

The model selection described in the previous paragraph can be used to detect single changepoints. In real applications, it is often desirable to detect multiple changepoints. The changepoint kernel can be extended to support multiple changepoints (see (13)). In optimization, however, the

changepoint locations would then need to be constrained such that each changepoint location parameter estimates a unique location. Consequently, we combine the detection of single changepoints with a sequential search strategy to detect multiple changepoints.

To detect multiple changepoints, we propose segmenting changepoint Gaussian process regression (SegCPGP) to estimate multiple changepoints at unknown locations. SegCPGP combines binary search with a changepoint GPR that estimates a changepoint at a single location.

SegCPGP estimates changepoints sequentially. The procedure is first run on the whole time series to identify a potential changepoint. If a changepoint is found, the time series is divided at that point. The method then repeats this process on each resulting subwindow. Two GPRs — the changepoint GPR and the single GPR, as defined in the previous section — are fit on the full signal by optimizing the log marginal likelihood (LML). Any valid kernel function can be used as base kernels ($k_1(t, t')$ or $k_2(t, t')$) in the change point kernel, and may be selected to reflect prior beliefs about the change type. We evaluate standard kernel choices as well as the SM kernel that could be adopted in situations when there are no prior beliefs about the types of changes.

The likelihood in (5) is known to suffer from multiple local optima [Williams and Rasmussen, 2006]. Thus, we apply the standard GPR practice of restarting the optimization multiple times before selecting the highest likelihood model.

The single and changepoint GPR are compared via the likelihood ratio statistic described in the previous section. As the null distribution, we use the χ^2_d distribution, setting d equal to the difference in the dimensionality between the single and changepoint GPR. The value of d depends on the number of kernel hyperparameters in k_1 and/or k_2. For most of our experiments, we set the p-value of the LRT at $p = 0.1$ unless otherwise specified.

The changepoint detection procedure is sequential. If the LRT returns significant, the value of the changepoint kernel's location parameter is the estimated changepoint, which we denote by $\hat{t}$. Since $\hat{t}$ is regressed it is rounded to the nearest integer. Then, the signal is split into two halves. To avoid detecting the same changepoint multiple times, we remove a margin ϵ of the signal in the neighborhood of the detected changepoint. For a detected changepoint $\hat{t}$, the signal is therefore split at $\hat{t} + \epsilon$ and $\hat{t} - \epsilon$, where we set ϵ to 5 timesteps in practice. The changepoint search stops when $\hat{t}$ is outside the domain of the signal, when the LRT is not significant, or when only a single time step is left in the signal.

Pseudocode for the above procedure can be found in Algorithm 1.

5 EXPERIMENTS

Here, we demonstrate the performance of SegCPGP on synthetic and real-world datasets and compare it against several baseline algorithms. We provide an empirical analysis of SegCPGP's and ADAGA's Type I and Type II error rates.

Evaluation Results are reported in terms of the modified F_1-score, a commonly used metric in changepoint detection [Caldarelli et al., 2022, Killick et al., 2012, Van den Burg and Williams, 2020]. A detailed description of the F1-score is provided in Appendix H.1. An estimated changepoint is considered a true positive (TP) if it falls within a small margin around the true change point. A false positive (FP) is then any estimated changepoint outside of these margins, while a false negative (FN) is any missed changepoint within these margins and a true negative (TN) is the correctly identified absence of a changepoint. Setting the margin around the true changepoint to 0 skews the accuracy of classification metrics, since changepoints are only a small subset of the total number of datapoints. Thus, the margin is often set to 5 time steps in practice [Caldarelli et al., 2022, Killick et al., 2012, Van den Burg and Williams, 2020].

Differences in performance between methods in the benchmark dataset of Van den Burg and Williams [2020] are tested via a Wilcoxon signed-rank test. When ranking two algorithms, one with performance P, another with performance Q, the null hypothesis of the Wilcoxon signed-rank test is that the distribution F of the differences in performance $F(P - Q)$ is symmetric around 0, or equivalently, that $F(Q - P) = F(P - Q)$, meaning the algorithms are effectively interchangeable. The Wilcoxon signed-rank test is appropriate for evaluating the pairwise differences between algorithms in our experiments [Benavoli et al., 2016, Van den Burg and Williams, 2020]. We set the significance level of the test to 10% (i.e., p-value = 0.1). In order to correct for multiple testing, we apply a Holm correction [Demšar, 2006].

Baseline methods We use a subset of the methods available in the Turing changepoint detection benchmark of Van den Burg and Williams [2020] in our experimental evaluation. In particular, we include the following commonly used methods: BinSeg [Scott and Knott, 1974], PELT [Killick et al., 2012], BOCPD [Adams and MacKay, 2007] and RBOCPDMS [Knoblauch and Damoulas, 2018]. Additionally, we incorporate kernel-based and Gaussian process-based methods in our comparison, namely KCPA [Harchaoui et al., 2009] and ADAGA [Caldarelli et al., 2022], as well as nonparametric methods, namely CPNP [Haynes et al., 2017] and ECP [Matteson and James, 2014]. For these algorithms, their default initializations are used, which corresponds to applying the algorithms without prior knowledge of what reasonable hyperparameter settings might be. This experimental setting is also adopted, and was described

as being the most realistic, in Van den Burg and Williams [2020], Caldarelli et al. [2022]. A ZERO method is included in the evaluation, which corresponds to a method that by definition finds no changepoints.

We evaluate SegCPGP with four different base kernels: a spectral mixture kernel with 4 mixture components (SegCPGP-SM4), a Matern kernel with smoothness 5/2 (SegCPGP-Mat52), a squared exponential kernel (SegCPGP-RBF), and a linear kernel (SegCPGP-Lin). The number of mixture components was selected such that the spectral mixture kernel is sufficiently expressive. The other Gaussian process based method, ADAGA, is combined with these same kernels, except for the spectral mixture kernel, due to software version incompatibilities.

Synthetic Data We evaluate the performance of several CPD methods on mean, variance, periodicity, and trend changes.

By combining the changepoint kernel with various base kernels, we can create changepoint datasets with predefined change locations and transition steepness. Trend and periodicity change datasets are not generated with the changepoint kernel.

For each change category, ten 400-point datasets are generated, each containing three change points at index 100, 200 and 300. Section D of the appendix provides the exact generative parameters of each of the datasets and examples for each of the change categories.

Table 1 shows F_1 scores for a variety of changepoint detection methods for mean, variance, trend and periodicity changes, as well as each method's average. SegCPGP, when combined with the 4-component spectral mixture kernel, achieves particularly strong overall performance. In particular, it is the best performing method for the trend and periodicity change category, although the performance of SegCPGP with the Matern52 or RBF kernels is not significantly different. In the mean and trend change categories, multiple methods perform equally well to the best performing methods (ECP for mean changes, BOCPD for variance changes) according to the Wilcoxon signed-rank test. For mean changes, SegCPGP with linear or RBF base kernels is able to perform equally well to the best performing method; the SM4 kernel also leads to good results, but does perform significantly differently from ECP.

