Robust and Scalable Bayesian Online Changepoint Detection

Matias Altamirano [1] **François-Xavier Briol** [1 2] **Jeremias Knoblauch** [1]

Abstract

This paper proposes an online, provably robust, and scalable Bayesian approach for changepoint detection. The resulting algorithm has key advantages over previous work: it provides provable robustness by leveraging the generalised Bayesian perspective, and also addresses the scalability issues of previous attempts. Specifically, the proposed generalised Bayesian formalism leads to conjugate posteriors whose parameters are available in closed form by leveraging diffusion score matching. The resulting algorithm is exact and can be updated through simple algebra.

1. Introduction

Changepoint (CP) detection is the task of identifying sudden changes in the statistical properties of a data stream. The methods to detect CPs are used in applications including systems health monitoring (Stival et al., 2022; Yang et al., 2006), financial data (Kim et al., 2022; Kummerfeld & Danks, 2013), climate change (Reeves et al., 2007; Itoh & Kurths, 2010), and cyber security (Hallgren et al., 2022).

Detecting CPs in an online fashion is an even more challenging task, but can allow practitioners to act on these systems in real-time. In a Bayesian context, the most popular method is *Bayesian online changepoint detection (BOCD)* (Adams & MacKay, 2007; Fearnhead & Liu, 2007). Here, the data stream is assumed to come from one of several different underlying distributions; and the goal is to quantify our uncertainty over the most recent time at which the data distribution changed. BOCD has many desirable properties: it is suitable for multivariate data and has the capacity to quantify uncertainty. However, it also has a significant flaw inherited from Bayesian inference: it is not robust under outliers or model misspecification. This can lead to failures, where most data points inferred to be CPs are simply mild heterogeneities in the data. This is a significant problem, and can causes practitioners to act on safety-critical systems based upon an erroneously declared CPs.

This paper proposes a new generalised Bayesian inference scheme based on *diffusion score matching* (Barp et al., 2019), which is effectively a weighted version of the original score-matching divergence of Hyvärinen (2006). If the weights are chosen appropriately, the resulting posteriors are provably robust, and the corresponding CP detection algorithm, denoted $\mathcal{D}_m$-BOCD, is also robust to outliers.

Additionally, $\mathcal{D}_m$-posteriors also have a conjugacy property for likelihoods of the exponential family so long as the prior is chosen to be a normal, truncated normal, or any other squared exponential distribution. This makes $\mathcal{D}_m$-BOCD very fast: specifically, it ensures that all posteriors used in the algorithm can be updated exactly and efficiently through elementary vector and matrix calculations. If one uses the pruning strategies for the CP posterior proposed in Adams & MacKay (2007), the computational complexity of our algorithm is $\mathcal{O}(T(d^2 + p^2))$; where T is the length of the data stream, d is the dimension of the observations, and p is the number of model parameters. This is the same computational complexity as the original BOCD algorithm.

Beyond that, $\mathcal{D}_m$-BOCD has benefits that make it more attractive than standard BOCD even from a purely computational point of view in certain settings. For example, when modelling d-dimensional observations with non-Gaussian exponential family distributions, we can obtain conjugate $\mathcal{D}_m$-posteriors, even though no conjugate posteriors exist in the standard Bayesian case.

In summary, we make two key contributions:

(1) We derive and propose the $\mathcal{D}_m$-posterior; *proving its robustness and closed form updates* in the process;

(2) We use this posterior for BOCD, leading to the first algorithm that is *both robust and scalable*.

2. Background

Our method merges generalised Bayesian posteriors based on diffusion score matching with the BOCD algorithm. Here, we provide a short explanation of the concepts relevant for understanding this interface.

[1]Department of Statistical Science, University College London, London, United Kingdom [2]The Alan Turing Institute, London, United Kingdom. Correspondence to: Matias Altamirano <matias.altamirano.22@ucl.ac.uk>, François-Xavier Briol <f.briol@ucl.ac.uk>, Jeremias Knoblauch <j.knoblauch@ucl.ac.uk>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

2.1. Bayesian Online Changepoint Detection (BOCD)

Let $x_{1:T}$ be a sequence of observations $x_1, x_2, \ldots, x_T$, where $x_t \in \mathcal{X} \subseteq \mathbb{R}^d$ for the time index $t \in \{1, \ldots, T\}$. Throughout, $x_{1:T}$ follows the product partition model of Barry & Hartigan (1993): the data is partitioned through a sequence of changepoints (CPs) $0 = \tau_1 < \tau_2 < \ldots$ so that the i-th segment is $x_{\tau_i:\tau_{i+1}-1}$, and data within the i-th segment is independently and identically distributed (i.i.d.) conditional on τ_i, τ_{i+1}. In the model underlying BOCD (Adams & MacKay, 2007; Fearnhead & Liu, 2007), the data in each segment is modelled with the same model class $\{p_\theta : \theta \in \Theta\}$, but with a different parameter for each segment. Instead of defining a prior and posterior over the CPs directly, BOCD therefore seeks to infer the so-called *run-length* r_t of the current segment—the amount of time since the most recent CP.

The quantity of interest is the joint distribution between run-length and observations given by

$$p(r_t, x_{1:t}) = \sum_{r_{t-1}=0}^{t-1} \underbrace{p\left(x_t | x_{t-1}^{(r_t)}\right)}_{\text{Predictive Posterior}} \underbrace{H(r_t|r_{t-1})}_{\text{CP prior}} p(r_{t-1}, x_{1:t-1}),$$

where $x_{t-1}^{(r_t)} = x_{t-r_t:t-1}$ is the segment with run-length r_t except the most recent observation x_t, and the predictive of x_t constructed from $x_{t-1}^{(r_t)}$ is

$$p(x_t|x_{t-1}^{(r_t)}) = \int_\Theta p_\theta(x_t)\pi^{\mathrm{B}}(\theta|x_{t-1}^{(r_t)})d\theta, \tag{1}$$

where $\pi^{\mathrm{B}}(\theta|x_{t-1}^{(r_t)}) \propto \prod_{i=1}^{r_t} p_\theta(x_{t-i})\pi(\theta)$ is the Bayes posterior over θ in the current segment.

Unfortunately, BOCD is not robust: it finds spurious CPs whenever the model is a poor description of data. To address this issue, one can replace the standard Bayesian parameter posterior in (1) with a robust generalised Bayesian posterior.

2.2. Generalised Bayesian (GB) inference

If the statistical model p_θ is well-specified so that for some $\theta_0 \in \Theta$, the true data-generating mechanism is p_{θ_0}, standard Bayesian updating is the optimal way of integrating prior information with data (Zellner, 1988). Crucially, this no longer holds if the model is misspecified. In this setting, uncertainties are miscalibrated, posterior inferences are sensitive to outliers and heterogeneity, and the Bayesian update may no longer be the best way of processing information. To address these issues, a recent line of research has advocated for the use of generalised Bayesian inference (see e.g. Grünwald, 2012; Bissiri et al., 2016; Knoblauch et al., 2022; Matsubara et al., 2022b) which, once conditioned on some data $x_{1:T}$, is based on a belief distribution of the form

$$\pi_\omega^{\mathcal{D}}(\theta|x_{1:T}) \propto \pi(\theta)\exp\{-\omega T \cdot \widehat{\mathcal{D}}(\theta)\}. \tag{2}$$

While $\widehat{\mathcal{D}}(\theta)$ could in principle represent any loss function, we consider a narrowed scope. Specifically, for $\mathcal{D}$ being a discrepancy measure on the space of probability measures on $\mathcal{X}$, and p_0 being the true data-generating process, $\widehat{\mathcal{D}} : \Theta \to \mathbb{R}$ uses $x_{1:T}$ to estimate the part of the discrepancy $\mathcal{D}(p_0, p_\theta)$ that depends on θ. Here, $\omega > 0$ is called the *learning rate* and acts as a scaling parameter that determines how quickly the posterior learns from the data.

Using such generalisations is usually done for two main arguments: to provide robustness, and to improve computation.

3. Methodology

We present the methodological innovations of the current paper in three steps: After an exposition of diffusion score matching, we first explain how the resulting generalised Bayesian posterior yields closed form updates. In a second step, we provide formal robustness guarantees for these posteriors. In the last step, we show how to integrate them into the BOCD framework, yielding $\mathcal{D}_m$-BOCD.

3.1. Diffusion Score Matching Bayes

Notation. We write the divergence operator on a vector field f as $\nabla \cdot f$. This condenses the formulae derived in this paper, but we provide all uncondensed versions in Appendix A. The d-dimensional vector (and $d \times p$ sized matrix) of partial derivatives for $f : \mathcal{X} \to \mathbb{R}$ (and $g : \mathcal{X} \to \mathbb{R}^p$) evaluated at $x \in \mathcal{X}$ is written as $\nabla f(x)$ (and $\nabla g(x)$).

Diffusion Score Matching. We consider a particular generalisation of the diffusion matching (Hyvärinen, 2006) hereafter is *diffusion score matching*, which was introduced in Barp et al. (2019) and amounts to a weighted version of the Fisher divergence between the statistical model $\{p_\theta : \theta \in \Theta\}$ and the data-generating process p_0.

