BayesProject: Fast computation of a projection direction for multivariate changepoint detection

Georg Hahn¹ · Paul Fearnhead¹ · Idris A. Eckley¹

Received: 16 July 2019 / Accepted: 23 July 2020 / Published online: 1 August 2020
© The Author(s) 2020

Abstract

This article focuses on the challenging problem of efficiently detecting changes in mean within multivariate data sequences. Multivariate changepoints can be detected by projecting a multivariate series to a univariate one using a suitable projection direction that preserves a maximal proportion of signal information. However, for some existing approaches the computation of such a projection direction can scale unfavourably with the number of series and might rely on additional assumptions on the data sequences, thus limiting their generality. We introduce BayesProject, a computationally inexpensive Bayesian approach to compute a projection direction in such a setting. The proposed approach allows the incorporation of prior knowledge of the changepoint scenario, when such information is available, which can help to increase the accuracy of the method. A simulation study shows that BayesProject is robust, yields projections close to the oracle projection direction and, moreover, that its accuracy in detecting changepoints is comparable to, or better than, existing algorithms while scaling linearly with the number of series.

Keywords Multivariate data sequence · Segmentation · Dimension reduction · Structural break · Breakpoint · Cusum

1 Introduction

Changepoint detection is an area of research with immediate practical applications in the monitoring of financial data (Bai and Perron 1998; Frick et al. 2014), network traffic data (Lévy-Leduc and Roueff 2009; Lung-Yut-Fong et al. 2012), as well as bioinformatics (Maidstone et al. 2017; Guédon 2013), environmental (Nam et al. 2015) and signal or speech processing applications (Desobry et al. 2005; Haynes et al. 2017). For instance, a sudden change in (mean) activity of one or more data streams in a network could hint at an intruder sending data to an unknown host from several infected computers. In general, a user is faced with the task to monitor several data streams simultaneously, ideally in real time, with the aim of detecting a change occurring in at least one series as soon as it occurs.

Starting with the work of Page (1954), considerable contributions have been made to the literature on changepoint detection in the areas of both methodology and applications. Of particular interest for the work we present are the developments in multivariate changepoint detection. Other aspects of changepoint detection address nonparametric asymptotic tests (Aue et al. 2009), scan and segmentation algorithms based on Chi-squared statistics (Zhang et al. 2010), detection of mean changes in intervals for biomedical applications (Siegmund et al. 2011), consistent nonparametric estimation of the number and locations of changepoints (Matteson and James 2012) and detection of multiple structural breaks in the autocovariance of a multivariate piecewise stationary process (Preuß et al. 2015). See Horváth and Rice (2014) and Truong et al. (2018) for a comprehensive overview of the pertinent literature on changepoint analysis.

The Inspect algorithm of Wang and Samworth (2017) is of particular interest, introducing a new approach for the detection of changes in a multivariate data sequence. Detection is achieved by projecting the multivariate data to a univariate series in which a changepoint is detected via a cusum approach, thus reducing the problem to the known univariate case. However, estimating the projection direction scales unfavourably with the number of series in Inspect due to the

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s11222-020-09966-2>) contains supplementary material, which is available to authorized users.

✉ Georg Hahn
ghahn@cantab.net

¹ Department of Mathematics and Statistics, Lancaster University, Lancaster LA1 4YF, UK

fact that the projection vector is computed as the leading left singular vector of a matrix (calculated via a convex optimisation problem or a cusum transform). This SVD computation leads to an algorithmic runtime of higher order in the number of series.

Our article presents an alternative Bayesian framework to compute an estimate of the projection direction. One of the main features of the proposed framework is that it permits the incorporation of prior knowledge of the changepoint scenario under investigation through the choice of a suitable prior on the change in each series, when such information is available. Its computational effort is linear in the number of series and superlinear in the series length.

The problem of detecting changes in multivariate data has received increasing attention over recent years. We compare the performance of our proposed algorithm not only to Inspect, but also to state-of-the-art algorithms from the following papers. Cho and Fryzlewicz (2015) propose to only consider the cusum statistics of those series which pass a certain threshold, thereby reducing the impact of noise. Cho (2016) proposes a double cusum statistic to aggregate cusum statistics in high dimensions and proves consistency in single and multiple changepoint detection. Enikeeva and Harchaoui (2019) propose a two-component test to detect a change in mean in a sequence of high-dimensional Gaussian vectors which yields optimal rates of convergence for a wide range of changepoint regimes. Yu and Chen (2020) introduce a Gaussian multiplier bootstrap to calibrate the critical values for the cusum statistic in high dimensions, as well as an estimator for the changepoint location which achieves a near-optimal convergence rate. Grundy et al. (2020b) propose to project a multivariate dataset to two dimensions instead of one, allowing the detection of a change in mean and variance by applying univariate changepoint detection methods to the two projected series.

The article is structured as follows. Section 2 formulates the changepoint setting under consideration (Sect. 2.1) and revisits the Inspect algorithm (Sect. 2.2) before introducing the alternative Bayesian approach for computing the projection direction (Sect. 2.3). Section 3 considers practical implementation details, including threshold calibration (Sect. 3.1), the computation of a grid for projecting the data (Sect. 3.2) and pseudo-code of the proposed algorithm (Sect. 3.3). A simulation study, presented in Sect. 4, evaluates the original Inspect algorithm, the proposed Bayesian projection approach and two classical methods (sum-cusum and max-cusum) with respect to the accuracy of the estimated projection direction, the accuracy of the resulting changepoint and their runtime. The simulation scenarios investigate the dependence on a range of features: the number of series and the series length, the proportion of series with a change, the size of the change and the location of the changepoint. Results for multiple changepoint detection on simulated data

are presented in Sect. 5. In the article, $\|\cdot\|_q$ denotes the L_q norm of a vector and $|\cdot|$ denotes the absolute value of a scalar or the size of a set.

2 Changepoint detection with projections

2.1 Mathematical formulation of the problem

Consider the problem of estimating the location of a single changepoint in a high-dimensional dataset, $X \in \mathbb{R}^{p \times n}$, composed of independent p -dimensional normal random column vectors

$$X_t \sim N_p(\mu^{(t)}, \kappa^2 I_p),$$

with parameters $\mu^{(t)} \in \mathbb{R}^p$, $\kappa^2 > 0$ for $t \in \{1, \dots, n\}$, where $0 \leq d \leq p$ out of the p series exhibit a changepoint at ζ , meaning that $\mu_A := \mu^{(1)} = \dots = \mu^{(\zeta)}$ and $\mu^{(\zeta+1)} = \dots = \mu^{(n)} =: \mu_B$ under the condition that $\left| \left\{ i : \mu_i^{(\zeta)} \neq \mu_i^{(\zeta+1)} \right\} \right| = d$. Here, μ_A (μ_B) is the mean before (after) the change, and I_p denotes the $p \times p$ identity matrix. The goal is to estimate the location of the changepoint, ζ .

One particularly simple way to attempt to find the time point, ζ , is by computing a univariate cusum transform for each of the p series (Page 1954). The cusum transform of the univariate series $X_{i,\cdot}$ is a scaled difference of means before and after each assumed changepoint location $t \in \{1, \dots, n\}$. Let $T(\cdot)$ denote the row-wise cusum transform of the p series, given by

$$[T(X)]_{i,t} = \sqrt{\frac{t(n-t)}{n}} \left(\frac{1}{n-t} \sum_{j=t+1}^n X_{i,j} - \frac{1}{t} \sum_{j=1}^t X_{i,j} \right) \quad (1)$$

for $i = 1, \dots, p$ and $t = 1, \dots, n-1$. A changepoint is declared in each series i , individually, if at any location, the cusum transform $T(X)_{i,\cdot}$ exceeds some prespecified threshold $\tau > 0$, with the changepoint location in series i being $\arg \max_t T(X)_{i,t}$.

To find a multivariate changepoint, some form of aggregation of the cusum information from all p series is required, thus effectively resulting in a reduction of dimensionality. Two simple aggregation approaches include *sum-cusum* and *max-cusum*. In *sum-cusum*, the p cusum transforms are summed (and averaged) for each assumed location $t \in \{1, \dots, n\}$, and a changepoint is declared if the average exceeds a prespecified threshold. The *sum-cusum* approach works well if most series change; however, it can lose power in the situation where the changes are sparse. In *max-cusum*, the cusum information is aggregated by computing

the maximum across all p series for each $t \in \{1, \dots, n\}$, $\max_{i \in \{1, \dots, p\}} T(X)_{i,t}$, and by declaring a changepoint once the maximum exceeds τ . This approach works well if changes are sparse but pronounced. The computational complexity of both approaches is $O(np)$.