The proposed method, SegCPGP, using either the SM4 or Matern52 kernel, demonstrates strong performance overall and across various change categories, where the spectral mixture kernel may be preferred for the trend and periodicity changes, the RBF kernel may be preferred for mean changes, and the Matern52 kernel may be preferred for variance changes.

Benchmark Data The performance of SegCPGP is evaluated on the Turing changepoint detection benchmark datasets [Van den Burg and Williams, 2020]. The datasets are annotated by multiple experts. We incorporate the same datasets as in Caldarelli et al. [2022]: Business Inventories (businv), Ozone (ozone), and GDPs of Japan, Iran and Argentina (gdp_argentina, gdp_iran, gdp_japan). We omit the Run Log dataset, as our method does not yet support multivariate datasets. All datasets are standardized to have zero mean and unit variance. Appendix E provides more extensive descriptions of the benchmark datasets.

Table 2 displays the F_1-score for various CPD algorithms on each of the benchmark datasets, as well as each method's average F_1-score. SegCPGP-SM4 obtains the highest F_1 score on GDP Japan. For the GDP Argentina dataset, SegCPGP-SM4 performs on par with BOCPD, obtaining the highest scores. PELT detects the changepoints perfectly on the ozone dataset, while SegCPGP-Lin and SegCPGP-SM4, as well as ADAGA-Lin and ADAGA-Mat52, achieve the second best score of 0.966. On both the Business Inventories and GDP Japan dataset, SegCPGP does not outperform the zero method. For the Business Inventories dataset, only the ADAGA-based methods outperform the ZERO method; for the GDP Japan dataset, no method outscores the ZERO method.

On average, SegCPGP-SM4 obtains the highest F_1 score in absolute terms (0.790), closely followed by ADAGA-Matern52 (0.786) and SegCPGP-Lin (0.784), highlighting the utility of Gaussian process-based changepoint detection methods.

Calibration In this section we provide an empirical analysis of the FPR and FNR for ADAGA and SegCPGP. We generate 3000 random mean-change datasets. On each dataset, ADAGA is fit with $\delta = 0.3$ and $\delta = 0.6$. SegCPGP is fit with $p = 0.05$ and $p = 0.1$. For each fit and each method the number of true positives (TPs), false positives (FPs), true negatives (TNs) and false negatives (FNs) are computed; we again consider changepoints estimated within 5 time steps of the true changepoint TPs. We elaborate on the computation of TPs, FPs, TNs and FNs, as well as the FNR and FPR in Appendix H.3.

As a heuristic to detect changepoints, ADAGA [Caldarelli et al., 2022] uses a likelihood ratio test (LRT) at the window level: A Gaussian process regression is fit locally, on a data window $\mathcal{W}$, and another GPR is fit on a subwindow $\mathcal{S}$. If the fit on $\mathcal{W}$ differs from the fit on $\mathcal{S}$ according to the LRT, a changepoint is detected. The threshold δ for the likelihood ratio statistic is chosen such that the probability of a Type I and Type II error on the window level is at most δ. Since the null and alternative hypotheses are defined for GPRs fit on the window and sub-window only, the bounds derived in Caldarelli et al. [2022] may not hold for the entire signal.

Table 1: F_1 score per method for synthetic datasets, grouped by change type. Each F_1 score is the method's average over that change category, across 10 datasets. Methods that do not perform differently from the best-performing method (**bold**) according to a Holm-corrected Wilcoxon signed-rank ($p = 0.05$) test are indicated with an *. SegCPGP-SM4 performs well overall. Most methods, including the ZERO method, perform equally well to the best performing method on the mean change datasets.

Change category	ADAGA (Linear)	ADAGA (Mat'ern52)	ADAGA (RBF)	BinSeg	BOCPD	CPNP	ECP	KCPA	PELT	RBOCPDMS	SegCPGP (SM4)	SegCPGP (Linear)	SegCPGP (Matern)	SegCPGP (RBF)	ZERO
mean	0.638	0.376	0.395	0.943*	0.839*	0.916*	0.975*	0.020	0.943*	0.462	0.888*	0.811*	0.946*	**0.986***	0.400
periodicity	0.396	0.392	0.409	0.393	0.401	0.358	0.403	0.020	0.381	0.513	**0.877***	0.393	0.738*	0.851*	0.400
trend	0.242	0.308	0.289	0.256	0.160	0.223	0.180	0.020	0.263	0.227	**0.870***	0.231	0.828*	0.671*	0.400
variance	0.369	0.312	0.324	0.393	**0.738***	0.427	0.599*	0.020	0.356	0.282	0.515*	0.400	0.575*	0.699*	0.400
overall	0.411	0.347	0.354	0.496	0.534	0.481	0.539	0.020	0.486	0.371	0.787*	0.459	0.772*	**0.802***	0.400

Table 2: Comparison of several changepoint detection methods on benchmark datasets. For each dataset the best performing methods are highlighted in bold; the best overall mean F_1 score is also in bold. SegCPGP performs comparably to the best performing methods in the benchmark. Note that none of the methods performs differently from the zero method, according to a Holm-corrected Wilcoxon signed-rank test with $p = 0.05$.

	businv	gdp_argentina	gdp_iran	gdp_japan	ozone	mean F_1
adaga (Lin)	0.630	0.824	0.713	0.471	0.966	0.720
adaga (Matern52)	**0.723**	0.824	0.800	0.615	0.966	0.786
adaga (RBF)	0.681	0.824	0.800	0.615	0.776	0.739
binseg	0.370	0.889	0.492	0.615	0.650	0.603
bocpd	0.270	**0.947**	0.622	0.800	0.650	0.715
cpnp	0.304	0.818	0.330	0.667	0.750	0.574
ecp	0.301	0.824	0.652	**0.889**	0.723	0.697
kcpa	0.047	0.131	0.219	0.068	0.109	0.121
pelt	0.370	0.889	0.492	0.615	**1.000**	0.673
segcpgp (SM4)	0.370	**0.947**	**0.868**	0.800	0.966	**0.790**
segcpgp (Lin)	0.588	0.824	0.652	**0.889**	0.966	0.784
segcpgp (Matern52)	0.559	0.824	0.589	**0.889**	0.651	0.702
segcpgp (RBF)	0.588	0.824	0.673	**0.889**	0.750	0.745
zero	0.588	0.824	0.652	**0.889**	0.723	0.735

Table 3 shows the empirical FNR and FPR for SegCPGP and ADAGA computed across the 3000 random mean-change datasets, with various parameter settings for their hypothesis tests.

The FPR does seem to be bounded by δ. When $\delta = 0.3$ the FNR is $0.983 (> 0.3)$, while when $\delta = 0.6$ the FNR (0.245) indeed falls within the bounds. When looking to bound the false positive rate, ADAGA could be a good method to use, but the results for $\delta = 0.3$ suggest that ADAGA is sensitive to the setting of the δ parameter: changes in δ have a large effect on the trade-off between the FNR and the FPR.

For $p = 0.05$, the empirical FPR for SegCPGP is 0.074 while the FNR is 0.258; for $p = 0.1$, the empirical FPR is 0.096 and the FNR is 0.273. Thus, SegCPGP closely approximates the FPR. SegCPGP's FNR appears to be less sensitive to changes in its p-value. Thus, we can conclude SegCPGP is slightly miscalibrated, but remains reasonably reliable.