This method takes its name from the fact that for a density p on $\mathcal{X}$ and $s_p(x) = \nabla \log p(x)$—the so-called *score function* of the density p—the weighted Fisher divergence is

$$\mathcal{D}_m(p_0\|p_\theta) = \mathbb{E}_{X\sim p_0}\left[\|m^\top(X)(s_{p_\theta}(X) - s_{p_0}(X))\|_2^2\right],$$

for a pointwise invertible matrix-valued function $m : \mathcal{X} \to \mathbb{R}^{d\times d}$. This divergence is therefore minimised by matching the scores of the model to that of the data-generating process p_0.

The function m is also known as diffusion matrix due to the construction of this distance as a Stein discrepancy with a pre-conditioned diffusion Stein operator; see Anastasiou et al. (2023) for full details.

$\mathcal{D}_m$ is a statistical divergence between densities p_0 and p_θ on $\mathcal{X} = \mathbb{R}^d$ whenever $\int_\mathcal{X} |s_{p_\theta}(x) - s_{p_0}(x)|^2 p_0(x)dx < \infty$.

Estimating $\mathcal{D}_m$ directly is challenging, as it would require

estimating the unknown score s_{p_0}. Fortunately, under the aforementioned smoothness and boundary conditions (Appendix B.5) (Liu et al., 2022), we can expand the above equation and use integration by parts. Then, up to a constant that does not depend on θ, we can rewrite $\mathcal{D}_m(p_0\|p_\theta)$ as

$$\mathbb{E}_{X\sim p_0}[\|(m^\top s_{p_\theta})(X)\|_2^2 + (2\nabla\cdot(mm^\top s_{p_\theta}))(X)]. \quad (3)$$

Crucially, the quantity above no longer features s_{p_0}, and only depends on p_0 through an expectation. This leads to a natural estimator which for $x_{1:T}$ is given by

$$\widehat{\mathcal{D}}_m(\theta) = \tfrac{1}{T}\sum_{t=1}^T d_m(\theta, x_t), \quad \text{where}$$
$$d_m(\theta, x_t) = \|(m^\top s_{p_\theta})(x_t)\|_2^2 + (2\nabla\cdot(mm^\top s_{p_\theta}))(x_t).$$

$\mathcal{D}_m$-Bayes. Based on the estimator $\widehat{\mathcal{D}}_m$ for the part of $\mathcal{D}_m$ that depends on θ, we can construct

$$\pi_\omega^{\mathcal{D}_m}(\theta|x_{1:T}) \propto \pi(\theta)\exp(-\omega T\widehat{\mathcal{D}}_m(\theta)). \quad (4)$$

Using score matching for a generalised Bayes posterior was first discussed in passing in Section 4.2 of Giummolè et al. (2019), though the context is about reference priors for objective Bayesian inference, and the method is only briefly mentioned. This previous work also does not robustify the resulting posterior through the introduction of a weighting matrix m, or derive its conjugate posteriors.

3.2. Conjugacy for Exponential Family Models

The conjugacy of posteriors of the form (4) make them more attractive than potential alternatives. For exponential family likelihoods, these posteriors depend on two parameters available in closed form. The exponential family is given the collection of models with a probability density function

$$p_\theta(x) = \exp\left(\eta(\theta)^\top r(x) - a(\theta) + b(x)\right), \quad (5)$$

where $\eta : \Theta \to \mathbb{R}^p$, $r : \mathcal{X} \to \mathbb{R}^p$, $a : \Theta \to \mathbb{R}$, and $b : \mathcal{X} \to \mathbb{R}$. When $\eta(\theta) = \theta$, we say that the exponential family model is in natural form, and one can reparametrise a model to natural form by reparameterising with the map η^{-1}. Exponential family class of distributions includes the Gaussian, exponential, Gamma, and Beta distributions.

Proposition 3.1. *If p_θ is given by* (5)*, then*

$$\pi_\omega^{\mathcal{D}_m}(\theta|x_{1:T}) \propto \pi(\theta)\exp(-\omega T[\eta(\theta)^\top\Lambda_T\eta(\theta) + \eta(\theta)^\top\nu_T]),$$

for $\Lambda_T = \frac{1}{T}\sum_{t=1}^T \Lambda(x_t)$, $\nu_T = \frac{2}{T}\sum_{t=1}^T \nu(x_t)$, and

$$\Lambda(x) = (\nabla r^\top mm^\top\nabla r)(x),$$
$$\nu(x) = \left(\nabla r^\top mm^\top\nabla b + \nabla\cdot(mm^\top\nabla r)\right)(x).$$

Taking $\eta(\theta) = \theta$ and choosing a squared exponential prior $\pi(\theta) \propto \exp\left(-\frac{1}{2}(\theta-\mu)^\top\Sigma^{-1}(\theta-\mu)\right)$, also makes $\pi_\omega^{\mathcal{D}_m}(\theta|x_{1:T})$ a (truncated) normal of the form

$$\pi_\omega^{\mathcal{D}_m}(\theta|x_{1:T}) \propto \exp\left(-\tfrac{1}{2}(\theta-\mu_T)^\top\Sigma_T^{-1}(\theta-\mu_T)\right),$$

for $\Sigma_T^{-1} = \Sigma^{-1} + 2\omega T\Lambda_T$ and $\mu_T = \Sigma_T\left(\Sigma^{-1}\mu - \omega T\nu_T\right)$.

The proof is in Appendix B.1. The natural exponential family allows us to recover a form of Gaussian conjugacy, since the diffusion score matching squared becomes a quadratic form in this case. This renders $\mathcal{D}_m$-Bayes scalable; as we will elaborate upon in Section 3.4, Σ_T^{-1} and μ_T can be updated with a new observation in $\mathcal{O}(p^2 + d^2)$ operations.

3.3. Global Bias-Robustness

Building a BOCD algorithm based on $\pi_\omega^{\mathcal{D}_m}(\theta|x_{1:T})$ is attractive not only computationally, but also due to its robustness.

We prove this robustness formally by using the framework first introduced by Ghosh & Basu (2016), and extended by Matsubara et al. (2022b). See details in Appendix B.3

We build on these advances, and provide a simple condition on m for global bias-robustness of $\pi_\omega^{\mathcal{D}_m}(\theta|x_{1:T})$ in some exponential family models.

Proposition 3.2. *If p_θ is as in* (5) *so that $\eta(\theta) = \theta$ and $\nabla b = 0$, and if the prior is a squared exponential as in Proposition 3.1, then $\pi_\omega^{\mathcal{D}_m}(\theta|x_{1:T})$ is globally bias-robust if $m : \mathcal{X} \to \mathbb{R}^{d\times d}$ is chosen so that $\theta^\star \neq 0_p$ and*

$$m_{ij}(x) = \begin{cases} \frac{1}{\sqrt{1+(\nabla r(x)\theta^\star)_i^2}} & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

While m could in principle depend on θ, this would break the conjugacy presented in Proposition 3.1. The above choice of m does *not* depend on θ, and therefore maintains the computational advantages of $\pi_\omega^{\mathcal{D}_m}(\theta|x_{1:T})$. The result's conditions are also mild: we can always ensure that $\eta(\theta) = \theta$ by re-parameterising. Similarly, most distributions of interest satisfy $\nabla b = 0$. Examples include Gaussians, exponentials, (inverse) Gamma, and Beta distributions.

3.4. $\mathcal{D}_m$-BOCD

Using our robust posterior within BOCD is straightforward, as its only appearance is in the posterior predictive via

$$p\left(x_t|x_{t-1}^{(r)}\right) = \int_\Theta p_\theta(x_t)\pi_\omega^{\mathcal{D}_m}\left(\theta|x_{t-1}^{(r)}\right)d\theta.$$

If p_θ is a natural exponential family with a squared exponential prior, then $\pi_\omega^{\mathcal{D}_m}(\theta|x_{t-1}^{(r)}))$ is a normal distribution parameterised by inverse covariance matrix $\Sigma_{t-1,r}^{-1}$ and mean $\mu_{t-1,r}$ by virtue of 3.1. This makes the predictive easy to compute—either in closed form or by sampling from $\pi_\omega^{\mathcal{D}_m}$

Our approach allows for a cheap and exact update: if we store $\Sigma_{t-1,r}^{-1}$ and $\mu_{t-1,r}$, we can perform the update $\pi_\omega^{\mathcal{D}_m}(\theta|x_{t-1}^{(r)}) \mapsto \pi_\omega^{\mathcal{D}_m}(\theta|x_t^{(r+1)})$ that adds x_t into the parameter posterior of the segment $x_{t-1}^{(r)}$ via

$$\Sigma_{t,r+1}^{-1} = \Sigma_{t-1,r}^{-1} + 2\omega\Lambda(x_t),$$
$$\mu_{t,r+1} = \Sigma_{t,r+1}\left(\Sigma_{t-1,r}^{-1}\mu_{t-1,r} - 2\omega\nu(x_t)\right).$$

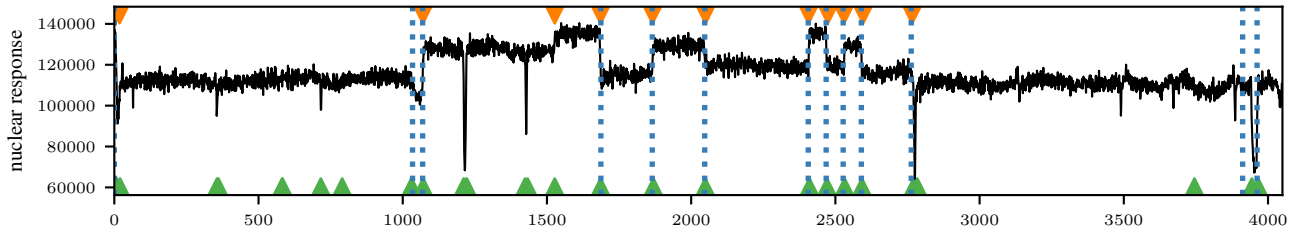


Figure 1. Well-log data. MAP segmentation indicated by **blue** dashed lines for $\mathcal{D}_m$-BOCD, ▼ for β-BOCD, and ▲ for standard BOCD. Standard BOCD mistakenly labels outliers as CPs, while both $\mathcal{D}_m$-BOCD and β-BOCD are robust and identify lasting changes.