Alternatively, the problem of multivariate changepoint detection can be reduced to the known univariate case by projecting a multivariate series to one dimension. Wang and Samworth (2017) show that there is an ideal projection direction, given by the oracle direction $\theta = \mu_B - \mu_A$, which if used for projecting the data will involve no loss of information about the changepoint location. In particular, projecting the data X along $v := \theta$ maximises the signal-to-noise ratio at ζ , $v^\top X_\zeta \sim N(v^\top \mu^{(\zeta)}, \sigma_\zeta^2)$. Since μ_A and μ_B (and thus v) are unknown, the goal in this setting is to estimate v .

Assuming a changepoint occurs at t , a natural estimate $\hat{v}$ of v is given by $\hat{v}_i \propto \bar{X}_{i,1:t} - \bar{X}_{i,(t+1):n}$ (where we assume $\hat{v}$ to be normalised to have Euclidean norm one), the difference of sample means before and after t , where $\bar{X}_{i,1:t}$ is the mean of the first t observations of series i and $\bar{X}_{i,(t+1):n}$ is the mean of series i from time $t + 1$ to n . Projecting the data using $\hat{v}$ leads to the univariate series $\hat{v}^\top X$, for which the square of the cusum statistic is proportional to the sum of the square of the cusum statistics of each individual series. This is just the test statistic one obtains from a likelihood-ratio test for a change in mean in all series for normal data (James et al. 1985; Srivastava and Worsley 1986); and like sum-cusum this works well if all series change, but loses power when only a small proportion of series change (Enikeeva and Harchaoui 2019). For these sparse settings, the estimate, $\hat{v}$, of the projection direction can be improved if additional assumptions on the type of change are made. This is the basis of the approaches described in the next sections. Specifically, in Sect. 2.2, we briefly summarise the Inspect algorithm proposed by Wang and Samworth (2017), while in Sect. 2.3 we introduce a new, Bayesian approach to estimating $\hat{v}$.

2.2 The Inspect algorithm

The Inspect algorithm, proposed by Wang and Samworth (2017), estimates the projection direction by maximising the signal

$$\hat{v}_{\max,k} \in \arg \max_{\hat{v} \in \mathbb{S}^{p-1}(k)} \|T(X)^\top \hat{v}\|_2, \quad (2)$$

where $T(\cdot)$ is the row-wise cusum transform given in (1) and $\mathbb{S}^{p-1}(k)$ is the unit ball in p dimensions with at most k nonzero components. However, optimising over all (exponentially many) k -sparse vectors in $\mathbb{S}^{p-1}(k)$ is computationally very expensive.

To mitigate this, Wang and Samworth (2017) reformulate (2) as the convex relaxation $\max_{M \in \mathcal{M}} \langle M, T \rangle$, where $\mathcal{M} :=$

$\{M \in \mathbb{R}^{p \times (n-1)} : \|M\|_* = 1, \text{rank}(M) = 1, \text{nz}(M) \leq k\}$, $\text{nz}(M)$ is the number of nonzero rows of M , and $\|\cdot\|_*$ is the nuclear norm of a matrix. The definition of $\mathcal{M}$ as a subspace of $\mathbb{R}^{p \times (n-1)}$ stems from the fact that the Euclidean norm in (2) can be rewritten as $\hat{v}^\top T(X) \hat{w} = \langle \hat{v} \hat{w}^\top, T(X) \rangle$ with $v \in \mathbb{S}^{p-1}$ and $w \in \mathbb{S}^{n-2}$ [Wang and Samworth 2017, Eq. (12)]. This can be relaxed further to

$$\hat{M} \in \arg \max_{M \in S_1} \{\langle T, M \rangle - \lambda \|M\|_1\}, \quad (3)$$

where $S_1 := \{M \in \mathbb{R}^{p \times (n-1)} : \|M\|_* \leq 1\}$. Here, $\lambda > 0$ is a tuning parameter controlling the sparseness of the solution. The problem in (3) can be solved with the alternating direction of multipliers algorithm (ADMM), see Gabay and Mercier (1976) and Boyd et al. (2011). Furthermore, Wang and Samworth (2017) establish that (3) can be further relaxed by replacing the set S_1 by $S_2 := \{M \in \mathbb{R}^{p \times (n-1)} : \|M\|_2 \leq 1\}$. In particular, they show that this allows for the closed-form solution

$$\tilde{M} = \frac{\text{soft}(T, \lambda)}{\|\text{soft}(T, \lambda)\|_2}, \quad (4)$$

where $(\text{soft}(A, \lambda))_{ij} = \text{sgn}(A_{ij}) \max\{|A_{ij}| - \lambda, 0\}$ is a soft-thresholding operation for a matrix $A = (A_{ij})$ and a threshold $\lambda \geq 0$.

Finally, the estimate of the projection direction, $\hat{v}$, is obtained as the leading left singular vector of $\hat{M} \hat{M}^\top$ or $\tilde{M} \tilde{M}^\top$, respectively, which is of dimension $p \times p$. This relies on an SVD computation with effort $O(p^3)$ (Golub and van Loan 2012). Using $\hat{v}$, X is projected to a univariate series $\hat{v}^\top X$, to which the cusum transform $T(\cdot)$ is applied. For a given threshold $\tau > 0$ and $t_0 = \arg \max T(\hat{v}^\top X)$, a changepoint is declared at t_0 if $\max T(\hat{v}^\top X) > \tau$. Section 3.1 discusses the choice of τ . The aforementioned approach can be combined with wild binary segmentation of Fryzlewicz (2014) to detect multiple changepoints.

Computing an estimate of M via soft-thresholding is considerably faster than computing (3) via ADMM, and thus Inspect will be employed with soft-thresholding in the remainder of the article. The soft-thresholding operation, as well as the cusum transform, requires an effort of $O(pn)$. The SVD computation for $\hat{M} \hat{M}^\top$ or $\tilde{M} \tilde{M}^\top$ (each matrix multiplication takes $O(p^2n)$ time) has effort $O(p^3)$, thus leading to an overall linear runtime dependence of Inspect on n and a cubic dependence on p . As an alternative to the standard SVD employed in the implementation of Wang and Samworth (2016), the leading left singular vector can also be efficiently computed via partial SVD algorithms (Lanczos iterations). We explore this idea in the supplementary material and show that empirically, Inspect with partial SVD has a slightly improved asymptotic runtime in p .

2.3 A Bayesian approach

While the work of Wang and Samworth (2017) is very powerful, in practice it is still computationally intensive. In particular, while soft-thresholding helps reduce the computational cost, computing an estimate of the projection direction in the Inspect algorithm still relies on the computation of a SVD. Thus, ideally, a faster way of estimating a (good) projection direction would be desirable.

We begin by recalling that Inspect is based on certain sparsity assumptions (Wang and Samworth 2017, Sect. 3) which lead to relaxation (3) to detect sparse changes. Other alternative assumptions on the change in mean are also possible. For example, one possible framework is to adopt a Bayesian setting in which a prior is specified for the change in mean of each series. After calculating the posterior, the projection direction is estimated proportionally to the posterior change in mean of each series. This framework permits the straightforward incorporation of different assumptions for the changepoint scenario via the selection of an appropriate prior.

A Normal prior on the magnitude of change in mean in each series is equivalent to a Chi-squared test statistic (for the changepoint location) and results in a simple estimate which is proportional to the sample change in mean of each series. However, to incorporate that a change often only occurs in a subset of the p series, we adopt a mixture prior—the so-called spike and slab prior—on the change in each individual series, consisting of a Normal prior and a point mass at zero for those series without a change. In particular, the mixture prior on each $\theta_i = \mu_{A,i} - \mu_{B,i}$ proposed in this article is given by $\theta_i \sim (1 - \bar{\pi})\delta_0 + \bar{\pi}N(0, \omega^2)$, where $\bar{\pi} \in [0, 1]$ is the proportion of series which change, ω^2 is the variance of the assumed size of a change, and δ_0 is the point mass at zero.