We further investigate the miscalibration by approximating the distribution of the LR statistic. Recall from Section 4.1 that the distribution of the LR statistic may not be χ^2_d under the null hypothesis. We generate 3900 noised datasets without a changepoint — the null hypothesis — from a GP with an RBF kernel. Then, we fit a changepoint Gaussian process regression to these data and collect corresponding LR statistics.

Figure 2 shows a quantile-quantile (Q-Q) plot of the empirical null distribution versus the χ^2_d distribution, for SegCPGP with RBF base kernels (left) and SegCPGP with 4-component spectral mixture base kernels (right). The Q-Q plot is slightly shifted from the diagonal (in blue), indicating that the empirical distribution has more degrees of freedom than the χ^2_d-distribution. To test whether the samples from the empirical null distribution are drawn from a χ^2_d-distribution, we apply a Kolmogorov-Smirnoff test: for both kernels, the null hypothesis of the sample coming from the χ^2_d-distribution are firmly rejected ($p < 10^{-60}$ for both kernels). The empirical null distribution closely matches a χ^2_d-distribution with a higher number of degrees of freedom.

6 DISCUSSION

Limitations and possible extensions The cubic computational complexity of GPs may be a limiting factor in large-scale CPD problems. Future extensions could address this by incorporating a sparse GP implementation of the proposed framework, for instance, by parameterizing the covariance

Table 3: Empirical FPR and FNR over 3000 samples of random mean-change datasets, for ADAGA and SegCPGP. In the ADAGA hypothesis test, the probability of Type I/II errors should be at most δ.

Model	FNR	FPR
ADAGA, $\delta = 0.3$	0.245	0.597
ADAGA, $\delta = 0.6$	0.984	0.002
SegCPGP, $p = 0.05$	0.258	0.073
SegCPGP, $p = 0.10$	0.273	0.095

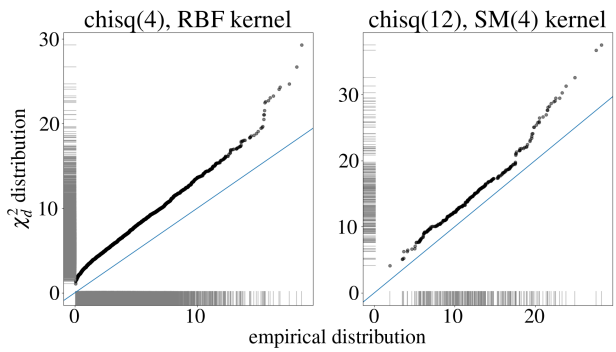


Figure 2: Quantile-quantile plot of the empirical null distribution versus the χ^2_d distribution, for changepoint GPR's with a RBF (left) and 4-component spectral mixture kernel SM4 (right). If the empirical null distribution and the χ^2_d-distribution are equal, the black scatter should lie along the blue line; here, that is not the case.

using locations of pseudoinputs [Snelson and Ghahramani, 2005]. In this case, the evidence lower bound (ELBO) could be naturally used for model selection. Additionally, an extension of SegCPGP could involve a variational implementation of GPR [Hensman et al., 2015], enabling the modeling of data with different likelihoods, thereby further increasing the versatility of the proposed approach. Furthermore, expanding SegCPGP to multivariate data, for instance via multi-output GPs, would broaden the method's applicability.

Uncertainty quantification over the location and/or number of change points could be another worthwhile extension. Using Markov chain Monte Carlo (MCMC) methods, one can obtain a distribution over the changepoint locations, similar to Green [1995]. However, this would require deriving the posterior over changepoint locations, as well as an efficient implementation of MCMC.

For broader applications it may be desirable to devise an automated procedure for the selection of the number of components in the spectral mixture kernel. A similar problem is considered in Gaussian Mixture Models (GMMs), which could inspire the model selection procedure for the spectral mixture kernel. For example, model compression [Chen et al., 2024] or a variational solution as proposed in [Corduneanu and Bishop, 2001] could be applied to mitigate overfitting.

Finally, while most online methods can be directly applied in the offline setting, the reverse is not true; adapting SegCPGP's hypothesis testing procedure into an online method might thus be another research direction.

Benchmark annotations When testing the pairwise differences between the methods in Table 2 with a Wilcoxon signed-rank test, we found that none of the methods perform significantly differently from the ZERO method. The *Default* experiment of Van den Burg and Williams [2020] shows a similar result, explaining that this may either be due to the small number of changepoints as compared to the total number of datapoints or be due to each method detecting a large number of false positives. Upon closer examination, we found that none of the expert annotators performs differently from the ZERO method (Appendix G).

The likely culprit is the inclusion of $t = 1$ as a trivial changepoint for all annotators as well as the predictions, which skews the F_1-score upwards. We discuss this in more detail and provide an example in Appendix H.2.

Overall, while the benchmark is certainly useful for comparing changepoint detection methods amongst themselves, we recommend excluding the ZERO method from evaluation or conducting further research to establish metrics that is less beneficial to the ZERO method.

Calibration Addressing the slight miscalibration requires estimating the null distribution separately for each kernel, but the computational cost may be too high and this estimation is beyond the scope of this work. Alternatively, deriving new statistics, as in finite mixture models [Frühwirth-Schnatter, 2006, Chen et al., 2004], or approximating the null distribution via Monte Carlo methods [Wolfe, 1971, Hogg, 1956] could be promising directions for future research in SegCPGP.

7 CONCLUSION

In this work, we introduced SegCPGP, a flexible framework for changepoint detection in the offline setting, based on Gaussian process regression (GPR). We showed that SegCPGP can detect a wide range of changes without requiring prior knowledge of their types. We tested the algorithm with various kernels and compared its performance on simulated and benchmark data sets to state-of-the-art methods. We found that SegCPGP provides better overall performance on simulated data and comparable performance on benchmark data sets.

Acknowledgements

We thank the anonymous reviewers for their helpful feedback and suggestions.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Ryan Prescott Adams and David JC MacKay. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*, 2007.

Matias Altamirano, François-Xavier Briol, and Jeremias Knoblauch. Robust and scalable bayesian online changepoint detection. In *International Conference on Machine Learning*, pages 642–663. PMLR, 2023.

Samaneh Aminikhanghahi and Diane J Cook. A survey of methods for time series change point detection. *Knowledge and Information Systems*, 51(2):339–367, 2017.

Elena Andreou and Eric Ghysels. Structural breaks in financial time series. *Handbook of Financial Time Series*, pages 839–870, 2009.

Alexander Aue and Lajos Horváth. Structural breaks in time series. *Journal of Time Series Analysis*, 34(1):1–16, 2013.

Ivan E Auger and Charles E Lawrence. Algorithms for the optimal identification of segment neighborhoods. *Bulletin of Mathematical Biology*, 51(1):39–54, 1989.

Francis Bach. Exploring large feature spaces with hierarchical multiple kernel learning. *Advances in Neural Information Processing Systems*, 21, 2008.

Alessio Benavoli, Giorgio Corani, and Francesca Mangili. Should we really use post-hoc tests based on mean-ranks? *The Journal of Machine Learning Research*, 17(1):152–161, 2016.