If we have access to the un-inverted matrix $\Sigma_{t,r+1}$, all of these operations are basic matrix and vector additions or multiplications that take $\mathcal{O}(p^2 + d^2)$ operations to execute. While naively computing $\Sigma_{t,r+1}$ from $\Sigma^{-1}_{t,r+1}$ would take $\mathcal{O}(p^3)$ operations, we can apply the Sherman-Morrison formula to the update of $\Sigma^{-1}_{t,r+1}$ to reduce this to $\mathcal{O}(p^2)$, maintaining the overall complexity of $\mathcal{O}(p^2+d^2)$. In CP methods for high-frequency data, both the number of parameters p and the data dimension d are typically small, so that an update of $\mathcal{O}(p^2 + d^2)$ is attractive.

4. Experiments

We investigate $\mathcal{D}_m$-BOCD empirically in several numerical experiments. In doing so, we highlight its computational and inferential advantages over standard BOCD and β-BOCD. In all experiments, we choose conjugate priors as in Proposition 3.1, and m and ω as in Section 3.4. All code and data is publicly available at `https://github.com/maltamiranomontero/DSM-bocd`.

Computational complexity. We compare the complexity of the three BOCD methods in different settings and show that $\mathcal{D}_m$-BOCD is considerably faster than β-BOCD, even when sampling is needed. Moreover, Figure 2 shows that $\mathcal{D}_m$ is as fast as standard BOCD when $d = 1$ and the predictive posterior is available in closed form. See Appendix D.1 for details.

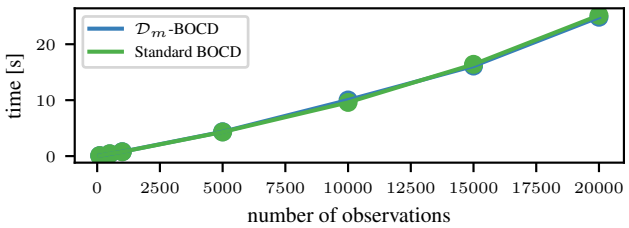


Figure 2. Overall time in seconds versus number of observations. We observe that both methods are equally fast for any number of observations.

Accuracy and detection delay. We quantify the method's performance advantage by comparing detection delay and accuracy on artificially generated data with outliers. We generate 600 samples with 2% of outliers and 6 CPs; then, we report the positive predictive value (PPV), true positive rate (TPR), and detection delay. See Appendix D.3 for the exact expression of the metrics. Table 1 shows that our method detects the same amount of true positives as the standard BOCD while not detecting many false positives, showing the strength of our method. Moreover, the detection delay shows that in spite of being robust to outliers, $\mathcal{D}_m$-BOCD does not cause any delay in the detection of CP.

Method	PPV	TPR	Delays
$\mathcal{D}_m$-BOCD	**0.907±0.154**	**0.883±0.13**	1.643±0.475
Standard BOCD	0.6±0.128	0.833±0.149	**1.05±1.545**

Table 1. Performance indices-mean and standard deviation-using the positive predictive value (PPV), true positive rate (TPR), and the detection delays over 10 realisations. For PPV and TPR, the nearest to 1, the better. For delays, the lower, the better.

Well-log. The well-log data was introduced in Ruanaidh & Fitzgerald (1996), and consists in 4,050 nuclear magnetic resonance measurements recorded while drilling a well. CPs in the sequence correspond to changes in the sediment layers the drill is penetrating. On top of these clear changes, the data contains outliers and contaminants corresponding to more short-term events in geological history—such as flooding, earthquakes, or volcanic activity. When this data set is studied, its outliers have traditionally been removed before CP detection algorithms are run (see e.g. Adams & MacKay, 2007; Ruggieri & Antonellis, 2016; Levy-leduc & Harchaoui, 2008). We leave them in, and Figure 1 shows that this is unproblematic for $\mathcal{D}_m$-BOCD, but does lead to falsely labelled CPs with BOCD. We also compare the algorithm with β-BOCD (Knoblauch et al., 2018), and find that the detected changes are almost identical. On a machine with processor Intel i7-7500U 2.7 GHz, and 12GB of RAM, $\mathcal{D}_m$-BOCD took about 10 times less than β-BOCD.

Multivariate synthetic data. In certain settings, $\mathcal{D}_m$-posteriors are conjugate when standard posteriors are not. An example is a multivariate time series whose dimensions follow different distributions belonging to the exponential family. To this end, we generate 1000 samples from a time series with CPs at $t = 250, 750$. Conditional on the CPs, the data is generated independently from an exponential in the first dimension and Gaussian distribution in the second dimension. $\mathcal{D}_m$-BOCD is immediately applicable, and Figure 4 shows that the algorithm functions reliably. We do

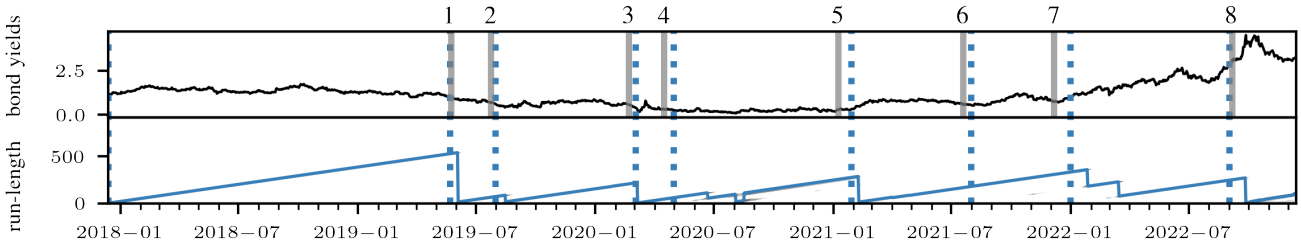


Figure 3. UK's 10 year government bond yield 2018-2023. The MAP-segmentation resulting from $\mathcal{D}_m$-BOCD is indicated in dashed **blue** lines. The bottom panel displays the corresponding run-length posterior, with the most likely run-length marked in **blue**. A series of political events of national importance closely track the segmentation, and are marked with solid **gray** lines: 1. Theresa May announces her resignation from her position as prime minister; 2. Boris Johnson sworn in as prime minister; 3. the first Covid case recorded in EU; 4. the first Covid wave in the UK is officially declared; 5. the third Covid wave in the UK is officially declared; 6. the legal limits on social contact removed in UK; 7. Covid 'Plan B' measures are implemented in UK in response to the spread of the Omicron variant; 8. Liz Truss is sworn in as prime minister.

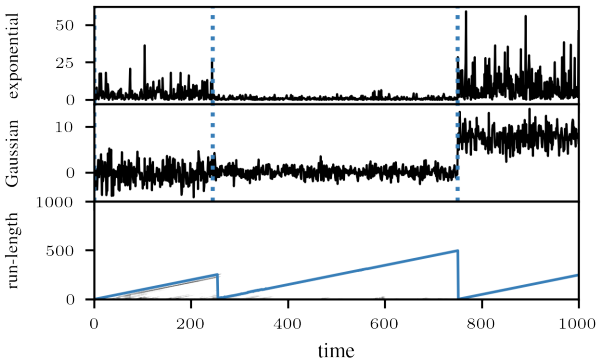


Figure 4. Multivariate synthetic example. A 2-dimensional CP problem. For the chosen model, $\mathcal{D}_m$-BOCD is computationally efficient, but standard BOCD is computationally infeasible. The MAP segmentation is indicated by dashed **blue** lines, and the bottom panel shows the run-length distribution, with the most likely value in **blue**.

not compare to BOCD in this setting: for this model, standard Bayesian posteriors would require expensive sampling algorithms or variational approximations to be employed, rendering the algorithm impractical.

UK 10 year government bond yield. Finally, we run the $\mathcal{D}_m$-BOCD on the daily yield of 10 year UK government bonds from 2018 to 2022 (see Figure 3). The data is publicly available via the Bank of England database.[1] Since the 10-year yield has been positive throughout history, we model it using the gamma distribution. As shown in Figure 3, we detect changes in the yield curve that correspond to important political events in the UK. This distribution leads to a $\mathcal{D}_m$-posterior that is a Gaussian truncated at zero. For standard Bayes, a conjugate prior exists, but it leads to a posterior with intractable normalisation constant. Like the multivariate synthetic data example, this constitutes another instance where $\mathcal{D}_m$-posteriors have better computational properties than standard Bayes.