Under the assumption that a changepoint occurs at time $t \in \{1, \dots, n\}$, for series i , let $D_i^{(t)} = \bar{X}_{i,1:t} - \bar{X}_{i,(t+1):n}$ be the difference of the means before and after the changepoint, and let $\sigma_t = \sqrt{\frac{1}{t} + \frac{1}{n-t}}$. Using the prior on θ_i , an estimate of v_i for $i \in \{1, \dots, p\}$ is given by

$$\hat{v}_i^{(t)} \propto \frac{D_i^{(t)}}{K + \exp\left(\frac{-(D_i^{(t)})^2}{2\sigma_t^2\left(1 + \frac{\sigma_t^2}{\omega^2}\right)}\right)}. \quad (5)$$

Here, $K \in \mathbb{R}$ is a tuning parameter, and we assume that the data have been standardised so that the variance $\kappa = 1$. A derivation of (5) is provided in Appendix A.

The exponential term in (5) depends on the ratio of the prior variance of the size of the change in mean, ω^2 , to the variance of the cusum statistic, σ_t^2 . If prior knowledge

on ω is available, (5) can be used directly in several ways. For instance, ω can be estimated from a set of changepoints detected in the individual series, or expert knowledge of the changepoint scenario can be used. However, we show in the supplementary material that (5) seems to be robust towards an under- or over-specification of the value σ_t^2/ω^2 . In the detectable scenario in which the size of the change is relatively large compared to the noise in the cusum statistic, thus implying $1 + (\sigma_t/\omega)^2 \approx 1$, (5) can be simplified to

$$\hat{v}_i^{(t)} \propto \frac{D_i^{(t)}}{K + \exp\left(\frac{-(D_i^{(t)})^2}{2\sigma_t^2}\right)}. \quad (6)$$

In (6), the exponential term shrinks small values of $D_i^{(t)}$ more towards zero than larger values. The estimate of the projection direction returned by our approach is $\hat{v}^{(t_0)} = (\hat{v}_1^{(t_0)}, \dots, \hat{v}_p^{(t_0)})$ for the choice $t_0 \in \{1, \dots, n\}$ which maximises the cusum statistic at t_0 after projection.

As derived in [Appendix A, Eq. (9)], the tuning parameter K in (6) depends on two quantities:

1. K is proportional to the fraction $\bar{\pi}/(1 - \bar{\pi})$, where $\bar{\pi}$ is the proportion of the p series affected by the change. This could be set using expert information on the changepoint scenario or estimated using individual changepoint detection applied to each series. If it is known that changes are sparse, one obtains $K \propto \bar{\pi}$.
2. K is roughly antiproportional to ω/σ_t , the ratio of the size of a change to the noise in the cusum statistic. This ratio is assumed to be relatively large as already used to simplify (5) into (6).

Section 4 investigates the robustness of the Bayesian approach on the choice of K .

3 Implementation details

This section considers some details required for a practical implementation of the algorithm of Sect. 2.3. In particular, we look at the tuning of the threshold used to declare a changepoint, the number and positions of the timepoints at which the projection direction of (6) is computed, and present our proposed approach as pseudo-code.

3.1 Choice of the threshold

We standardise each series of the input data $X \in \mathbb{R}^{p \times n}$ before applying either Inspect, our Bayesian approach, or sum/max-cusum. For this, we replace each series with its differencing

series which we normalise using the median absolute deviation (Rousseeuw and Croux 1993).

The standardisation of the variance allows us to tune the threshold of the four aforementioned changepoint methods independently of the input data by applying it to simulated standard Normal data of equal dimensions $p \times n$, but without a changepoint. The returned projection direction is employed to project the data, and the maximum cusum value of the univariate dataset after projection is recorded. This process is repeated for $r \in \mathbb{N}$ generated datasets. To minimise Monte Carlo effects, τ is chosen as the 95% quantile among the r maximum cusum statistics. In the simulations (Sect. 4), the choice $r = 100$ is used. Our approach differs slightly from the one used in Inspect (Wang and Samworth 2016), where the threshold is chosen as the maximal cusum statistic among r repetitions.

3.2 Choice of the projection timepoints

Naïvely, one would compute projection direction (6) at every possible changepoint location t , project the data and find a changepoint in the projected series. However, computing (6) for every $t \in \{1, \dots, n\}$ is computationally expensive and not necessary. This is due to the fact that projection direction (6) does not change appreciably between two nearby timepoints. Therefore, in order to test for a change at a certain timepoint t , the projection direction from a nearby timepoint can be used.

One possible way of choosing the timepoints t in the grid, which we outline below, ensures that for any location of the true changepoint ζ (Sect. 2.1) and any series i , the cusum statistic at ζ is within a factor $\gamma \in [0, 1]$ of the expectation of $D_i^{(t)}$ for some t . As in Sect. 2.3, for a fixed time series $i \in \{1, \dots, p\}$, denote $\mathbb{E}(X_{i,j}) = \mu_{A,i}$ for $j \leq \zeta$, and $\mathbb{E}(X_{i,j}) = \mu_{B,i}$ otherwise, as well as $\theta_i = \mu_{A,i} - \mu_{B,i}$. Assume we project at a timepoint before the true changepoint, i.e. $t \leq \zeta$. In this case, the expectation of the observations up to t equals the true mean of that segment, but the expectation of the observations after t does not, thus decreasing the value of $D_i^{(t)}$. To be precise, since $\mathbb{E}(\bar{X}_{i,1:t}) = \mu_{A,i}$ and since $\mathbb{E}(\bar{X}_{i,(t+1):n}) = \frac{\zeta-t}{n-t}\mu_{A,i} + \frac{n-\zeta}{n-t}\mu_{B,i}$ one obtains $\mathbb{E}(\bar{X}_{i,1:t} - \bar{X}_{i,(t+1):n}) = \frac{n-\zeta}{n-t}\theta_i$. Similarly, if $t \geq \zeta$ then $\mathbb{E}(\bar{X}_{i,1:t}) = \frac{\zeta}{t}\mu_{A,i} + \frac{t-\zeta}{t}\mu_{B,i}$ and $\mathbb{E}(\bar{X}_{i,(t+1):n}) = \mu_{B,i}$ together yield that $\mathbb{E}(\bar{X}_{i,1:t} - \bar{X}_{i,(t+1):n}) = \frac{\zeta}{t}\theta_i$. This shows that the expected difference in means differs by a factor $\frac{n-\zeta}{n-t}$ or $\frac{\zeta}{t}$ from the true difference in means θ_i , respectively. We aim to bound these two quantities from below. This leads to the following strategy to choose a grid of timepoints for computing the projection direction.

Fix $\gamma \in [0, 1]$. Choose a set of timepoints $T_1 \subseteq \{1, \dots, n\}$ such that for any $\zeta \in \{1, \dots, n-1\}$, there is a $t \in T_1$ satisfying $t \geq \zeta$ and $\frac{\zeta}{t} \geq \gamma$. Similarly, choose $T_2 \subseteq \{1, \dots, n\}$

such that for any $\zeta \in \{1, \dots, n-1\}$, there is a $t \in T_2$ satisfying $t \leq \zeta$ and $\frac{n-\zeta}{n-t} \geq \gamma$. The grid $\mathcal{T} = T_1 \cup T_2$ is used for computing (6). This construction leads to the first timepoint being placed roughly at $1/\gamma$ in T_1 , the next one at roughly $1/\gamma^2$, etc. (analogously for T_2). In total, $\mathcal{T}$ will contain $O(\log_\gamma n)$ timepoints. With this construction, the grid $\mathcal{T}$ becomes denser towards the beginning and the end of the time series. Towards the middle of the time series, fewer grid points are allocated.

This is in line with the intuition that in order to detect a change in the middle of a sequence, where the error on the cusum statistic is less than at the borders, the projection direction need not be estimated as accurately as on the sides. Naturally, the larger the choice of γ , the more timepoints the grid will contain. The choice of the parameter γ is investigated in the supplementary material.