Edoardo Caldarelli, Philippe Wenk, Stefan Bauer, and Andreas Krause. Adaptive Gaussian process change point detection. In *International Conference on Machine Learning*, pages 2542–2571. PMLR, 2022.

Alain Celisse, Guillemette Marot, Morgane Pierre-Jean, and GJ Rigaill. New efficient algorithms for multiple change-point detection with reproducing kernels. *Computational Statistics & Data Analysis*, 128:200–220, 2018.

Hanfeng Chen, Jiahua Chen, and John D Kalbfleisch. Testing for a finite mixture model with two components. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 66(1):95–115, 2004.

Kai Chen, Twan van Laarhoven, and Elena Marchiori. Compressing spectral kernels in Gaussian process: Enhanced generalization and interpretability. *Pattern Recognition*, page 110642, 2024.

Yudong Chen, Tengyao Wang, and Richard J Samworth. High-dimensional, multiscale online changepoint detection. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1):234–266, 2022.

Adrian Corduneanu and Christopher M Bishop. Variational Bayesian model selection for mixture distributions. In *Proceedings eighth International conference on Artificial Intelligence and Statistics*, pages 27–34. Morgan Kaufmann, 2001.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

David Duvenaud, James Lloyd, Roger Grosse, Joshua Tenenbaum, and Ghahramani Zoubin. Structure discovery in nonparametric regression through compositional kernel search. In *International Conference on Machine Learning*, pages 1166–1174. PMLR, 2013.

Sylvia Frühwirth-Schnatter. *Finite mixture and Markov switching models*, pages 114–115. Springer, 2006.

Piotr Fryzlewicz. Wild binary segmentation for multiple change-point detection. *Annals of Statistics*, 42(6):2243–2281, 2014.

Roman Garnett, Michael A Osborne, and Stephen J Roberts. Sequential Bayesian prediction in the presence of changepoints. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 345–352. PMLR, 2009.

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

Zaid Harchaoui, Félicien Vallet, Alexandre Lung-Yut-Fong, and Olivier Cappé. A regularized kernel-based approach to unsupervised audio segmentation. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1665–1668. IEEE, 2009.

Kaylea Haynes, Paul Fearnhead, and Idris A Eckley. A computationally efficient nonparametric approach for changepoint detection. *Statistics and Computing*, 27:1293–1305, 2017.

James Hensman, Alexander Matthews, and Zoubin Ghahramani. Scalable variational Gaussian process classification. In *Artificial Intelligence and Statistics*, pages 351–360. PMLR, 2015.

Robert V Hogg. On the distribution of the likelihood ratio. *The Annals of Mathematical Statistics*, pages 529–532, 1956.

Nicholas A James and David S Matteson. ecp: An R package for nonparametric multiple change point analysis of multivariate data. *arXiv preprint arXiv:1309.3295*, 2013.

Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. An online algorithm for segmenting time series. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 289–296. IEEE, 2001.

Hossein Keshavarz, Clayton Scott, and XuanLong Nguyen. Optimal change point detection in Gaussian processes. *Journal of Statistical Planning and Inference*, 193:151–178, 2018.

Rebecca Killick and Idris A Eckley. changepoint: An r package for changepoint analysis. *Journal of Statistical Software*, 58:1–19, 2014.

Rebecca Killick, Paul Fearnhead, and Idris A Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107 (500):1590–1598, 2012.

Jeremias Knoblauch and Theodoros Damoulas. Spatio-temporal Bayesian on-line changepoint detection with model selection. In *International Conference on Machine Learning*, pages 2718–2727. PMLR, 2018.

Jeremias Knoblauch, Jack E Jewson, and Theodoros Damoulas. Doubly robust Bayesian inference for non-stationary streaming data with β-divergences. *Advances in Neural Information Processing Systems*, 31, 2018.

Tze Leung Lai. Sequential changepoint detection in quality control and dynamical systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(4):613–644, 1995.

Émilie Lebarbier. Detecting multiple change-points in the mean of Gaussian process by model selection. *Signal Processing*, 85(4):717–736, 2005.

David Leeftink and Max Hinne. Spectral discontinuity design: Interrupted time series with spectral mixture kernels. In *Machine Learning for Health*, pages 213–225. PMLR, 2020.

James Lloyd, David Duvenaud, Roger Grosse, Joshua Tenenbaum, and Zoubin Ghahramani. Automatic construction and natural-language description of nonparametric regression models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

David S Matteson and Nicholas A James. A nonparametric approach for multiple change point analysis of multivariate data. *Journal of the American Statistical Association*, 109(505):334–345, 2014.

Alexander G de G Matthews, Mark Van Der Wilk, Tom Nickson, Keisuke Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. GPflow: A Gaussian Process Library using TensorFlow. *J. Mach. Learn. Res.*, 18(40):1–6, 2017.

Adam B Olshen, E Seshan Venkatraman, Robert Lucito, and Michael Wigler. Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, 5(4):557–572, 2004.

Jaxk Reeves, Jien Chen, Xiaolan L Wang, Robert Lund, and Qi Qi Lu. A review and comparison of changepoint detection techniques for climate data. *Journal of Applied Meteorology and Climatology*, 46(6):900–915, 2007.

JJ O Ruanaidh, William J Fitzgerald, and Kenneth J Pope. Recursive Bayesian location of a discontinuity in time series. In *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing*, pages IV–513. IEEE, 1994.

Yunus Saatçi, Ryan D Turner, and Carl E Rasmussen. Gaussian process change point models. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 927–934. PMLR, 2010.

Andrew Jhon Scott and M Knott. A cluster analysis method for grouping means in the analysis of variance. *Biometrics*, pages 507–512, 1974.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. *Advances in Neural Information Processing Systems*, 18, 2005.

Alexander G Tartakovsky, Aleksey S Polunchenko, and Grigory Sokolov. Efficient computer network anomaly detection by changepoint detection methods. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):4–11, 2012.

Charles Truong, Laurent Oudre, and Nicolas Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167:107299, 2020.

Gerrit JJ Van den Burg and Christopher KI Williams. An evaluation of change point detection algorithms. *arXiv preprint arXiv:2003.06222*, 2020.

Lyudmila Yur'evna Vostrikova. Detecting "disorder" in multidimensional random processes. *Doklady Akademii Nauk*, 259(2):270–274, 1981.

Samuel S Wilks. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The Annals of Mathematical Statistics*, 9(1):60–62, 1938.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Andrew Wilson and Ryan Adams. Gaussian process kernels for pattern discovery and extrapolation. In *International Conference on Machine Learning*, pages 1067–1075. PMLR, 2013.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial Intelligence and Statistics*, pages 370–378. PMLR, 2016.

John H Wolfe. *A Monte Carlo study of the sampling distribution of the likelihood ratio for mixtures of multinormal distributions*, volume 72. Naval Personnel and Training Research Laboratory San Diego, 1971.