[1] https://www.bankofengland.co.uk/boeapps/database/

5. Conclusion

We proposed $\mathcal{D}_m$-BOCD, a new version of BOCD that is both *robust to outliers and scalable*. The algorithm relies on a new generalised Bayesian inference scheme constructed with diffusion score-matching. These posteriors have closed form updates for models that are members of the exponential family, and provide robustness by appropriately tuning the diffusion matrix m. For T observations, d-dimensional data, and p model parameters, the overall run time of the method is $\mathcal{O}(T(p^2 + d^2))$, and we demonstrate that it is just as fast as standard BOCD. By showcasing the various computational and inferential benefits of $\mathcal{D}_m$-BOCD on a range of examples, we demonstrate that it is a powerful and needed addition to the literature.

References

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

Anastasiou, A., Barp, A., Briol, F.-X., Ebner, B., Gaunt, R. E., Ghaderinezhad, F., Gorham, J., Gretton, A., Ley, C., Liu, Q., Mackey, L., Oates, C. J., Reinert, G., and Swan, Y. Stein's method meets computational statistics: A review of some recent developments. *Statistical Science*, 38: 120–139, 2023.

Barp, A., Briol, F.-X., Duncan, A., Girolami, M., and Mackey, L. Minimum Stein discrepancy estimators. In *Advances in Neural Information Processing Systems*, pp. 12964–12976, 2019.

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

Basu, A., Harris, I. R., Hjort, N. L., and Jones, M. Robust and efficient estimation by minimising a density power divergence. *Biometrika*, 85(3):549–559, 1998.

Bissiri, P. G., Holmes, C. C., and Walker, S. G. A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):1103–1130, 2016.

Bochkina, N. Bernstein-von Mises theorem and misspecified models: a review. *arXiv:2204.13614*, 2022.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`.

Chérief-Abdellatif, B.-E. and Alquier, P. MMD-Bayes: Robust Bayesian estimation via maximum mean discrepancy. In *Symposium on Advances in Approximate Bayesian Inference*, 2020.

Fearnhead, P. and Liu, Z. On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):589–605, 2007.

Ghosh, A. and Basu, A. Robust Bayes estimation using the density power divergence. *Annals of the Institute of Statistical Mathematics*, 68(2):413–437, 2016.

Giummolè, F., Mameli, V., Ruli, E., and Ventura, L. Objective Bayesian inference with proper scoring rules. *TEST*, 28(3):728–755, 2019.

Grünwald, P. The safe Bayesian. In *International Conference on Algorithmic Learning Theory*, pp. 169–183, 2012.

Hallgren, K. L., Heard, N. A., and Adams, N. M. Changepoint detection in non-exchangeable data. *Statistics and Computing*, 32(6):1–19, 2022.

Hooker, G. and Vidyashankar, A. Bayesian model robustness via disparities. *TEST*, 23(3):556–584, 2014.

Huber, P. J. Robust statistics. *Wiley Series in Probability and Mathematical Statistics*, 1981.

Hyvärinen, A. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–708, 2006.

Itoh, N. and Kurths, J. Change-point detection of climate time series by nonparametric method. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 1, pp. 445–448, 2010.

Kim, K., Park, J. H., Lee, M., and Song, J. W. Unsupervised change point detection and trend prediction for financial time-series using a new CUSUM-based approach. *IEEE Access*, 10:34690–34705, 2022.

Knoblauch, J. Robust deep Gaussian processes. *arXiv:1904.02303*, 2019.

Knoblauch, J. and Damoulas, T. Spatio-temporal Bayesian on-line changepoint detection with model selection. In *International Conference on Machine Learning*, pp. 2718–2727, 2018.

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

Knoblauch, J., Jewson, J., and Damoulas, T. An optimization-centric view on Bayes' rule: Reviewing and generalizing variational inference. *Journal of Machine Learning Research*, 23(132):1–109, 2022.

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

Levy-leduc, C. and Harchaoui, Z. Catching change-points with lasso. In *Advances in Neural Information Processing Systems*, pp. 617–624, 2008.

Liu, S., Kanamori, T., and Williams, D. J. Estimating density models with truncation boundaries using score matching. *Journal of Machine Learning Research*, 23(186):1–38, 2022.

Lyddon, S. P., Holmes, C. C., and Walker, S. G. General Bayesian updating and the loss-likelihood bootstrap. *Biometrika*, 106(2):465–478, 2019.

Matsubara, T., Knoblauch, J., Briol, F.-X., Oates, C., et al. Generalised Bayesian inference for discrete intractable likelihood. *arXiv:2206.08420*, 2022a.

Matsubara, T., Knoblauch, J., Briol, F.-X., and Oates, C. J. Robust generalised Bayesian inference for intractable likelihoods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 84(3):997–1022, 2022b.

Reeves, J., Chen, J., Wang, X. L., Lund, R., and Lu, Q. Q. A review and comparison of changepoint detection techniques for climate data. *Journal of Applied Meteorology and Climatology*, 46(6):900–915, 2007.

Ruanaidh, J. J. O. and Fitzgerald, W. J. *Numerical Bayesian methods applied to signal processing*. Springer Science & Business Media, 1996.

Ruggieri, E. and Antonellis, M. An exact approach to Bayesian sequential change point detection. *Computational Statistics & Data Analysis*, 97:71–86, 2016.

Saatçi, Y., Turner, R. D., and Rasmussen, C. E. Gaussian process change point models. In *International Conference on Machine Learning*, 2010.

Stival, M., Bernardi, M., and Dellaportas, P. Doubly-online changepoint detection for monitoring health status during sports activities. *arXiv:2206.11578*, 2022.

Syring, N. and Martin, R. Calibrating general posterior credible regions. *Biometrika*, 106(2):479–486, 2019.

Wu, P.-S. and Martin, R. A comparison of learning rate selection methods in generalized Bayesian inference. *Bayesian Analysis*, 18(1):105–132, 2023.

Yang, P., Dumont, G., and Ansermino, J. M. Adaptive change detection in heart rate trend monitoring in anesthetized children. *IEEE Transactions on Biomedical Engineering*, 53(11):2211–2219, 2006.

Yu, S., Drton, M., and Shojaie, A. Generalized score matching for non-negative data. *Journal of Machine Learning Research*, 20:1–70, 2019.

Zellner, A. Optimal information processing and Bayes's theorem. *The American Statistician*, 42(4):278–280, 1988.

Supplementary Materials

In Appendix A, we provide mathematical background. In Appendix B, we present the proofs and derivations of all the theoretical results in our paper, while Appendix D contains additional details regarding our experiments.

A. Background

Let $\nabla = (\partial/\partial x_1, \ldots, \partial/\partial x_d)^\top$, $f : \mathcal{X} \to \mathbb{R}^d$ and $g : \mathcal{X} \to \mathbb{R}^{d\times p}$; the divergence operator is define as follows:

$$(\nabla \cdot f)(x) = \sum_{i=1}^d \frac{\partial f_i}{\partial x_i}(x), \qquad (\nabla \cdot g)_j(x) = \sum_{i=1}^d \frac{\partial g_{ij}}{\partial x_i}(x), \quad \forall j \in \{1, ..., p\}.$$

Expanding the term $\nabla \cdot mm^\top \nabla \log p_\theta(x)$ appearing as part of $d_m(\theta, x)$, we get

$$\begin{aligned}
\nabla \cdot mm^\top \nabla \log p_\theta(x) &= \sum_{i=1}^d \frac{\partial}{\partial x_i}(mm^\top \nabla \log p_\theta(x))_i \\
&= \sum_{i=1}^d \sum_{j=1}^d \frac{\partial}{\partial x_i}\left((mm^\top)_{ij}(\nabla \log p_\theta(x))_j\right) \\
&= \sum_{i=1}^d \sum_{j=1}^d \left(\frac{\partial}{\partial x_i}(mm^\top)_{ij}\right)(\nabla \log p_\theta(x))_j + \sum_{i=1}^d \sum_{j=1}^d (mm^\top)_{ij}\left(\nabla^2 \log p_\theta(x)\right)_{ij} \\
&= \sum_{i=1}^d \sum_{j=1}^d \left(\frac{\partial}{\partial x_i}(mm^\top)_{ij}\right)(\nabla \log p_\theta(x))_j + \sum_{j=1}^d \left(mm^\top \nabla^2 \log p_\theta(x)\right)_{jj} \\
&= \sum_{j=1}^d \sum_{i=1}^d \left(\frac{\partial}{\partial x_i}(mm^\top)_{ij}\right)(\nabla \log p_\theta(x))_j + \mathrm{Tr}\left(mm^\top \nabla^2 \log p_\theta(x)\right).
\end{aligned}$$

Where ∇^2 is the Hessian. The expression in the last line is more straightforward to implement in practice and it is therefore the one we use in our code.