3.3 Pseudo-code of the algorithm

Algorithm 1: BayesProject

input: $X \in \mathbb{R}^{p \times n}$, $\gamma > 0$;

- 1 $c \leftarrow 0 \in \mathbb{R}^n$;
- 2 Normalise X and calibrate τ (Sect. 3.1);
- 3 Choose K (Sect. 2.3) and $\mathcal{T}$ for the choice of γ (Sect. 3.2);
- 4 **for** $t \in \mathcal{T}$ **do**
- 5 Compute $\hat{v}^{(t)}$ according to (6);
- 6 $\hat{v}^{(t)} \leftarrow \hat{v}^{(t)} / \|\hat{v}^{(t)}\|_2$;
- 7 $x \leftarrow (\hat{v}^{(t)})^\top X$;
- 8 $c_t \leftarrow \left| \frac{1}{t} \sum_{i=1}^t x_i - \frac{1}{n-t} \sum_{i=t+1}^n x_i \right| / \sqrt{\frac{1}{t} + \frac{1}{n-t}}$;
- 9 **end**
- 10 $t_0 \leftarrow \arg \max_{t \in \mathcal{T}} c_t$;
- 11 **return** $\hat{v}^{(t_0)}$;

Pseudo-code of our approach (called *BayesProject*) is given in Algorithm 1. The algorithm works on a multivariate input $X \in \mathbb{R}^{p \times n}$. First, X is normalised row-wise using the *median absolute deviation (MAD)* estimator of the variance and the threshold τ is calibrated (Sect. 3.1). The set $\mathcal{T}$ of projection timepoints is computed as in Sect. 3.2 for a pre-specified parameter γ chosen by the user. The parameter K is chosen as in Sect. 2.3.

For any $t \in \mathcal{T}$, an estimate, $\hat{v}^{(t)}$, is computed using (6). The resulting vector is normed, and the data are projected to a univariate series $(v^{(t)})^\top X$, for which the standard cusum statistic is then computed at t and stored in c_t . *BayesProject* returns the projection direction $\hat{v}^{(t_0)}$ for the timepoint $t_0 \in \mathcal{T}$ leading to the largest absolute cusum statistic at the point of projection.

Although the projection direction that *BayesProject* returns is selected from a finite set of timepoints $\mathcal{T}$, *BayesProject* is not restricted to a detection of a changepoint at the loca-

tions in $\mathcal{T}$ only. Instead, the projection direction $\hat{v}^{(t_0)}$ is used to project the data onto a univariate time series in which a changepoint is declared anywhere within $\{1, \dots, n\}$ if the maximum of the cusum transform applied to the projected series satisfies $\max T\left(\left(\hat{v}^{(t_0)}\right)^\top X\right) > \tau$. The changepoint location is accordingly given by $\arg \max T\left(\left(\hat{v}^{(t_0)}\right)^\top X\right)$. This approach is readily extendable to the multiple changepoint setting: we simply combine BayesProject with wild binary segmentation proposed by Fryzlewicz (2014) to detect several changepoints.

For any $t \in \mathcal{T}$, computing the projection direction $\hat{v}^{(t)}$ according to (6) requires effort $O(np)$, and likewise projecting to a univariate series requires effort $O(np)$. Together, the effort of BayesProject is given by $O(np|\mathcal{T}|) = O(np \log_\gamma n)$ when using the grid of Sect. 3.2.

4 Simulation studies

The article presents five simulation scenarios to assess the dependence of the four algorithms under consideration (Inspect with soft-thresholding of Sect. 2.2, BayesProject of Sect. 3.3, as well as sum-cusum and max-cusum of Sect. 2.1) on the following:

1. the dependence on the series length n ,
2. the number of series p ,
3. the proportion d/p of series exhibiting a change, where $d < p$ is the number of series which change,
4. the size s of the change,
5. and the location of the changepoint.

The simulation scenarios are chosen to better understand the scaling behaviour of all four algorithms both as the series length and the number of series increase; moreover, we are interested in their ability to detect a change in challenging situations, for instance if only a small proportion of series has a change, the change size is small, or the change occurs at the beginning or end of the series. The supplementary material additionally presents experiments on univariate changepoint detection and the null case in which no changepoint is present.

In this section, all experiments use simulated standard Normal data with a single changepoint. Apart from the investigation of the dependence on the location of the changepoint, the changepoints are always drawn uniformly among $\{1, \dots, n\}$ in each simulation. Each experiment is considered in three settings: in a sparse ($d/p = 0.05$), moderate ($d/p = 0.2$) and dense ($d/p = 0.5$) scenario. The size of the change was set to $s = 0.04$. All results are averages over 1000 repetitions.

The quality of the estimated changepoint is evaluated with two measures: first, results show the proportion among all repetitions in which a changepoint was found (denoted as proportion of estimated changepoints in the figures). Second, as a measure of accuracy of the position of the estimated changepoint, the average distance to the true changepoint location is reported. Additionally, the methods are evaluated with respect to the accuracy of the estimated projection direction, measured as the L_2 difference between the (normalised) estimated and ideal projection directions. Lastly, all algorithms are compared with respect to their runtimes in seconds. Empirical runtimes are computed with the aim to verify the theoretical runtimes in practice.

Other practical considerations The threshold for each method is computed as described in Sect. 3.1 for every new pair of parameters (n, p) . All simulation results for BayesProject in this and the following sections are computed with the fixed choice $K = 1$ (corresponding to a scenario in which half of all series are assumed to exhibit a change and the size of each change is of the same size as the variance of the data, see Sect. 2.3). The grid of projection timepoints for BayesProject is computed as in Sect. 3.2 with $\gamma = 0.6$. This is an arbitrary choice. Simulations in the supplementary material indicate that the estimated projection direction (and thus the found changepoint by BayesProject) does not change appreciably as long as the grid of projection timepoints is not too sparse (corresponding to low values of γ). That is, empirically, for roughly the range $\gamma \in [0.5, 1]$, the accuracy of BayesProject does not considerably change, though the method becomes slower as γ increases (due to the increasing number of computed projection directions as well as the resulting projected series scanned for a changepoint).

Dependence on p . Figure 1 shows the dependence of four methods (BayesProject, Inspect, sum-cusum and max-cusum) on the number of series p , while $n = 10,000$ is fixed. The rows display the proportion of estimated changes (first row), the accuracy of the estimated changepoint location measured via average distance to the true changepoint location (second row), the L_2 difference between ideal and estimated projection directions (third row) as well as the runtime (fourth row). Columns show a sparse (left), moderate (middle) and dense (right) scenario.

Figure 1 shows that BayesProject and sum-cusum perform similarly in terms of proportion of estimated changes and average distance to the true changepoint location, followed by Inspect. Max-cusum is noticeably worse. Except for the sparse case, Inspect, BayesProject and sum-cusum do not show a strong dependence of the accuracy of the estimated changepoint on p . Both BayesProject and sum-cusum most accurately estimate the projection direction.

The empirical runtime analysis shows a superlinear runtime dependence on p for Inspect, while BayesProject as well as sum-cusum and max-cusum show a linear runtime

Fig. 1 Dependence of BayesProject (black crosses), Inspect (red circles), sum-cusum (green squares) and max-cusum (blue triangles) on the number of series p while $n = 10,000$. Rows show proportion of estimated changes (first row), average distance to the true changepoint location (second row), L_2 distance between the estimated and ideal pro-

jection directions (third row) and runtime (fourth row). Columns show sparse (left), moderate (middle) and dense (right) scenarios. Log–log plots to assess runtime scalings in the fourth row. Slope estimates for the moderate scenario (middle column) are 1.23 (Inspect), 1.00 (BayesProject), 1.03 (sum-cusum) and 1.03 (max-cusum). (Color figure online)

dependence. Additional experiments for $n = 100$ and $p > n$ included in the supplementary material further investigate the asymptotic runtime in p .

Qualitatively similar results are seen as we vary the number of series n , the dependence on the proportion d/p of series with a change, the size of the change s and the change-point location—see the supplementary material for details. These simulations show max-cusum consistently performing worse than the other methods. While overall performance of the other methods is similar, Inspect is most accurate for sparse scenarios, whereas sum-cusum and BayesProject are more accurate for the moderate and denser scenarios.

Overall, we find that BayesProject yields projections close to the oracle projection direction and detects changepoints with a consistently high accuracy across all scenarios considered while exhibiting a linear runtime in p .