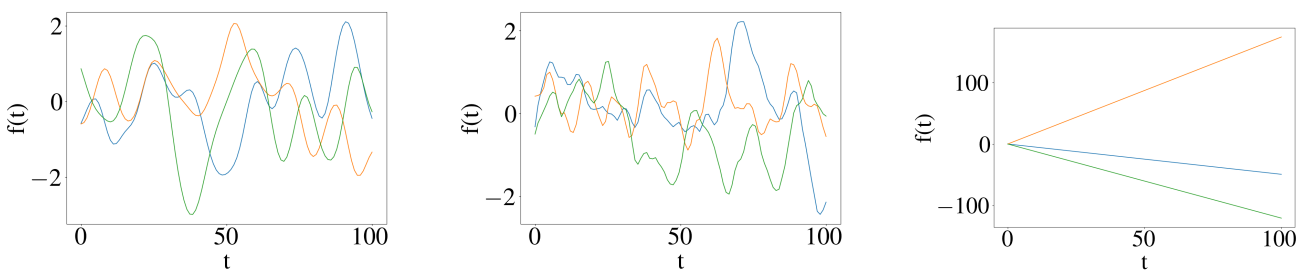
Changliang Zou, Guosheng Yin, Long Feng, and Zhaojun Wang. Nonparametric maximum likelihood approach to multiple change-point problems. *The Annals of Statistics*, 42(3):970–1002, 2014.

Revisiting Gaussian Processes For Changepoint Detection (Supplementary Material)

Janneke Verbeek[1] **Tom Heskes**[1] **Yuliya Shapovalova**[1]

[1]Radboud University Nijmegen, Institute for Computing and Information Sciences, Nijmegen, The Netherlands

A KERNEL TYPES



(a) (b) (c)

Figure 3: Illustration of samples from a squared exponential kernel with $\ell = 5, \sigma^2 = 1$ (a), samples from a Matérn kernel with $\ell = 5, \sigma^2 = 1$, and samples from a linear kernel with σ^2=1. The squared exponential kernel results in smoother functions than the Matérn kernel. The linear kernel results in straight lines.

Squared exponential kernel The squared exponential kernel is a stationary kernel — a kernel dependent on the distance between t and t' scaled by lengthscale ℓ, $\frac{||t-t'||}{\ell}$ — given by

$$k(t, t') = \sigma^2 \exp(0.5 \frac{||t - t'||}{\ell}). \tag{10}$$

where σ^2 is the variance, ℓ is the lengthscale, and $||t-t'||$ is the Euclidean distance between t and t'. The squared exponential kernel is also known as the radial basis function (RBF) kernel.

Matérn kernel A Matérn kernel with smoothness parameter $\nu = 5/2$ is given by

$$k(t, t') = \sigma^2(1 + \sqrt{5}\frac{||t - t'||}{\ell}) + 5/3\frac{||t - t'||}{\ell} \exp(-\sqrt{5}\frac{||t - t'||}{\ell}). \tag{11}$$

Linear kernel The linear kernel is given by

$$k(t, t') = \sigma^2 tt', \tag{12}$$

where σ^2 is again a variance parameter. The kernel models linear functions. Note that since this kernel does not depend on $||t - t'||$, it is nonstationary.

Figure 3 illustrates samples from a Gaussian process prior with (a) squared exponential, (b) Matérn and (c) linear kernels.

B STEEPNESS IN THE CHANGEPOINT KERNEL

Below, we provide a detail analysis for the effect of setting the steepness parameter in the changepoint kernel to ∞.

The specification of the CP kernel is

$$k(f(t), f(t)) = k_1(t, t')\psi(t, t') + k_2(t, t'), \bar{\psi}(t, t'),$$

where for a location t_0 and steepness s,

$$\psi(t, t') = \psi(t)\psi(t') = \frac{1}{1 + \exp(-s(t - t_0))} \times \frac{1}{1 + \exp(-s(t' - t_0))},$$

and

$$\bar{\psi}(t, t') = (1 - \psi(t))(1 - \psi(t'))$$

In the case that $s = \infty$, the components $\psi(t, t')$ and $\bar{\psi}(t, t')$ are driven to 1 and 0, respectively:

$$\psi(t, t') = \psi(t)\psi(t') = \frac{1}{1 + \exp(-\infty(t - t_0))} \times \frac{1}{1 + \exp(-\infty(t' - t_0))},$$

then

$$\psi(t, t') = \psi(t)\psi(t') = \frac{1}{1 + 0} \times \frac{1}{1 + 0},$$

and thus

$$\psi(t, t') = 1;$$
$$\bar{\psi}(t, t') = (1 - \psi(t))(1 - \psi(t')) = 0.$$

so we would conclude that then the changepoint kernel becomes equivalent to the first base kernel,

$$k(f(t), f(t)) = k_1(t, t').$$

When the location of the changepoint is moved to one of the extremes of the data window (say, $t_0 = 0$), we instead get

$$\psi(t, t') = \psi(t)\psi(t') = \frac{1}{1 + \exp(-s(t - 0))} \times \frac{1}{1 + \exp(-s(t' - 0))}.$$

We will briefly discuss the difference between setting the steepness to an extreme versus moving the location to an extreme (i.e. the edge of a window).

If the location of the change is at, or even beyond the edge of the window, both kernels can still describe the signal in the window if the steepness is sufficiently low. In Figure 4, we visualize this in one dimension by plotting the area influenced by each base kernel via sigmoids. The location t_0 is plotted with the red dotted line. We have shaded the area influenced by kernel 1 blue and the area influenced by kernel 2 orange. As can be seen from the figure, even if the location is placed outside the right bound of the data window, the signal in the window would still be modeled by both kernels.

In practice the steepness can of course be set to some high value, which would result in a similar effect as for $s = \infty$, that is, only one of the kernels will describe the signal in the window. Figure 5 shows the effect for steepness 500.

In conclusion, the effect of moving the location is different from setting the steepness to infinity.

C SEGCPGP ALGORITHM

Algorithm 1 shows pseudocode for the SegCPGP procedure.

D SYNTHETIC DATA GENERATION

We describe the generation process for the synthetic datasets. Examples of datasets from each change category are found in Figure 6.

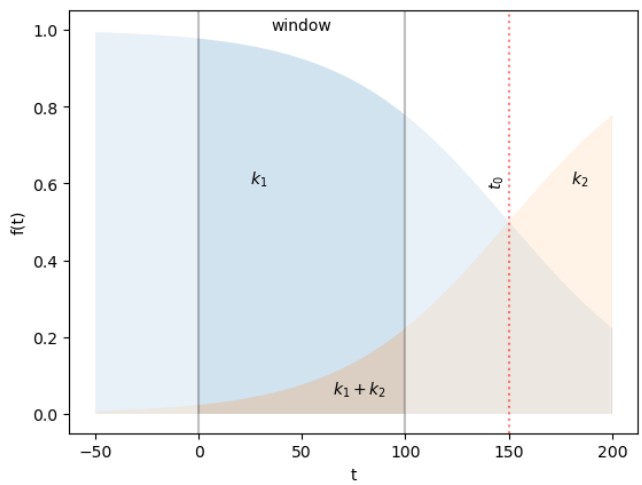


Figure 4: Visualization of the effect of setting the steepness parameter to a low value, while the location of the changepoint is outside the window.