The term $\nu(x)$ in Proposition 3.1 contains $(\nabla \cdot (mm^\top \nabla r)(x))$. The j-th index of this p-dimensional vector equals

$$(\nabla \cdot (mm^\top \nabla r)(x))_j = \sum_{i=1}^d \frac{\partial}{\partial x_i}(mm^\top \nabla r(x))_{ij}.$$

A.1. Generalised Bayesian Inference in BOCD

Knoblauch et al. (2018) first proposed a robustification of BOCD based on (2) and the β-divergence, which is robust and well-defined for $\beta \in (0, \infty)$ when p_θ is uniformly bounded on $\mathcal{X}$, and whose natural estimator was derived by Basu et al. (1998) and is given by

$$\widehat{\mathcal{D}}_\beta(\theta) = \frac{1}{T}\sum_{t=1}^T \frac{1}{1+\beta}\int_{\mathcal{X}} p_\theta(x)^{1+\beta}dx + \frac{1}{\beta}p_\theta(x_t)^\beta.$$

While the resulting method can be made robust, it has several key failures that make it computationally infeasible in most settings. Firstly, the loss depends on $\int_{\mathcal{X}} p_\theta(x)^{1+\beta}dx$. Unless this integral is available in closed form, using $\widehat{\mathcal{D}}_\beta$ will introduce the same challenges as working with an intractable likelihood in a standard Bayesian setting. Secondly, the hyperparameter β enters the loss as the exponent of a likelihood. Numerically, this makes the loss extremely sensitive to even very minor changes in β, which makes it very difficult to tune β and counteracts the very robustness one hopes to achieve. This numerical instability is compounded by the fact that (2) depends on the exponentiation of $\widehat{\mathcal{D}}_\beta$—if p_θ is an exponential family member, then even if one ignores the integral term, $\exp\{-\omega T\widehat{\mathcal{D}}_\beta(\theta)\}$ is a double exponential. Thirdly, posteriors based on $\widehat{\mathcal{D}}_\beta$ often have to be approximated using variational methods. Since this has to be done for all run-lengths r_t at each time step t for the recursive relationship powering the algorithm, this represents a substantive computational overhead.

Taken together, these issues often render posteriors based on the β-divergence computationally infeasible; especially in high-dimensional settings. In principle, one could replace the β-divergence with various robust alternatives whose numerical issues are less substantive and whose hyperparameters are easier to tune—such as α-divergences (Hooker & Vidyashankar, 2014), γ-divergences (Knoblauch, 2019), or maximum mean discrepancies (Chérief-Abdellatif & Alquier, 2020). Unfortunately, none of these alternatives alleviate the problem of computationally expensive variational approximations. This is an issue, since ultimately, it is the conjugate forms that can be updated in terms of sufficient statistics that make BOCD computationally attractive.

In the face of this, it may be tempting to postulate an inherent trade-off between robustness and computational tractability for generalised Bayes. But this is not so; recently, it was shown that robust posteriors based on kernel Stein discrepancies

have a conjugacy property (Proposition 2 of Matsubara et al., 2022b). These generalised posteriors however are not suitable for BOCD: Updating them from $t-1$ to t observations takes $\mathcal{O}(t)$ operations—as opposed to the $\mathcal{O}(1)$ operations required for standard Bayesian posteriors. Such updates would lead to an algorithm whose computational demands per iteration increase linearly the longer it is run, leading to an 'online' algorithm in name only. This is why the current paper proposes a new class of generalised posteriors based on diffusion score matching (Barp et al., 2019): we prove that they are robust, and lead to conjugacy, with closed forms updates that take $\mathcal{O}(1)$ operations.

B. Theoretical Results

In this section we present the derivation of the $\mathcal{D}_m$-posterior, along with the proof of its robustness.

B.1. Proof of Proposition 3.1

In this Subsection we present the proof of the main result of Section 3.2: the conjugacy for exponential family models.

Proof. Let p_θ be an exponential family model. Then $\nabla \log p_\theta = \nabla r(x)^\top \eta(\theta) + \nabla b(x)$, and the DSM estimator has the following form:

$$\widehat{\mathcal{D}}_m(\theta) = \tfrac{1}{T}\sum_{t=1}^T \underbrace{\|m^\top(\nabla r(x_t)\eta(\theta) + \nabla b(x_t))\|_2^2}_{(1)} + 2\underbrace{\nabla\cdot(mm^\top(\nabla r(x_t)\eta(\theta) + \nabla b(x_t)))}_{(2)}.$$

Let $\overset{+C}{=}$ indicate equality up to an additive term that does not depend on θ.

$$\begin{aligned}(1) &= \eta(\theta)^\top \nabla r(x_t)^\top mm^\top \nabla r(x_t)\eta(\theta) + \nabla b(x_t)^\top mm^\top \nabla b(x_t) + 2\eta(\theta)^\top \nabla r(x_t)^\top mm^\top \nabla b(x_t)\\ &\overset{+C}{=} \eta(\theta)^\top \nabla r(x_t)^\top mm^\top \nabla r(x_t)\eta(\theta) + 2\eta(\theta)^\top \nabla r(x_t)^\top mm^\top \nabla b(x_t),\end{aligned}$$

and

$$\begin{aligned}(2) &= \nabla\cdot(mm^\top \nabla r(x_t)\eta(\theta)) + \nabla\cdot(mm^\top \nabla b(x_t))\\ &\overset{+C}{=} \nabla\cdot(mm^\top \nabla r(x_t)\eta(\theta))\\ &= \eta(\theta)^\top(\nabla\cdot(mm^\top \nabla r(x_t))).\end{aligned}$$

Therefore, $\widehat{\mathcal{D}}_m(\theta) = \eta(\theta)^\top \Lambda_T \eta(\theta) + \eta(\theta)^\top \nu_T$ where

$$\begin{aligned}\Lambda_T &:= \tfrac{1}{T}\sum_{t=1}^T \nabla r(x_t)^\top mm^\top \nabla r(x_t),\\ \nu_T &:= \tfrac{2}{T}\sum_{t=1}^T \nabla r(x_t)^\top mm^\top \nabla b(x_t) + \nabla\cdot(mm^\top \nabla r(x_t)).\end{aligned}$$

Now, assuming the prior has a p.d.f. π, the DSM-Bayes generalised posterior has a p.d.f.

$$\pi_\omega^{\mathcal{D}_m} \propto \pi(\theta)\exp\left(-\omega T[\eta(\theta)^\top \Lambda_T \eta(\theta) + \eta(\theta)^\top \nu_T]\right).$$

For $\eta(\theta) = \theta$, and the prior $\pi(\theta) \propto \exp\left(-\frac{1}{2}(\theta-\mu)^\top \Sigma^{-1}(\theta-\mu)\right)$, we obtain the generalised posterior by completing the square as follows:

$$\begin{aligned}\pi_\omega^{\mathcal{D}_m}(\theta) &\propto \exp\left(-\tfrac{1}{2}(\theta-\mu)^\top \Sigma^{-1}(\theta-\mu_T)\right)\exp\left(-\omega T[\theta^\top \Lambda_T \theta + \theta^\top \nu_T]\right)\\ &= \exp\left(-\tfrac{1}{2}\left(\theta^\top \Sigma^{-1}\theta - 2\theta^\top \Sigma^{-1}\mu + \mu^\top \Sigma^{-1}\mu + \theta^\top 2\omega T\Lambda_T\theta + \theta^\top 2\omega n\nu_T\right)\right)\\ &\propto \exp\left(-\tfrac{1}{2}\left(\theta^\top(\Sigma^{-1} + 2\omega T\Lambda_T)\theta - 2\theta^\top(\Sigma^{-1}\mu - \omega T\nu_T)\right)\right)\\ &\propto \exp\left(-\tfrac{1}{2}(\theta-\mu_T)^\top \Sigma_T^{-1}(\theta-\mu_T)\right),\end{aligned}$$

where

$$\begin{aligned}\Sigma_T^{-1} &:= \Sigma^{-1} + 2\omega T\Lambda_T,\\ \mu_T &:= \Sigma_T\left(\Sigma^{-1}\mu - \omega T\nu_T\right).\end{aligned}$$

■

B.2. Update Parameters for online DSM-Bayes

In this Section we derive the efficient parameter updates presented in Section 3.4. To do so, we expand the expressions Λ and ν as follows:

$$\begin{aligned}
\Lambda_{T+1} &:= \tfrac{1}{T+1}\textstyle\sum_{t=1}^{T+1} \nabla r(x_t)^\top m m^\top \nabla r(x_t) \\
&= \tfrac{1}{T+1}\left(\textstyle\sum_{t=1}^{T} \nabla r(x_t)^\top m m^\top \nabla r(x_t) + \nabla r(x_{T+1})^\top m m^\top \nabla r(x_{T+1})\right) \\
&= \tfrac{1}{T+1}\left(n\Lambda_T + \nabla r(x_{T+1})^\top m m^\top \nabla r(x_{T+1})\right) \\
\nu_{T+1} &:= \tfrac{2}{T+1}\textstyle\sum_{t=1}^{T+1} \nabla r(x_t)^\top m m^\top \nabla b(x_t) + \nabla\cdot(m m^\top \nabla r(x_t)) \\
&= \tfrac{1}{T+1}\left(T\nu_T + 2(\nabla r(x_{T+1})^\top m m^\top \nabla b(x_{T+1}) + \nabla\cdot(m m^\top \nabla r(x_{T+1}))\right).
\end{aligned}$$