Choice of K . The qualitative dependence of the tuning parameter K is investigated in a detailed simulation study in the supplementary material, where parts of the simulation study reported in this section are repeated for the optimal choice of K (using oracle knowledge of the changepoint scenario), as well as an underestimate $K/2$ and an overestimate $2K$. The supplementary material shows that the performance of BayesProject is largely identical for the three choices of K . For instance, Fig. 2 shows the dependence of BayesProject on the number of series p for the three choices of K in the same scenario as the one used for Fig. 1. As can be seen, the projection direction estimate is slightly better for $K/2$ or $2K$ in some scenarios, however at the expense of performing worse in the other scenarios. The supplementary material shows that the estimation accuracy can be slightly improved if prior knowledge on the proportion of series with a change, or the size of the change is available.

5 Multiple changepoint detection in simulated data

We now turn to consider the performance of various approaches for the multiple changepoint detection scenario. We replicate the simulation setting of (Wang and Samworth 2017, Sect. 5.3) which considers Normal distributed data of dimensions $n = 2000$ and $p = 200$, with changepoints at the locations $\zeta = (500, 1000, 1500)$ and $\kappa^2 = 1$. At the three changepoints, the signal changes in $k = 40$ series by varying amounts, but with an aggregated signal size for the three changes of $(\vartheta^{(1)}, \vartheta^{(2)}, \vartheta^{(3)}) = (\vartheta, 2\vartheta, 3\vartheta)$, where $\vartheta \in \{0.4, 0.6\}$. Three scenarios are considered: (i) the *complete overlap* case, where the same k series change at each of the three changepoints; (ii) the *half overlap* case, where changes occur in the series $\frac{i-1}{2}k + 1, \dots, \frac{i+1}{2}k$ for $i \in \{1, 2, 3\}$; and (iii) the *no overlap* case, where changes occur at disjoint sets of indices. As in Wang and Samworth

(2017), all results provided in this section are based on 100 repetitions.

Additionally, we include the five algorithms described in Sect. 1 in our study. Those are the Double Cusum algorithm (Cho 2016), Sparsified Binary Segmentation (SBS) of Cho and Fryzlewicz (2015), as well as the approaches of Enikeeva and Harchaoui (2019), Yu and Chen (2020) and Grundy et al. (2020b). For SBS, the parameter for detecting a change was set to $\sqrt{\log(p/0.05)}$. The method of Enikeeva and Harchaoui (2019) was run with significance level 0.05. The method of Yu and Chen (2020) used an exclusion boundary of $\underline{s} = \lfloor n^{0.1} \rfloor$ as proposed by the authors, 100 bootstrap repetitions, and the threshold was computed as the 0.95 quantile of the bootstrap samples. The method of Grundy et al. (2020b) is provided in the R-package *changepoint.geo* on CRAN (Grundy et al. 2020a).

All algorithms are combined with binary segmentation in order to find multiple changepoints, apart from Grundy et al. (2020b) since the *changepoint.geo* package is already able to detect multiple changes. Before running BayesProject, its threshold is calibrated as described in Sect. 3.1. The tuning parameter K is set to $K = 0.1$. The grid of timepoints is chosen as in Sect. 3.2 with $\gamma = 0.6$.

Table 1 shows simulation results. To compare the four algorithms with respect to their accuracy in multiple changepoint detection, the v -measure is employed (Rosenberg and Hirschberg 2007), a widely used measure in changepoint analysis (Eriksson and Olofsson 2019; Ludkin et al. 2018; Li and Munk 2016; Frick et al. 2014). The measure is normed to the interval $[0, 1]$, with zero being the worst score and one indicating a perfect match. Within Table 1, the complete, half and no overlap scenarios are given as the first, second and third block of the table, with nine rows each for the two choices of ϑ determining the size of the change at each changepoint. The table shows frequency counts for the number of estimated changes (between zero and five or more changepoints), the average v -measure score, as well as the number of true positives and false positives (defined as being at most a distance 15 away from any true changepoint).

As shown in Table 1, BayesProject performs slightly worse than Inspect. Of the other approaches, the method of Enikeeva and Harchaoui (2019) performs best across all scenarios when assessed with the v -measure metric. The double cusum and SBS algorithms perform comparably when assessed with the v -measure metric, though SBS has a higher false-positive rate. The approaches of Yu and Chen (2020) and Grundy et al. (2020b) are competitive for $\vartheta = 0.6$ but lose power for $\vartheta = 0.4$. Since Grundy et al. (2020b) detects changes in both mean and variance, a loss in power compared to the other algorithms is expected.

We expect BayesProject to perform well in scenarios with uneven segment lengths, since the projections computed along the gridpoints introduced in Sect. 3.2 allow BayesPro-

Table 1 Parameters $n = 2000$, $p = 200$, $k = 40$, $\zeta = (500, 1000, 1500)$

$(\vartheta^{(1)}, \vartheta^{(2)}, \vartheta^{(3)})$	Method	Frequency of estimated cpts						v -measure	True pos.	False pos.
		0	1	2	3	4	5+			
(0.6, 1.2, 1.8)	BayesProject	0	0	63	36	1	0	0.86	1.90	0.48
	Inspect	0	0	33	65	2	0	0.92	2.34	0.35
	Sum-cusum	0	0	43	51	5	1	0.86	1.73	0.91
	Max-cusum	0	0	41	53	6	0	0.88	1.90	0.75
	Cho (2016)	0	0	42	51	7	0	0.87	1.88	0.77
	Cho and Fryzlewicz (2015)	0	0	29	60	10	1	0.89	1.83	1.00
	Enikeeva and Harchaoui (2019)	0	0	12	73	14	1	0.93	2.29	0.75
	Yu and Chen (2020)	0	0	96	4	0	0	0.82	1.60	0.44
	Grundy et al. (2020b)	0	0	97	3	0	0	0.82	1.54	0.49
(0.4, 0.8, 1.2)	BayesProject	0	0	92	8	0	0	0.81	1.43	0.65
	Inspect	0	0	80	17	3	0	0.84	1.77	0.46
	Sum-cusum	0	0	77	17	6	0	0.80	1.20	1.09
	Max-cusum	0	0	87	13	0	0	0.80	1.24	0.89
	Cho (2016)	0	0	85	15	0	0	0.80	1.27	0.88
	Cho and Fryzlewicz (2015)	0	0	72	25	3	0	0.80	1.21	1.10
	Enikeeva and Harchaoui (2019)	0	0	72	24	4	0	0.84	1.66	0.66
	Yu and Chen (2020)	0	9	91	0	0	0	0.77	1.16	0.75
	Grundy et al. (2020b)	0	34	64	2	0	0	0.70	0.86	0.82
(0.6, 1.2, 1.8)	BayesProject	0	0	63	36	1	0	0.86	1.77	0.61
	Inspect	0	0	31	67	2	0	0.92	2.23	0.48
	Sum-cusum	0	0	48	45	7	0	0.86	1.62	0.97
	Max-cusum	0	0	44	55	1	0	0.89	1.91	0.66
	Cho (2016)	0	0	46	53	1	0	0.89	1.89	0.66
	Cho and Fryzlewicz (2015)	0	0	29	54	17	0	0.88	1.74	1.14
	Enikeeva and Harchaoui (2019)	0	0	17	71	12	0	0.94	2.37	0.58
	Yu and Chen (2020)	0	0	98	2	0	0	0.83	1.69	0.33
	Grundy et al. (2020b)	0	0	99	1	0	0	0.82	1.55	0.46
(0.4, 0.8, 1.2)	BayesProject	0	0	74	25	1	0	0.80	1.31	0.96
	Inspect	0	0	76	20	3	1	0.83	1.57	0.72
	Sum-cusum	0	0	77	20	3	0	0.80	1.07	1.19
	Max-cusum	0	0	94	6	0	0	0.81	1.38	0.68
	Cho (2016)	0	0	94	6	0	0	0.81	1.41	0.65
	Cho and Fryzlewicz (2015)	0	0	76	22	2	0	0.80	1.15	1.11
	Enikeeva and Harchaoui (2019)	0	0	74	22	4	0	0.84	1.65	0.65
	Yu and Chen (2020)	0	25	75	0	0	0	0.75	1.22	0.53
	Grundy et al. (2020b)	0	33	60	7	0	0	0.71	0.87	0.87
(0.6, 1.2, 1.8)	BayesProject	0	0	59	41	0	0	0.86	1.96	0.45
	Inspect	0	0	31	67	2	0	0.92	2.32	0.39
	Sum-cusum	0	0	42	49	9	0	0.85	1.70	0.97
	Max-cusum	0	0	42	55	3	0	0.89	2.10	0.51
	Cho (2016)	0	0	44	53	3	0	0.89	2.09	0.50
	Cho and Fryzlewicz (2015)	0	0	32	47	21	0	0.88	1.74	1.15
	Enikeeva and Harchaoui (2019)	0	0	17	73	10	0	0.94	2.43	0.50
	Yu and Chen (2020)	0	0	98	2	0	0	0.84	1.87	0.15
	Grundy et al. (2020b)	0	0	97	3	0	0	0.82	1.54	0.49