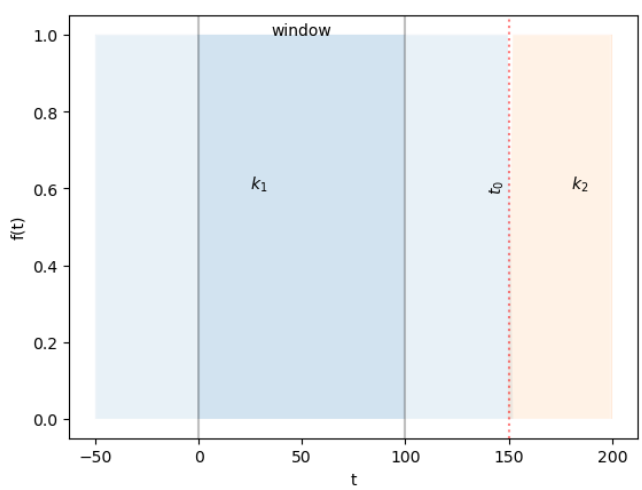


Figure 5: Visualization of the effect of setting the steepness parameter to 500, while the location of the changepoint is outside the window.

Multiple changepoints The changepoint kernel can be extended to multiple change points

Let $[k_c]_{c=0}^{C}$ denote a list of kernels. For C kernels, there are $C-1$ changepoints, and the kernel for multiple changepoints becomes

$$\text{cov}(f(t), f(t')) = \bar{\psi}_1(t,t')k_1(t,t') + \sum_{c=0}^{C-2}(\psi_c(t,t')\bar{\psi}_{c+1}(t,t')k_c(t,t')\ ) + \psi_{C-1}(t,t')k_C(t,t'). \tag{13}$$

Mean changes Mean change data is sampled from a changepoint Gaussian process with a list of Constant kernels. A constant kernel,

$$k(t,t') = \sigma^2, \tag{14}$$

has only a variance parameter σ^2. To the mean change data, we add Gaussian noise with mean 0 and variance 0.01.

Variance changes Variance change data is sampled from a changepoint Gaussian process with a list of noise (or white) kernels. The noise kernel,

$$k(t_i, t_j) = \delta_{ij}\sigma^2, \tag{15}$$

Algorithm 1 Segmenting CPGP

location $\leftarrow []$
procedure SEGMENTINGCPGP(X, y, k_1, k_2)
  location $\leftarrow \min_x X + (\max_x X - \min_x X)/2)$
  steepness $\leftarrow 1$
  $M_1 :=$ GPR($X, y,$ CHANGEPOINT($k_1, k_2,$ *location*, *steepness*))
  $M_0 :=$ GPR(X, y, k_1)
  for M in $[M_0, M_1]$ **do**:
    $\hat{M} \leftarrow$ OPTIMIZE(M)
  $df \leftarrow |M_1| - |M_0|$
  $\mathcal{R} \leftarrow -2 \log p(y|\hat{M}_1) - \log p(y|\hat{M}_0)$
  $p \leftarrow \chi^2(\mathcal{R}, df)$
  if $p > r$ **then return**
  if $p <= r$ **then**
    location $\leftarrow \hat{M}_1$.*location*
    steepness $\leftarrow \hat{M}_1$.*steepness*
    $\epsilon \leftarrow 5$
    if $\min_x X <$ *location* $< \max_x X$ **then**
      $X_{\text{left}}, X_{\text{right}} \leftarrow X[:$ *location* $+ \epsilon], X[$*location* $- \epsilon :]$
      $y_{\text{left}}, y_{\text{right}} \leftarrow y[:$ *location* $+ \epsilon], y[$*location* $- \epsilon :]$
      SEGMENTINGCPGP($X_{\text{left}}, y_{\text{left}}, k_1, k_2$)
      SEGMENTINGCPGP($X_{\text{right}}, y_{\text{right}}, k_1, k_2$)
    return
  return

where $\delta_{ij} = 1$ if $i = j$, and 0 otherwise, and σ^2 is again the variance parameter. To ensure that there exist variance changes, the variance parameter is 1 when the kernel index c is even, and sampled from $[3, 20)$ otherwise. As the variance data represents changes in noise, we do not add extra noise to the variance change data.

Trend changes Trend changes are generated according to the well-known line equation $f(t) = at + b$, where in each segment the slope a is randomly sampled from $U(0, 2)$, and the sign of the slope switches for each segment.

Since our objective is to test the ability of each method to detect only one particular type of change, it is crucial that no other types of changes occur in the signal. A bias term b is thus added to the signal, such that there are no jumps in the signal. To the trend change data, we add Gaussian noise with mean 0 and variance 1.

Periodicity changes A periodic signal is generated according to a sine wave, $f(t) = \sin(\omega t)$, where the angular frequency ω is randomly sampled from $[1, 100)$. As we stated earlier, we only want to test the detection capacity of the benchmark models on a single change type. We therefore do not change the amplitude of the signal, since that might be interpreted as a change in variance. To the periodicity change data, we add Gaussian noise with mean 0 and variance 0.1.

E DESCRIPTION OF BENCHMARK DATASETS

We briefly describe the benchmark datasets used in this paper, which were originally presented in Van den Burg and Williams [2020]. All datasets used in this paper are univariate.

Business Inventories The Business Inventories dataset contains United States monthly total business inventories. The length of the dataset is 330. The minimum amount of changepoints found by annotators is 0; the maximum amount of changepoints found by annotators is 3.

GDP Argentina The GDP Argentina contains the gross domestic product of Argentina, measured from 1960 up to 2019. The dataset has length 59. The minimum amount of changepoints found by annotators is 0; the maximum amount of changepoints found by annotators is 3.