Now, assuming the prior has the conjugate form of Proposition 3.1, the $\mathcal{D}_m$-posterior is given by

$$\pi_\omega^{\mathcal{D}_M}(\theta) \propto \exp\left(-\tfrac{1}{2}(\theta-\mu_T)^\top \Sigma_T^{-1}(\theta-\mu_T)\right),$$

where

$$\begin{aligned}
\Sigma_T^{-1} &:= \Sigma^{-1} + 2\omega n \Lambda_T, \\
\mu_T &:= \Sigma_T\left(\Sigma^{-1}\mu - \omega n \nu_T\right).
\end{aligned}$$

So the parameter updates for Σ_T and μ_T as T increases are:

$$\begin{aligned}
\Sigma_{T+1}^{-1} &:= \Sigma^{-1} + 2\omega(T+1)\Lambda_{T+1} \\
&= \Sigma^{-1} + 2\omega\left(n\Lambda_T + \nabla t(x_{T+1})^\top m m^\top \nabla t(x_{T+1})\right) \\
&= \Sigma_T^{-1} + 2\omega \nabla r(x_{T+1})^\top m m^\top \nabla r(x_{T+1}) \\
\mu_{T+1} &:= \Sigma_{T+1}\left(\Sigma^{-1}\mu - \omega(T+1)\nu_{T+1}\right) \\
&= \Sigma_{T+1}\left(\Sigma^{-1}\mu - \omega\left(T\nu_T + 2(\nabla r(x_{T+1})^\top m m^\top \nabla b(x_{T+1}) + \nabla\cdot(m m^\top \nabla r(x_{T+1}))\right)\right) \\
&= \Sigma_{T+1}\left(\Sigma_T^{-1}\mu_T - 2\omega\left(\nabla r(x_{T+1})^\top m m^\top \nabla b(x_{T+1}) + \nabla\cdot(m m^\top \nabla r(x_{T+1}))\right)\right).
\end{aligned}$$

obtaining the desired expression.

B.3. Global Bias-Robustness

In this Subsection, we present the theory necessary to prove that the generalized posterior presented in Section 3.1 is global bias-robust conditioned to the choice of m.

Building a BOCD algorithm based on $\pi_\omega^{\mathcal{D}_m}(\theta|x_{1:T})$ is attractive not only computationally, but also due to its robustness. We prove this robustness formally by using the classical framework of ε-contamination models (see, e.g. Huber, 1981). Given a distribution $\mathbb{P}$, we consider its ε-contaminated counterpart $\mathbb{P}_{\varepsilon,y} = (1-\varepsilon)\mathbb{P} + \varepsilon\delta_y$, where δ_y is the dirac-measure at some $y \in \mathcal{X}$, and $\varepsilon \in [0,1]$. The classical perspective on robustness proceeds by defining a point estimator $E : \mathcal{P}(\mathcal{X}) \to \Theta$ that maps from $\mathcal{P}(\mathcal{X})$, the space of distributions on $\mathcal{X}$, to Θ. One then investigates its robustness via $\lim_{\varepsilon\to 0}\frac{1}{\varepsilon}\|E(\mathbb{P}) - E(\mathbb{P}_{\varepsilon,y})\|_2$, which under mild conditions is equivalent to the derivative $\frac{\partial}{\partial\varepsilon}\|E(\mathbb{P}_{\varepsilon,y})\|_2\big|_{\varepsilon=0}$. This limit is the so-called *influence function*. It quantifies the impact of an infinitesimal contamination at y on the estimator, and is a classical tool to measure outlier robustness.

The Bayesian case is slightly more complicated: we are not concerned by estimators on Θ, but on $\mathcal{P}(\Theta)$. The estimates under study are thus infinite-dimensional objects that vary over Θ. To get a handle on this, we first define an influence function *pointwise* for each $\theta \in \Theta$. To this end, note that $\widehat{\mathcal{D}}_m(\theta) = \mathbb{E}_{X\sim\mathbb{P}_T}[d_m(\theta, X)]$. We can now define the density-valued estimator $\pi_\omega^{\mathcal{D}_m}(\theta|\mathbb{P}) \propto \pi(\theta)\exp\{-\omega T \mathbb{E}_{X\sim\mathbb{P}}[d_m(\theta,X)]\}$, noting $\pi_\omega^{\mathcal{D}_m}(\theta|\mathbb{P}_T) = \pi_\omega^{\mathcal{D}_m}(\theta|x_{1:T})$ for $\mathbb{P}_T = \frac{1}{T}\sum_{t=1}^T \delta_{x_t}$. Its pointwise posterior influence function (PIF) is

$$\text{PIF}(y,\theta,\mathbb{P}) = \frac{d}{d\varepsilon}\pi_\omega^{\mathcal{D}_m}(\theta|\mathbb{P}_{\varepsilon,y})\big|_{\varepsilon=0}.$$

Since this is a definition of sensitivity that is local to both θ and y, making a global statement for all of $\pi_\omega^{\mathcal{D}_m}(\theta|x_{1:T})$ requires that we aggregate a notion of sensitivity over both arguments. The easiest way to do this is to investigate $\sup_{\theta\in\Theta, y\in\mathcal{X}} \mathrm{PIF}(y,\theta,\mathbb{P}_T)$. If this double supremum is bounded, we call a posterior *globally bias-robust*, which means that the impact of contamination on the posterior density is uniformly bounded—both over the parameter space, and the location of said contamination in the data space. This way of studying the robustness of generalised posteriors was pioneered in Ghosh & Basu (2016), and extended by Matsubara et al. (2022b).

We first need to review a result from Matsubara et al. (2022a) which states conditions for a discrepancy measure $\mathcal{D}(\theta)$ in order to prove that the corresponding generalised Bayes posterior $\pi_\omega^{\mathcal{D}}$ is globally robust to outliers. As in the original results, we will state our findings in terms of distributions $\mathbb{P}\in\mathcal{P}(\mathcal{X})$, where $\mathcal{P}(\mathcal{X})$ denotes the set of probability distributions on $\mathcal{X}$.

$$\pi_\omega^{\mathcal{D}}(\theta|\mathbb{P}) \propto \pi(\theta)\exp\{-\omega T\cdot\mathcal{D}(\theta;\mathbb{P})\} \qquad \text{for} \qquad \mathcal{D}(\theta;\mathbb{P}) = \mathbb{E}_{X\sim\mathbb{P}}[d(\theta,X)]. \tag{6}$$

Here, the discrepancy-based loss $\mathcal{D}(\theta;\mathbb{P}) = \mathbb{E}_{X\sim\mathbb{P}}[d(\theta,X)]$ allows us to recover theoretical posteriors based on averaging some kind of discrepancy $d:\Theta\times\mathcal{X}\to\mathbb{R}$ for any measure $\mathbb{P}$. This makes the results more general and more natural to derive. Note that all derived results apply to the $\mathcal{D}_m$-posterior computed from data points $x_{1:T}$, as we can recover it by considering $d=d_m$, and the corresponding empirical measure $\mathbb{P}_T = \frac{1}{T}\sum_{t=1}^T \delta_{x_t}$. In particular, for $d=d_m$ we have that $\mathcal{D}(\theta;\mathbb{P}_T) = \widehat{\mathcal{D}}_m(\theta)$ so that $\pi_\omega^{\mathcal{D}}(\theta|\mathbb{P}_T) = \pi_\omega^{\mathcal{D}_m}(\theta|x_{1:T})$ as defined in (4).

The original work of Matsubara et al. (2022b) constructed a proof of robustness for posteriors that did not depend on an averaged loss $\mathcal{D}(\theta;\mathbb{P}) = \mathbb{E}_{X\sim\mathbb{P}}[d(\theta,X)]$, and so the conditions they derive do not exploit this averaged form. Instead, they showed in Lemma 5 of their paper that global bias-robustness holds if

$$\sup_{\theta\in\Theta}\sup_{y\in\mathcal{X}}\left|\frac{d}{d\varepsilon}\mathcal{D}(\theta;\mathbb{P}_{\varepsilon,y})|_{\varepsilon=0}(y,\theta,\mathbb{P})\right|\pi(\theta)<\infty, \qquad \text{and} \tag{7}$$

$$\int_\Theta \sup_{y\in\mathcal{X}}\left|\frac{d}{d\varepsilon}\mathcal{D}(\theta;\mathbb{P}_{\varepsilon,y})|_{\varepsilon=0}(y,\theta,\mathbb{P})\right|\pi(\theta)d\theta<\infty. \tag{8}$$

Clearly, we can simplify this further because our loss is an average. As long as the function d over which the loss is averaged is sufficiently regular, the below result shows that we obtain global bias-robustness.

Proposition B.1. *For each $\theta\in\Theta$. Suppose that π is upper bounded over Θ. If there exists a function $\gamma:\Theta\to\mathbb{R}$ such that:*

1. $\sup_{y\in\mathcal{X}}|d(\theta,y)|\le\gamma(\theta)$,
2. $\sup_{\theta\in\Theta}\gamma(\theta)\pi(\theta)<\infty$, *and*
3. $\int_\Theta\gamma(\theta)\pi(\theta)d\theta<\infty$.