Table 1 continued

$(\vartheta^{(1)}, \vartheta^{(2)}, \vartheta^{(3)})$	Method	Frequency of estimated cpts						v -measure	True pos.	False pos.
		0	1	2	3	4	5+			
(0.4, 0.8, 1.2)	BayesProject	0	0	82	18	0	0	0.81	1.37	0.81
	Inspect	0	0	83	16	1	0	0.84	1.66	0.52
	Sum-cusum	0	0	78	21	1	0	0.80	1.18	1.05
	Max-cusum	0	0	92	7	1	0	0.82	1.45	0.64
	Cho (2016)	0	0	92	7	1	0	0.82	1.51	0.58
	Cho and Fryzlewicz (2015)	0	0	76	24	0	0	0.80	1.15	1.09
	Enikeeva and Harchaoui (2019)	0	0	78	20	2	0	0.84	1.77	0.47
	Yu and Chen (2020)	0	49	51	0	0	0	0.69	1.19	0.32
	Grundy et al. (2020b)	0	40	56	4	0	0	0.69	0.78	0.86

Results for multiple changepoint detection for the complete (top), half (middle) and no overlap (bottom) scenario

Table 2 Parameters $n = 2000$, $p = 200$, $k = 40$, $\zeta = (200, 400, 1600)$

$(\vartheta^{(1)}, \vartheta^{(2)}, \vartheta^{(3)})$	Method	Frequency of estimated cpts						v -measure	True pos.	False pos.
		0	1	2	3	4	5+			
(2.4, 1.8, 1.2)	BayesProject	0	0	0	98	1	1	0.98	2.86	0.17
	Inspect	0	0	0	95	5	0	0.98	2.91	0.14
	Sum-cusum	0	0	0	75	21	4	0.94	2.44	0.85
	Max-cusum	0	0	0	86	13	1	0.96	2.67	0.48
	Cho (2016)	0	0	0	85	14	1	0.96	2.70	0.47
	Cho and Fryzlewicz (2015)	0	0	0	85	14	1	0.95	2.51	0.65
	Enikeeva and Harchaoui (2019)	0	0	0	83	15	2	0.97	2.86	0.33
	Yu and Chen (2020)	0	0	0	99	1	0	0.96	2.60	0.41
	Grundy et al. (2020b)	0	0	0	97	3	0	0.95	2.43	0.60
(2.4, 1.8, 1.2)	BayesProject	0	0	0	84	16	0	0.98	2.87	0.29
	Inspect	0	0	0	94	5	1	0.98	2.96	0.11
	Sum-cusum	0	0	0	79	21	0	0.94	2.42	0.79
	Max-cusum	0	0	0	92	8	0	0.97	2.75	0.33
	Cho (2016)	0	0	0	91	9	0	0.97	2.72	0.37
	Cho and Fryzlewicz (2015)	0	0	0	76	22	2	0.95	2.47	0.79
	Enikeeva and Harchaoui (2019)	0	0	0	84	16	0	0.97	2.89	0.27
	Yu and Chen (2020)	0	0	1	99	0	0	0.97	2.72	0.27
	Grundy et al. (2020b)	0	0	3	93	4	0	0.93	2.31	0.70
(2.4, 1.8, 1.2)	BayesProject	0	0	0	83	16	1	0.98	2.91	0.27
	Inspect	0	0	0	93	6	1	0.98	2.88	0.20
	Sum-cusum	0	0	0	73	25	2	0.94	2.40	0.89
	Max-cusum	0	0	0	94	6	0	0.97	2.83	0.23
	Cho (2016)	0	0	0	95	5	0	0.97	2.82	0.23
	Cho and Fryzlewicz (2015)	0	0	0	66	32	2	0.95	2.58	0.78
	Enikeeva and Harchaoui (2019)	0	0	0	85	15	0	0.98	2.96	0.19
	Yu and Chen (2020)	0	0	0	100	0	0	0.97	2.82	0.18
	Grundy et al. (2020b)	0	0	3	92	4	1	0.93	2.29	0.74

Results for multiple changepoint detection for the complete (top), half (middle) and no overlap (bottom) scenario

Fig. 2 Dependence of BayesProject on the number of series p while $n = 10,000$. Rows show average distance to the true changepoint location (first row) and L_2 distance between the estimated and ideal projection directions (second row). Columns show sparse (left), mod-

erate (middle) and dense (right) scenarios. BayesProject with optimal K ($\times$), with $K/2$ ($+$) and $2K$ ($*$). The depicted measurements are virtually indistinguishable in the top left and top middle panels

ject to have comparatively less power for detecting changes in the middle of a sequence while having an advantage for detecting changes at the start and end. Table 2 repeats the previous experiment for the changepoints $\zeta = (200, 400, 1600)$ and $(\vartheta^{(1)}, \vartheta^{(2)}, \vartheta^{(3)}) = (2.4, 1.8, 1.2)$. We observe that BayesProject performs very competitively and mildly outperforms both Inspect and Enikeeva and Harchaoui (2019) when assessed with the v -measure metric. However, Inspect and Enikeeva and Harchaoui (2019) incur a lower false-positive rate for the no overlap scenario.

A scenario with two changepoints at the sides ($n = 1000$, $p = 100$ with a change in $k = 30$ series, changepoints $\zeta = (100, 800)$ with magnitudes $(\vartheta^{(1)}, \vartheta^{(2)}) = (1.5, 1.0)$) is presented in Table 3. Here, BayesProject performs better than or equal to all other algorithms throughout all three scenarios when assessed with the v -measure while having a competitive true-positive rate and a low false-positive rate.

We also look at the accuracy by means of density plots for the estimated changepoint locations. For improved readability of the density plots, we select the five best methods based on the previous analyses. Those are: BayesProject, Inspect, as well as the methods of Cho (2016), Enikeeva and Harchaoui

(2019) and Yu and Chen (2020). Since the empirical densities have sharp peaks, we employ the (smoothed) log-concave maximum likelihood estimator of Rufibach and Dümbgen (2009) which is free of tuning parameters, thus not requiring the selection of a bandwidth which could bias the visualisation. An implementation is provided in the R-package *logcondens* on CRAN (Rufibach and Dümbgen 2016).

Figure 3 shows density plots for the complete overlap scenario of Table 2. We observe that BayesProject detects the first changepoint with the highest accuracy, while the other methods are less accurate. BayesProject and Inspect are equally competitive for the second changepoint, while Inspect has the highest accuracy in detecting the third changepoint, followed by BayesProject and Enikeeva and Harchaoui (2019). Figure 4 shows similar density plots for the detection of the two changepoints in Table 3 in the no overlap scenario. We observe that for both changepoints, BayesProject achieves a high accuracy.

Density plots for all the scenarios displayed in Tables 1, 2 and 3 are included in the supplementary material and show a competitive performance of BayesProject in comparison with the other algorithms.