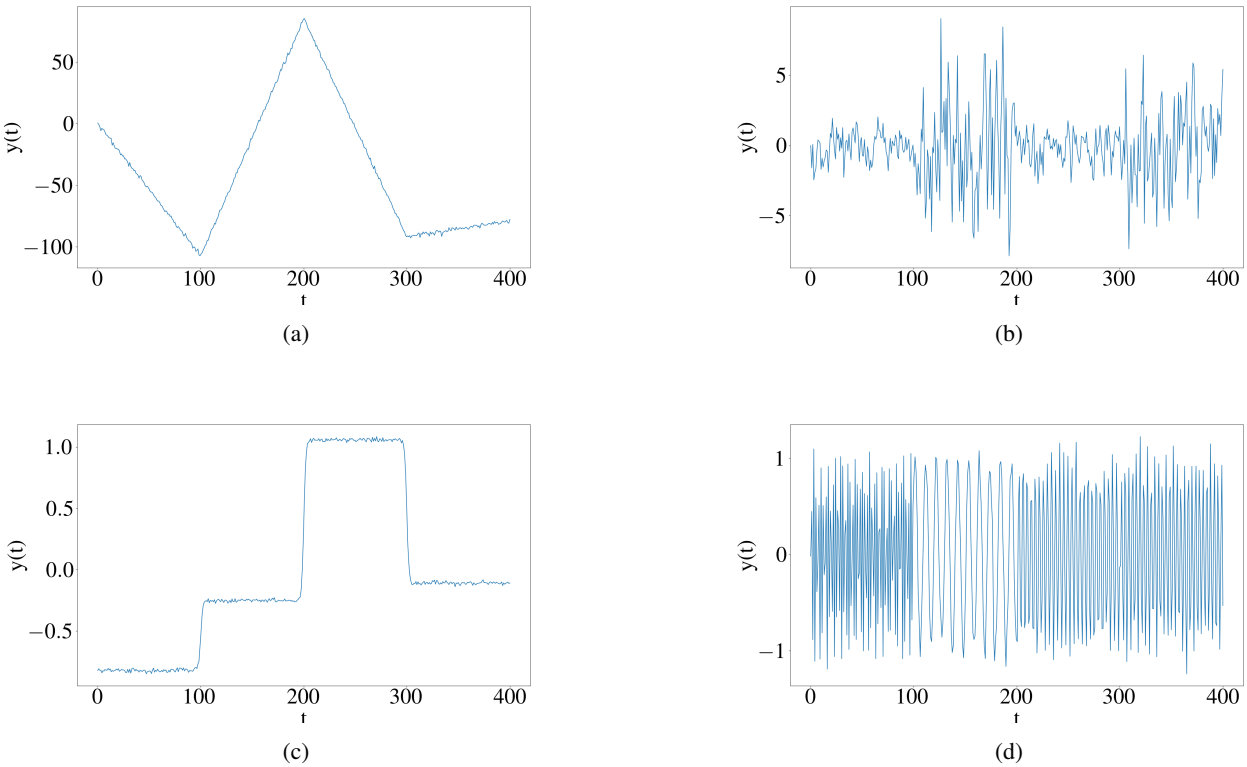


Figure 6: Examples of synthetic trend (a), variance (b), mean (c) and periodicity (d) changepoint dataset. The datasets include 400 samples, and changepoints at locations 100, 200 and 300. The steepness of the mean and variance changepoints, which were generated from changepoint Gaussian processes, is 1.

GDP Iran The GDP Iran dataset contains the gross domestic product of Iran, measured from 1960 to 2020. The dataset has length 58. The minimum amount of changepoints found by annotators is 0; the maximum amount of changepoints found by annotators is 3.

GDP Japan The GDP Japan dataset contains the gross domestic product of Japan measured yearly from 1960 to 2020. The dataset has length 58. The minimum amount of changepoints found by annotators is 0; the maximum amount of changepoints found by annotators is 1.

Ozone The Ozone dataset contains yearly measurements of the global emissions of ozone-depleting substances. The dataset has length 54. The minimum amount of changepoints found by annotators is 0; the maximum amount of changepoints found by annotators is 1.

F HYPERPARAMETERS OF COMPARED MODELS

We give an overview of the hyperparameters used for the models in our synthetic data and benchmark experiments.

F.1 GAUSSIAN PROCESS-BASED MODELS

Both ADAGA and SegCPGP use GPFlow [Matthews et al., 2017], a Python package implementing Gaussian processes and Gaussian process regression in TensorFlow, [Abadi et al., 2015]. The kernels used in both ADAGA and SegCPGP use their default hyperparameters from the GPFlow package.

ADAGA For ADAGA, as in Caldarelli et al. [2022], the minimal window size is set to 15 and the batch size is set to 1. δ is set to 0.6 by default. The version used in the benchmark experiment is the inducing points version; an implementation can

be found here.

SegCPGP The p-value for SegCPGP is set to 0.1 by default.

F.2 TURING CHANGEPOINT DATASET BENCHMARK METHODS

We briefly describe some specific hyperparameters used in Default setting for the Turing Changepoint Dataset Benchmark (TCPDBench). TCPDBench uses methods implemented in Python and R, which can be found here. In principle, running this benchmark after cloning the TCPDBench repository should already have the default parameters set correctly. The parameters of the default experiment are also described in Van den Burg and Williams [2020]; for completeness, we also describe them here.

Where possible, links to the documentation of the original packages are provided.

BinSeg & PELT The implementations of BinSeg and PELT originate from the changepoint R package [Killick and Eckley, 2014]. Both methods by default try to find a change in mean. They both use the Modified Bayesian Information Criterion as penalty. The test statistic used by both methods is the Normal test statistic, which assumes a normal distribution for the errors.

CPNP The documentation for CPNP, a nonparametric version of PELT implemented in R, is found here. In TCPDBench, the number of quantiles is set to 10.

ECP & KCPA Kernel Change Point Analysis, proposed by [Harchaoui et al., 2009] combines the kernel trick and dynamic programming to detect changepoints. The constant penalty of KCPA is set to 1.0; the maximum number of changepoints is set to the maximum number possible.

Energy change points, or ECP, was proposed by Matteson and James [2014]. The parameter α of ECP is set to 1. The minimum number of timesteps between changepoints is set to 30; 199 random permutations are used in each permutation test; the significance level is set to 0.05.

The documentation for both methods can be found here, [James and Matteson, 2013].

BOCPD The implementation of Bayesian online changepoint detection (BOCPD) is the one found in the Online Change-Point (OCP) package. The documentation for the OCP package can be found here. The prior parameters a, b and k are all set to 1. The hazard function intensity `lambda` is set to 100.

RBOCPDMS The authors of Van den Burg and Williams [2020] also created RBOCPDMS Knoblauch et al. [2018]. For the benchmark, the code is run from this repository. In case of RBOCPDMS, the run length is pruned to the best 100 run lengths; α_0 and α_{rld} were both set to 0.5. The timeout for RBOCPDMS is set to 4 hours by default for the benchmark experiment.

G ONE-VERSUS-REST F_1-SCORES

For each of the annotators in the annotations of Van den Burg and Williams [2020], we compute their one-versus-rest F_1-score for each benchmark dataset. Then, we compute the ZERO-versus-rest F_1 score for each benchmark dataset. We compare their pairwise differences using the Wilcoxon signed-rank test described earlier. We also added a PERFECT method, which (artificially) obtains an F_1-score of 1 on every single dataset.

The one versus rest F_1 scores for each annotator are found in Table 4. Both the ZERO and PERFECT method are included; the ZERO method never returns any changepoints, while the PERFECT method artificially obtains an F_1-score of 1 on each dataset. Table 4 displays the results. Unfortunately, none of the annotators, including the PERFECT annotator, performs significantly differently from the ZERO method.

In conclusion, if not any single expert annotator nor a perfect score can perform differently from the ZERO method, we conjecture that any changepoint algorithm set loose on this benchmark is faced with an impossible task.

Table 4: One-versus-rest F_1 scores for every annotator versus the rest of the annotators, for each dataset. Not all datasets have been annotated by all annotators; missing values are represented with —. The ZERO method never returns any changepoints; the PERFECT method artificially returns an F_1-score of 1 for every dataset.

annot.	businv	gdp_argentina	gdp_iran	gdp_japan	ozone
6	1.000	0.769	0.829	0.857	0.957
7	0.426	0.769	—	0.857	0.957
8	0.897	0.769	0.523	0.857	0.629
9	1.000	—	0.857	1.000	—
10	—	—	0.968	—	0.957
12	—	1.000	0.968	1.000	0.800
13	1.000	1.000	—	—	—
ZERO	0.588	0.824	0.652	0.889	0.723
PERFECT	1.000	1.000	1.000	1.000	1.000

H CLASSIFICATION MEASURES

Changepoint detection can be evaluated as a classification problem, when finding the locations of the changepoints is of interest. In this section we give a detailed description of the computation of the F_1-score for changepoint detection, as also presented in Van den Burg and Williams [2020]; then, we highlight a problem with the F_1-score when a trivial changepoint is included. Finally, we describe how the false negative and false positive rate are computed in our experiments.