Then the posterior influence function $\mathrm{PIF}(y,\theta,\mathbb{P})$ *of* $\pi_\omega^{\mathcal{D}}(\theta|\mathbb{P})$ *defined in* (6) *is bounded over both $\theta\in\Theta$ and $y\in\mathcal{Y}$, so that the* $\pi_\omega^{\mathcal{D}}(\theta|\mathbb{P})$ *is globally robust.*

Proof. As outlined above, we simply have to show that the the above conditions suffice to guarantee (7) and (8). Rewriting the loss function related to the contamination model would be:

$$\mathcal{D}(\theta;\mathbb{P}_{\varepsilon,y}) = \mathbb{E}_{x\sim\mathbb{P}_{\varepsilon,y}}[d(\theta,x)] = (1-\varepsilon)\mathbb{E}_{x\sim\mathbb{P}}[d(\theta,x)] + \varepsilon\mathbb{E}_{x\sim\delta_y}[d(\theta,x)].$$

Then, differentiating the last expression w.r.t. ε, and evaluating $\varepsilon=0$, we obtain:

$$\tfrac{d}{d\varepsilon}\mathcal{D}(\theta;\mathbb{P}_{\varepsilon,y})|_{\varepsilon=0} = \mathbb{E}_{x\sim\mathbb{P}}[d(\theta,x)] + \mathbb{E}_{x\sim\delta_y}[d(\theta,x)]$$

Using Jensen's inequality, we bound the expression $|\frac{d}{d\varepsilon}\mathcal{D}(\theta;\mathbb{P}_{\varepsilon,y})|_{\varepsilon=0}|$ as follows:

$$\begin{aligned}
|\tfrac{d}{d\varepsilon}\mathcal{D}(\theta;\mathbb{P}_{\varepsilon,y})|_{\varepsilon=0}| &\le |\mathbb{E}_{x\sim\delta_y}[d(\theta,x)]| + |\mathbb{E}_{x\sim\mathbb{P}}[d(\theta,x)]| \\
&\le \mathbb{E}_{x\sim\delta_y}[|d(\theta,x)|] + \mathbb{E}_{x\sim\mathbb{P}}[|d(\theta,x)|] \\
&= |d(\theta,y)| + \mathbb{E}_{x\sim\mathbb{P}}[|d(\theta,x)|] \\
&\le |d(\theta,y)| + \mathbb{E}_{x\sim\mathbb{P}}[\sup_{y\in\mathcal{X}}|d(\theta,y)|] \\
&= |d(\theta,y)| + \sup_{y\in\mathcal{X}}|d(\theta,y)|,
\end{aligned}$$

and taking a supremum over y we obtain the bound:

$$\sup_{y\in\mathcal{X}} |\tfrac{d}{d\varepsilon}\mathcal{D}(\theta;\mathbb{P}_{\varepsilon,y})| \leq 2\sup_{y\in\mathcal{X}} |d(\theta,y)| \leq 2\gamma(\theta),$$

where the last inequality holds since γ fulfil condition 1. Using this bound, we check the two conditions of Matsubara et al. (2022b).

1. $\sup_{\theta\in\Theta}\sup_{y\in\mathcal{X}} |\frac{d}{d\varepsilon}\mathcal{D}(\theta;\mathbb{P}_{\varepsilon,y})|\pi(\theta) \leq \sup_{\theta\in\Theta} 2\gamma(\theta)\pi(\theta) < \infty$, and
2. $\int_\Theta \sup_{y\in\mathcal{X}} |\frac{d}{d\varepsilon}\mathcal{D}(\theta;\mathbb{P}_{\varepsilon,y})|\pi(\theta)d\theta \leq \int_\Theta 2\gamma(\theta)\pi(\theta)d\theta < \infty$,

where the last inequalities hold because γ meets conditions 2 and 3. Therefore, by virtue of Matsubara et al. (2022b) the posterior is globally bias-robust. ■

B.4. Proof of Proposition 3.2

In this Subsection we provide the proof of Proposition 3.2. The strategy of the proof is simple: We show that $d = d_m$ admits a natural function γ that satisfies the conditions of Proposition B.1.

Proof. From Proposition B.1, it is sufficient to find a function γ such that:

$$\sup_{y\in\mathcal{X}} |\underbrace{\|m^\top(y)\nabla\log p_\theta(y)\|_2^2}_{(1)} + 2\underbrace{\nabla\cdot m(y)m^\top(y)\nabla\log p_\theta(y)}_{(2)}| \leq \gamma(\theta).$$

Now, following from the form of m in Proposition 3.2 and the fact that p_θ is an exponential family member as in (5), we have:

$$(1) = \|m^\top(y)\nabla\log p_\theta(y)\|_2^2 = \sum_{i=1}^d (m^\top(y)\nabla\log p_\theta(y))_i^2 = \sum_{i=1}^d \frac{(\nabla r(x)\theta)_i^2}{1+(\nabla r(x)\theta^\star)_i^2} \leq \sum_{i=1}^d \frac{(\nabla r(x)\theta)_i^2}{(\nabla r(x)\theta^\star)_i^2}.$$

Using the fact that $\|x\|_2^2 \leq \|x\|_1^2 \leq d\|x\|_2^2$ for $x\in\mathbb{R}^d$, we have:

$$\sum_{i=1}^d \frac{(\nabla r(x)\theta)_i^2}{(\nabla r(x)\theta^\star)_i^2} \leq \sum_{i=1}^d \frac{p\|\theta\|_2^2}{\|\theta^\star\|_2^2} = \frac{dp\|\theta\|_2^2}{\|\theta^\star\|_2^2} =: \gamma_1(\theta).$$

For the second expression, we have:

$$\begin{aligned}
(2) = |\nabla\cdot m(y)m^\top(y)\nabla\log p_\theta(y)| &= \left|\sum_{i=1}^d \frac{\partial}{\partial x_t}(m(y)m^\top(y)\nabla\log p_\theta(y))_i\right| \\
&= \left|\sum_{i=1}^d \frac{\partial}{\partial x_t}\left(\frac{(\nabla r(x)\theta)_i}{1+(\nabla r(x)\theta^\star)_i^2}\right)\right| \\
&= \left|\sum_{i=1}^d \frac{(\nabla^2 r(x)\theta)_{ii}(1+(\nabla r(x)\theta^\star)_i^2) - 2(\nabla r(x)\theta^\star)_i(\nabla r(x)\theta)_i(\nabla^2 r(x)\theta^\star)_{ii}}{(1+(\nabla r(x)\theta^\star)_i^2)^2}\right| \\
&\leq \sum_{i=1}^d \left|\frac{(\nabla^2 r(x)\theta)_{ii}}{1+(\nabla r(x)\theta^\star)_i^2)}\right| + 2\left|\frac{(\nabla r(x)\theta^\star)_i(\nabla r(x)\theta)_i(\nabla^2 r(x)\theta^\star)_{ii}}{(1+(\nabla r(x)\theta^\star)_i^2)^2}\right|.
\end{aligned}$$

For most distributions of interest, including Gaussians, exponentials, (inverse) Gamma, and Beta distributions, $\left|\frac{(\nabla^2 r(x)\theta)_{ii}}{1+(\nabla r(x)\theta^\star)_i^2)}\right|$ is bounded for every $\theta\in\Theta$, then:

$$\begin{aligned}
\sum_{i=1}^d \left|\frac{(\nabla^2 r(x)\theta)_{ii}}{1+(\nabla r(x)\theta^\star)_i^2}\right| + 2\left|\frac{(\nabla r(x)\theta^\star)_i(\nabla r(x)\theta)_i(\nabla^2 r(x)\theta^\star)_{ii}}{(1+(\nabla r(x)\theta^\star)_i^2)^2}\right| &\leq dC(\theta) + 2C(\theta)\sum_{i=1}^d \left|\frac{(\nabla r(x)\theta^\star)_i(\nabla r(x)\theta)_i}{(1+(\nabla r(x)\theta^\star)_i^2)^2}\right| \\
&\leq dC(\theta)(1+2d\frac{\|\theta\|_2^2}{\|\theta^\star\|_2^2}) =: \gamma_2(\theta).
\end{aligned}$$

Defining $\gamma(\theta) := \gamma_1(\theta) + \gamma_2(\theta)$ we have :

$$\sup_{y\in\mathcal{X}} \left|\|m^\top(y)\nabla\log p_\theta(y)\|_2^2 + 2\nabla\cdot m(y)m^\top(y)\nabla\log p_\theta(y)\right| \leq \gamma(\theta).$$

Now we are in a position to verify conditions (I) and (II) of Proposition B.1. Since $\gamma(\theta)$ is a polynomial function, $\pi(\theta)$ is a squared exponential prior, and the squared exponential has infinitely many moments, it is clear that:

$$\begin{aligned}
\sup_{\theta\in\Theta}\pi(\theta)\gamma(\theta) &< \infty, \\
\int_\Theta \pi(\theta)\gamma(\theta)d\theta &< \infty,
\end{aligned}$$

which completes the proof. ■

B.5. Boundary and smoothness conditions

In this subsection, we review the boundary and smoothness conditions discussed in Section 3.1, alongside the DSM extension to more general domains $\mathcal{X}$.