Table 3 Parameters $n = 1000$, $p = 100$, $k = 30$, $\zeta = (100, 800)$

$(\vartheta^{(1)}, \vartheta^{(2)})$	Method	Frequency of estimated cpts						v -measure	True pos.	False pos.
		0	1	2	3	4	5+			
(1.5, 1.0)	BayesProject	0	0	97	3	0	0	0.95	1.89	0.14
	Inspect	0	0	95	5	0	0	0.95	1.90	0.15
	Sum-cusum	0	2	98	0	0	0	0.91	1.62	0.36
	Max-cusum	0	1	95	4	0	0	0.90	1.70	0.33
	Cho (2016)	0	6	93	1	0	0	0.88	1.63	0.32
	Cho and Fryzlewicz (2015)	0	0	97	3	0	0	0.91	1.61	0.42
	Enikeeva and Harchaoui (2019)	0	0	81	17	0	2	0.95	1.90	0.33
	Yu and Chen (2020)	6	84	10	0	0	0	0.59	0.92	0.12
	Grundy et al. (2020b)	0	50	46	4	0	0	0.75	1.21	0.33
(1.5, 1.0)	BayesProject	0	0	97	3	0	0	0.95	1.91	0.12
	Inspect	0	1	95	4	0	0	0.95	1.86	0.17
	Sum-cusum	0	1	99	0	0	0	0.91	1.63	0.36
	Max-cusum	0	4	91	5	0	0	0.90	1.65	0.36
	Cho (2016)	0	6	94	0	0	0	0.90	1.65	0.29
	Cho and Fryzlewicz (2015)	0	1	95	4	0	0	0.89	1.51	0.52
	Enikeeva and Harchaoui (2019)	0	0	78	21	0	1	0.94	1.85	0.39
	Yu and Chen (2020)	15	74	11	0	0	0	0.53	0.84	0.12
	Grundy et al. (2020b)	0	52	41	7	0	0	0.73	1.16	0.39
(1.5, 1.0)	BayesProject	0	0	97	3	0	0	0.96	1.91	0.12
	Inspect	0	0	90	9	1	0	0.94	1.88	0.23
	Sum-cusum	0	3	96	1	0	0	0.91	1.61	0.37
	Max-cusum	0	3	90	7	0	0	0.90	1.69	0.35
	Cho (2016)	0	3	97	0	0	0	0.90	1.66	0.31
	Cho and Fryzlewicz (2015)	0	1	94	5	0	0	0.89	1.44	0.60
	Enikeeva and Harchaoui (2019)	0	0	82	16	1	1	0.94	1.87	0.34
	Yu and Chen (2020)	32	52	16	0	0	0	0.45	0.77	0.07
	Grundy et al. (2020b)	0	48	49	3	0	0	0.74	1.17	0.38

Results for multiple changepoint detection for the complete (top), half (middle) and no overlap (bottom) scenario

Fig. 3 Estimated densities of the location of changepoint estimates reported in Table 2 for the complete overlap scenario. BayesProject (black, solid line), Inspect (red, dashed line), Cho (2016) (orange, dot-

ted line), Enikeeva and Harchaoui (2019) (brown, dot dashed line), Yu and Chen (2020) (turquoise, long dashed line). Changepoints 200 (left), 400 (middle) and 1600 (right). (Color figure online)

Fig. 4 Estimated densities of the location of changepoint estimates reported in Table 3 for the no overlap scenario. BayesProject (black, solid line), Inspect (red, dashed line), Cho (2016) (orange, dotted line),

Enikeeva and Harchaoui (2019) (brown, dot dashed line), Yu and Chen (2020) (turquoise, long dashed line). Changepoints 100 (left) and 800 (right). (Color figure online)

6 Discussion

This article introduces BayesProject, a Bayesian approach to estimate the projection direction for multivariate changepoint detection. The proposed approach addresses changepoint detection in scenarios with a large number of series p and provides a linear runtime in the number of series and a super-linear runtime in the series length. Simulations indicate that BayesProject is robust, yields projections close to the oracle projection direction and, moreover, that its accuracy in detecting changepoints is comparable to existing algorithms. While we have focussed just on the change in mean in Gaussian data, with one specific form of prior for the change in mean, the idea can be applied more widely to other models and prior distributions. However, the use of conjugate priors would be necessary for a computationally efficient method that allows for calculating the projection analytically.

Acknowledgements The authors would like to thank Tengyao Wang for his help with the replication of Table 1 in simulation section. They also gratefully acknowledge the financial support of the Engineering and Physical Sciences Research Council (EP/N031938/1).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

A Posterior distribution

Assume the changepoint $t = \zeta$ is known and fix the index $i \in \{1, \dots, p\}$ of a series. Recall that $X_{i,j} \sim N(\mu_{A,i}, \kappa^2)$ for $j \leq t$ and some variance κ and $X_{i,j} \sim N(\mu_{B,i}, \kappa^2)$ otherwise. The distribution of $\bar{X}_{i,1:t}$ is $N(\mu_{A,i}, \kappa^2/t)$, and the distribution of $\bar{X}_{i,(t+1):n}$ is $N(\mu_{B,i}, \kappa^2/(n-t))$. Let $D = \bar{X}_{i,1:t} - \bar{X}_{i,(t+1):n}$ and $\theta = \theta_i = \mu_{A,i} - \mu_{B,i}$, where the dependence of D and θ on t and i is omitted for clarity. This yields $D \sim N(\theta, \kappa^2\sigma^2)$, where the dependence of $\sigma = \sqrt{\frac{1}{t} + \frac{1}{n-t}}$ on t is omitted in this section.

Using the prior on θ , given by $\theta \sim (1 - \bar{\pi})\delta_0 + \bar{\pi}N(0, \omega^2)$ with $\bar{\pi} \in [0, 1]$ (the proportion of series with a change) and point mass at zero δ_0 ,

$$\begin{aligned} \mathbb{P}(\theta|D) &\propto \mathbb{P}(D|\theta)\mathbb{P}(\theta) \\ &\propto \exp\left(\frac{-(D-\theta)^2}{2\kappa^2\sigma^2}\right) \left[(1-\bar{\pi})\delta_0 + \bar{\pi}\frac{1}{\sqrt{2\pi\omega^2}}\exp\left(\frac{-\theta^2}{2\omega^2}\right)\right] \\ &\propto (1-\bar{\pi})\delta_0 + \bar{\pi}\frac{1}{\sqrt{2\pi\omega^2}}\exp\left(\frac{-\theta^2 + 2\frac{\omega^2 D}{\omega^2 + \kappa^2\sigma^2}\theta}{2\frac{\kappa^2\sigma^2\omega^2}{\omega^2 + \kappa^2\sigma^2}}\right) \\ &\propto (1-\bar{\pi})\delta_0 + \bar{\pi}\frac{1}{\sqrt{2\pi\omega^2}}\exp\left(\frac{-\left(\theta - \frac{\omega^2 D}{\omega^2 + \kappa^2\sigma^2}\right)^2}{2\frac{\kappa^2\sigma^2\omega^2}{\omega^2 + \kappa^2\sigma^2}}\right) \\ &\quad \exp\left(\frac{\omega^2 D^2}{2\kappa^2\sigma^2(\omega^2 + \kappa^2\sigma^2)}\right) \\ &\propto (1-\bar{\pi})\exp\left(\frac{-D^2}{2\kappa^2\sigma^2\left(1 + \frac{\kappa^2\sigma^2}{\omega^2}\right)}\right)\delta_0 \\ &\quad + \bar{\pi}\frac{1}{\sqrt{2\pi\omega^2}}\exp\left(\frac{-\left(\theta - \frac{\omega^2 D}{\omega^2 + \kappa^2\sigma^2}\right)^2}{2\frac{\kappa^2\sigma^2\omega^2}{\omega^2 + \kappa^2\sigma^2}}\right) \end{aligned}$$

$$=: (1 - \bar{\pi})A(D)\delta_0 + \bar{\pi} \frac{1}{\sqrt{2\pi}\omega^2} B(\theta, D). \quad (7)$$

Thus in the case $\theta \neq 0$, the posterior distribution is a normal with mean $\frac{\omega^2 D}{\omega^2 + \kappa^2 \sigma^2}$ and variance $\frac{\kappa^2 \sigma^2 \omega^2}{\omega^2 + \kappa^2 \sigma^2}$. The normalising constant is given by

$$\int \mathbb{P}(\theta|D)d\theta = (1 - \bar{\pi}) \exp\left(\frac{-D^2}{2\kappa^2 \sigma^2 \left(1 + \frac{\kappa^2 \sigma^2}{\omega^2}\right)}\right) + \bar{\pi} \sqrt{\frac{\kappa^2 \sigma^2}{\omega^2 + \kappa^2 \sigma^2}}.$$