H.1 THE F_1-SCORE

In the context of changepoint detection, a true positive (TP) is any changepoint detected within a certain margin from the true changepoint [Van den Burg and Williams, 2020, Killick et al., 2012, Truong et al., 2020]. Let $\mathcal{X}$ denote the predictions of some changepoint detection algorithm on some dataset. Assume there are K annotators, that each provide an annotation, so that the set of all annotations is $\mathcal{T} = \{\mathcal{T}_k\}_{k=1}^K$. Since some of the annotators may naturally identify the same change points, we also define the set of unique annotations as $\mathcal{T}^* = \bigcup_k \{\mathcal{T}_k\}$.

Let $\text{TP}(\mathcal{X}, \mathcal{T}^*)$ be a set-based evaluation of true positives for predictions $\mathcal{X}$ and the set of all unique annotators $\mathcal{T}^*$,

$$\text{TP}(\mathcal{X}, \mathcal{T}^*) = \{\forall t \in \mathcal{X}, \forall \tau \in \mathcal{T}^* : |t - \tau| \leq M\},$$

and $\text{TP}(\mathcal{X}, \mathcal{T}_k)$ be the true positives found by annotator $k \in K$,

$$\text{TP}(\mathcal{X}, \mathcal{T}_k) = \{\forall t \in \mathcal{X}, \forall \tau \in \mathcal{T}_k : |t - \tau| \leq M\}.$$

where M is some margin around the true changepoint. Generally, $M \geq 0$, but M is usually set to 5 time steps in practice.

The precision (P) is calculated as the proportion of detected change points by the algorithm that are true positives,

$$\text{P} = \frac{\text{TP}(\mathcal{X}, \mathcal{T}^*)}{|\mathcal{X}|},$$

the recall (R) is calculated as the average true positives, computed over all annotators,

$$\text{R} = \frac{1}{K} \sum_{k=1}^{K} \frac{\text{TP}(\mathcal{X}, \mathcal{T}_k)}{|\mathcal{T}_k|}.$$

The F_1-score is then computed as

$$F_1 = 2 \cdot \frac{\text{P} \cdot \text{R}}{\text{P} + \text{R}}.$$

H.2 TRIVIAL CHANGEPOINTS

The F_1-score as defined in Van den Burg and Williams [2020] and Appendix H.1 adds the trivial changepoint $t = 1$ to all annotations, as well as to all predictions. While necessary to prevent the F_1-score from being undefined in case no changepoints are found or annotated, the F_1-score behaves strangely when no changepoints are detected by the algorithm. Due to the trivial changepoint, the precision of the ZERO method is always 1, and in cases where not many changepoints are annotated this will lead to unreasonably high F_1-scores.

Example Consider a dataset where three annotators provide the changepoints $\mathcal{T} = \{[45], [50]\}$, making $\mathcal{T}^* = \{45, 50\}$. Assuming the annotators are experts, it is reasonable to assume there is some unknown true changepoint around $t = 45$ to $t = 50$.

Now consider a ZERO method, which always gives $\mathcal{X} = \varnothing$ as a prediction. In order to compute the F_1-score, we add the trivial changepoint 1 to both the predictions and all annotations, so we have the annotations $\{[1, 45], [1, 50]\}$, which makes $\mathcal{T}^* = \{1, 45, 50\}$, and $\mathcal{X} = \{1\}$.

Computing the precision then leads to

$$\mathrm{P} = \frac{|\{1\}|}{|\{1\}|} = 1,$$

$$\mathrm{R} = \frac{1}{2}\sum_{k=1}^{K} \frac{|\{1\}|}{|\{1, 45\}|} + \frac{|\{1\}|}{|\{1, 50\}|} = \frac{1}{2}(\frac{1}{2} + \frac{1}{2}) = \frac{1}{2},$$

$$F_1 = 2 \cdot \frac{\mathrm{P} \cdot \mathrm{R}}{\mathrm{P} + \mathrm{R}} = 2 \cdot \frac{1/2}{3/2} = \frac{2}{3} \approx 0.67.$$

It is easy to see from this example that the inclusion of the trivial changepoints means that the ZERO method will always get a precision of 1 without finding any changepoint. Furthermore, without agreeing with any of the annotators, the ZERO method gets an F_1-score of 0.67. Thus, although it is necessary to include the trivial changepoint to prevent the precision and recall from being undefined, the subsequent results are arguably unreasonable.

In our synthetic data experiment, the tested methods did mostly manage to perform differently from the ZERO method. If the annotators provide enough unique changepoints (in this case, there were three ground truth changepoints), the recall will be somewhat lower — though we still think it is unreasonably high — and the tested methods are actually capable of performing differently from the ZERO method.

H.3 FALSE POSITIVE AND FALSE NEGATIVE RATE

In order to compute the true and false positives (TP and FP, respectively) and the true and false negatives (TN and FN, respectively), we use a similar method as in Appendix H.1, except that $\mathcal{X}$ now contains all performed hypothesis tests. We denote a hypothesis test by $h(t)$, which tests some location t and returns

$$h(t) = \begin{cases} H_0, & \text{if the null hypothesis cannot be rejected} \\ H_1, & \text{otherwise.} \end{cases}$$

A true positive is then a situation where the tested location t is within the margin of the true changepoint, and the test indicates H_1,

$$\mathrm{TP}(\mathcal{X}, \mathcal{T}^*) = \{\forall t \in \mathcal{X}, \forall \tau \in \mathcal{T}^* : |t - \tau| \leq M \wedge h(t) = H_1\},$$

whereas a false positive is when the tested location t is outside the margin of the true changepoint, and the test indicates H_0,

$$\mathrm{FP}(\mathcal{X}, \mathcal{T}^*) = \{\forall t \in \mathcal{X}, \forall \tau \in \mathcal{T}^* : |t - \tau| > M \wedge h(t) = H_1\}.$$

In contrast, a true negative is a situation where the tested location is outside the margin around the changepoint, and the test indicates H_0,

$$\mathrm{TN}(\mathcal{X}, \mathcal{T}^*) = \{\forall t \in \mathcal{X}, \forall \tau \in \mathcal{T}^* : |t - \tau| > M \wedge h(t) = H_0\},$$

and a false negative is a situation where the tested location x is inside the margin around the changepoint, and the test indicates H_0,

$$\text{FN}(\mathcal{X}, \mathcal{T}^*) = \{\forall x \in \mathcal{X}, \forall \tau \in \mathcal{T}^* : |t - \tau| > M \wedge h(t) = H_0\}.$$

The FNR and FPR, as used in Section 5 of the main paper, are computed as

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}},$$

and

$$\text{FNR} = \frac{\text{FN}}{\text{FN} + \text{TP}}.$$