The posterior mean for a change (the case $\theta \neq 0$) is given by $\frac{\omega^2 D}{\omega^2 + \kappa^2 \sigma^2}$ multiplied by the probability $\mathbb{P}(\theta \neq 0|D)$ in (7),

$$\begin{aligned} & \frac{\omega^2 D}{\omega^2 + \kappa^2 \sigma^2} \cdot \mathbb{P}(\theta \neq 0|D) \\ &= \frac{\omega^2 D}{\omega^2 + \kappa^2 \sigma^2} \cdot \frac{\bar{\pi} \frac{1}{\sqrt{2\pi}\omega^2} \int_{\theta \neq 0} B(\theta, D)d\theta}{(1 - \bar{\pi}) \exp\left(\frac{-D^2}{2\kappa^2 \sigma^2 \left(1 + \frac{\kappa^2 \sigma^2}{\omega^2}\right)}\right) + \bar{\pi} \sqrt{\frac{\kappa^2 \sigma^2}{\omega^2 + \kappa^2 \sigma^2}}} \\ &\propto \frac{D}{\exp\left(\frac{-D^2}{2\kappa^2 \sigma^2 \left(1 + \frac{\kappa^2 \sigma^2}{\omega^2}\right)}\right) + \frac{\bar{\pi}}{1 - \bar{\pi}} \sqrt{\frac{\kappa^2 \sigma^2}{\omega^2 + \kappa^2 \sigma^2}}}, \end{aligned} \quad (8)$$

where it suffices to omit prefactors since the vector of posterior means for all series can be assumed normalised. As can be seen from (8), the parameter K of (6) corresponds to

$$K \propto \frac{\bar{\pi}}{1 - \bar{\pi}} \sqrt{\frac{\kappa^2 \sigma^2}{\omega^2 + \kappa^2 \sigma^2}}. \quad (9)$$

References

- Aue, A., Hörmann, S., Horváth, L., Reimherr, M.: Break detection in the covariance structure of multivariate time series models. *Ann. Stat.* **37**(6B), 4046–4087 (2009)
- Bai, J., Perron, P.: Estimating and testing linear models with multiple structural changes. *Econometrica* **66**(1), 47–78 (1998)
- Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found Trends Mach. Learn.* **3**, 1–122 (2011)
- Cho, H.: Change-point detection in panel data via double CUSUM statistic. *Electron. J. Stat.* **10**, 2000–2038 (2016)
- Cho, H., Fryzlewicz, P.: Multiple-change-point detection for high dimensional time series via sparsified binary segmentation. *J. R. Stat. Soc. B* **77**(2), 475–507 (2015)
- Desobry, F., Davy, M., Doncarli, C.: An online kernel change detection algorithm. *IEEE Trans. Signal Process.* **53**(8), 2961–2974 (2005)
- Enikeeva, F., Harchaoui, Z.: High-dimensional change-point detection under sparse alternatives. *Ann. Stat.* **47**(4), 2051–2079 (2019)
- Eriksson, M., Olofsson, T.: Computationally efficient off-line joint change point detection in multiple time series. *IEEE Trans. Signal Process.* **67**(1), 149–163 (2019)
- Frick, K., Munk, A., Sieling, H.: Multiscale change point inference. *J. R. Stat. Soc. B Stat. Methodol.* **76**(3), 495–580 (2014)
- Fryzlewicz, P.: Wild binary segmentation for multiple change-point detection. *Ann. Stat.* **42**(6), 2243–2281 (2014)
- Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximations. *Comput. Math. Appl.* **2**, 17–40 (1976)
- Golub, G., van Loan, C.: *Matrix Computations*, 4th edn. Johns Hopkins University Press, New York (2012)
- Grundy, T., Killick, R., Mihaylov, G.: *Changeoint.geo: geometrically inspired multivariate change point detection*. R Package Version 1.0.1. <https://cran.r-project.org/package=changeoint.geo> (2020a)
- Grundy, T., Killick, R., Mihaylov, G.: High-dimensional changepoint detection via a geometrically inspired mapping. *Stat. Comput.* (2020b). <https://doi.org/10.1007/s11222-020-09940-y>
- Guédon, Y.: Exploring the latent segmentation space for the assessment of multiple change-point models. *Comput. Stat.* **28**(6), 2641–2678 (2013)
- Haynes, K., Eckley, I., Fearnhead, P.: Computationally efficient change-point detection for a range of penalties. *J. Comput. Graph. Stat.* **26**(1), 134–143 (2017)
- Horváth, L., Rice, G.: Extensions of some classical methods in change point analysis. *Test* **23**, 219–255 (2014)
- James, B., James, K.L., Siegmund, D.: Tests for a change-point. Technical Report No. 35, The Office for Naval Research, pp. 1–30 (1985)
- Lévy-Leduc, C., Roueff, F.: Detection and localization of change-points in high-dimensional network traffic data. *Ann. Appl. Stat.* **3**(2), 637–662 (2009)
- Li, H., Munk, A.: FDR-control in multiscale change-point segmentation. *Electron. J. Stat.* **10**, 918–959 (2016)
- Ludkin, M., Eckley, I., Neal, P.: Dynamic stochastic block models: parameter estimation and detection of changes in community structure. *Stat. Comput.* **28**(6), 1201–1213 (2018)
- Lung-Yut-Fong, A., Lévy-Leduc, C., Cappé, O.: Distributed detection/localization of change-points in high-dimensional network traffic data. *Stat. Comput.* **22**(2), 485–496 (2012)
- Maidstone, R., Hocking, T., Rigall, G., Fearnhead, P.: On optimal multiple changepoint algorithms for large data. *Stat. Comput.* **27**(2), 519–533 (2017)
- Matteson, D., James, N.: A nonparametric approach for multiple change point analysis of multivariate data. *J. Am. Stat. Assoc.* **109**(505), 334–345 (2012)
- Nam, C., Aston, J., Eckley, I., Killick, R.: The uncertainty of storm season changes: quantifying the uncertainty of autocovariance changepoints. *Technometrics* **57**(2), 194–206 (2015)
- Page, E.: Continuous inspection scheme. *Biometrika* **41**(1/2), 110–115 (1954)
- Preuß, P., Puchstein, R., Dette, H.: Detection of multiple structural breaks in multivariate time series. *J. Am. Stat. Assoc.* **110**(510), 654–668 (2015)
- Rosenberg, A., Hirschberg, J.: V-measure: a conditional entropy-based external cluster evaluation measure. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 410–420 (2007)
- Rousseeuw, P., Croux, C.: Alternatives to the median absolute deviation. *J. Am. Stat. Assoc.* **88**(424), 1273–1283 (1993)
- Rufibach, K., Dümbgen, L.: Maximum likelihood estimation of a log-concave density and its distribution function: basic properties and uniform consistency. *Bernoulli* **15**(1), 40–68 (2009)
- Rufibach, K., Dümbgen, L.: *logcondens: Estimate a log-concave probability density from IID observations*. R Package Version 2.1.5. <https://cran.r-project.org/package=logcondens> (2016)

- Siegmund, D., Yakir, B., Zhang, N.: Detecting simultaneous variant intervals in aligned sequences. *Ann. Appl. Stat.* **5**(2A), 645–668 (2011)
- Srivastava, M., Worsley, K.: Likelihood ratio tests for a change in the multivariate normal mean. *J. Am. Stat. Assoc.* **81**(393), 199–204 (1986)
- Truong, C., Oudre, L., Vayatis, N.: Selective review of offline change point detection methods. [arXiv:1801.00718](https://arxiv.org/abs/1801.00718), pp 1–46 (2018)
- Wang, T., Samworth, R.: *InspectChangepoint*: high-dimensional change point estimation via sparse projection. R Package Version 1.0.1. <https://cran.r-project.org/package=InspectChangepoint> (2016)
- Wang, T., Samworth, R.: High dimensional change point estimation via sparse projection. *J. R. Stat. Soc. B Stat. Methodol.* **80**(1), 57–83 (2017)
- Yu, M., Chen, X.: Finite sample change point inference and identification for high-dimensional mean vectors. [arXiv:1711.08747](https://arxiv.org/abs/1711.08747), pp. 1–71 (2020)
- Zhang, N., Siegmund, D., Ji, H., Li, J.: Detecting simultaneous change-points in multiple sequences. *Biometrika* **97**(3), 631–645 (2010)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.