The smoothness conditions needed to get the expansion of $\mathcal{D}_m$ as in Equation (3) for $\mathcal{X} = \mathbb{R}^d$ are the following:

Lemma B.2. *If p_θ is twice-differentiable, and $p_0 m m^\top \nabla \log p_\theta, \nabla \cdot (p_0 m m^\top \nabla \log p_\theta) \in L^1(\mathbb{R}^d)$, then we can rewrite $\mathcal{D}_m$ as in Equation* (3).

Yu et al. (2019) extended the DSM to densities with non-negative support, i.e. $\mathcal{X} = \mathbb{R}^d_{\geq 0}$.

Theorem B.3. *Suppose that $\log p_0(x)$ and m are continuously differentiable almost everywhere on $\mathbb{R}^d_+$ and $\log p_\theta$ is twice continuously differentiable with respect to x on $\mathbb{R}^d_+$. Furthermore, we assume the boundary condition,*

$$\lim_{|x^{(i)}| \to \infty} p_0(x) m_{ii}^2(x) \partial_i \log p_\theta(x) - \lim_{|x^{(i)}| \to 0+} p_0(x) m_{ii}^2(x) \partial_i \log p_\theta(x) = 0, \forall i \in \{1, ..., d\},$$

where x_i is the i-dimension of x. Then, we can rewrite $\mathcal{D}_m$ as in Equation (3)

Later, Liu et al. (2022) extended the DSM to densities with support in a Lipschitz Domain, which, intuitively speaking, are bounded connected open domains whose local boundary is a level set of some Lipschitz function.

Theorem B.4. *Assume $\mathcal{X} \subset \mathbb{R}^d$ is a Lipschitz domain. Suppose $p_0, \partial_i \log p_\theta \in H^1(\mathcal{X})$ and that for any $z \in \partial\mathcal{X}$ it holds that*

$$\lim_{x \to z} p_0(x) m_{ii}^2(x) \partial_i \log p_\theta(x) v_i(z) = 0, \forall i \in \{1, ..., d\},$$

where $x \to z$ takes any point sequence converging to $z \in \partial\mathcal{X}$ into account, $v = (v_1, ..., v_d)$ is the unit outward normal vector on $\partial\mathcal{X}$, and $H^1(\mathcal{X})$ is the Sobolev-Hilbert space. Then, we can rewrite $\mathcal{D}_m$ as in Equation (3).

The Sobolev-Hilbert space is defined as follows:

$$H^1(\mathcal{X}) = \left\{ f \in L^2(\mathcal{X}) \,\middle|\, \|f\|^2_{L^2(\mathcal{X})} + \sum_{i=1}^d \|D_i f\|^2_{L^2(\mathcal{X})} < \infty \right\},$$

where D_i is the weak derivative corresponding to ∂_i and $\|f\|_{L^2(\mathcal{X})} = \sqrt{\int_{\mathcal{X}} |f(x)|^2 dx}$.

Gamma distribution Let p_θ be a gamma distribution, and $\mathcal{X} = \mathbb{R}_+$. Assume p_0 bounded from above and that $\log p_0(x)$ is continuously differentiable almost everywhere on $\mathbb{R}_+$. Then for m as in Proposition 3.2, we have

$$\begin{aligned}
\lim_{|x| \to 0+} p_0(x) m^2(x) \partial \log p_\theta(x) &= \lim_{|x| \to 0+} p_0(x) \tfrac{\nabla r(x)\theta + \nabla b(x)}{1 + (\nabla r(x)\theta^\star)_i^2} \\
&= \lim_{|x| \to 0+} p_0(x) \frac{\frac{\theta_1}{x} - \theta_2}{1 + (\frac{\theta_1^\star}{x} - \theta_2^\star)^2} \\
&= \lim_{|x| \to 0+} p_0(x) \tfrac{x\theta_1 - x^2\theta_2}{x + (\theta_1^\star - x\theta_2^\star)^2} \\
&= 0.
\end{aligned}$$

The last equality holds since p_0 is bounded. Now, for the second boundary condition:

$$\lim_{|x| \to \infty} p_0(x) m^2(x) \partial \log p_\theta(x) = \lim_{|x| \to \infty} p_0(x) \tfrac{\nabla r(x)\theta + \nabla b(x)}{1 + (\nabla r(x)\theta^\star)_i^2} = \lim_{|x| \to \infty} p_0(x) \frac{\frac{\theta_1}{x} - \theta_2}{1 + (\frac{\theta_1^\star}{x} - \theta_2^\star)^2} = 0.$$

The last equality holds since p_0 a density, therefore, $\lim_{|x| \to \infty} p_0(x) = 0$. Then the m proposed in Proposition 3.2 satisfies the boundary conditions in Theorem B.3 for the gamma distribution.

Exponential distribution Let p_θ be a exponential distribution, and $\mathcal{X} = \mathbb{R}_+$. Assume p_0 bounded, such that $\log p_0(x)$ is continuously differentiable almost everywhere on $\mathbb{R}_+$. Then for m such as in Proposition 3.2 :

$$\lim_{|x| \to 0+} p_0(x) m^2(x) \partial \log p_\theta(x) = \lim_{|x| \to 0+} p_0(x) \tfrac{\nabla r(x)\theta + \nabla b(x)}{1 + (\nabla r(x)\theta^\star)_i^2} = \lim_{|x| \to 0+} p_0(x) = \tfrac{\theta}{1 + \theta^{\star 2}} \lim_{|x| \to 0+} p_0(x).$$

The term for the second limit would be similar:

$$\lim_{|x|\to\infty} p_0(x)m^2(x)\partial \log p_\theta(x) = \frac{\theta}{1+\theta^{\star 2}} \lim_{|x|\to\infty} p_0(x).$$

Therefore, the expression in Theorem B.3 looks as follows:

$$\lim_{|x^{(i)}|\to\infty} p_0(x)m_{ii}^2(x)\partial_i \log p_\theta(x) - \lim_{|x^{(i)}|\to 0+} p_0(x)m_{ii}^2(x)\partial_i \log p_\theta(x) = \frac{\theta}{1+\theta^{\star 2}} \left(\lim_{|x|\to\infty} p_0(x) - \lim_{|x|\to 0+} p_0(x)\right).$$

Then the m proposed in Proposition 3.2 satisfies the boundary conditions in Theorem B.3 if

$\lim_{|x|\to\infty} p_0(x) = \lim_{|x|\to 0+} p_0(x)$.

C. Practical Considerations

C.1. Run-length pruning.

A naive implementation of $\mathcal{D}_m$-BOCD would keep a posterior over all possible run-lengths $r_t = \{0, 1 \ldots, t-1\}$, but this would lead to an algorithm with overall complexity $\mathcal{O}(\sum_{t=1}^{T} t(d^2+p^2)) = \mathcal{O}(T^2(d^2+p^2))$ for a time series of length T. To prevent this, authors have proposed to 'prune' the run-length posterior to a constant length (Adams & MacKay, 2007; Fearnhead & Liu, 2007). Here, we follow the most popular strategy (e.g. Adams & MacKay, 2007; Saatçi et al., 2010; Knoblauch & Damoulas, 2018) by keeping only the k most probable run-lengths. For all experiments, we take $k = 50$.

C.2. Choice of m

Throughout, we choose m as per Proposition 3.2, as it ensures robustness—even for certain distributions with boundaries (see Appendix B.5). Regarding $\theta^\star$, we found that $\mathcal{D}_m$-BOCD was not very sensitive to this choice; likely because tuning ω offsets any sensitivity to it. In all experiments, we thus picked $\theta^\star$ as the maximum likelihood estimate computed on the full data set. We note that one known issue with robust CP detection method is that they can experience a latency when it comes to detecting actual CPs. Interestingly, this is not something we observe in our experiments with this choice of m and θ^*.

C.3. Choice of ω

How to choose ω is an important question for generalised Bayesian inference, and has more than one answer (Lyddon et al., 2019; Syring & Martin, 2019; Matsubara et al., 2022a; Bochkina, 2022; Wu & Martin, 2023). Previous methods are computationally expensive, asymptotically motivated, and focus on tuning the learning rate to provide asymptotically correct frequentist coverage. As the computational overhead of these methods is substantial and their asymptotic arguments generally do not apply to the CP setting, we pursue a different strategy: we match the uncertainty of the generalised posterior to that of its standard counterpart on the first $t^\star$ observations of the data stream. To operationalise this, we choose

$$\omega^\star = \arg\min_{\omega>0} \mathrm{KL}\left(\pi_\omega^{\mathcal{D}_m}(\theta|x_{1:t^\star}) \| \pi^{\mathrm{B}}(\theta|x_{1:t^\star})\right).$$

Computing $\omega^\star$ is implemented using automatic differentiation via `jax` (Bradbury et al., 2018). This is possible even if the standard Bayes posterior π^{B} is intractable, since $\pi_\omega^{\mathcal{D}_m}$ has a conjugacy property (see Proposition 3.1). Since the standard Bayes posterior is reliable in the absence of outliers and heterogeneity, this yields reasonable uncertainty quantification if the degree of misspecification is mild at the beginning of the data stream. Our experiments confirm this: the uncertainty is well-calibrated, both predictively and with regards to the run-length posterior.

D. Additional Details on Numerical Experiments

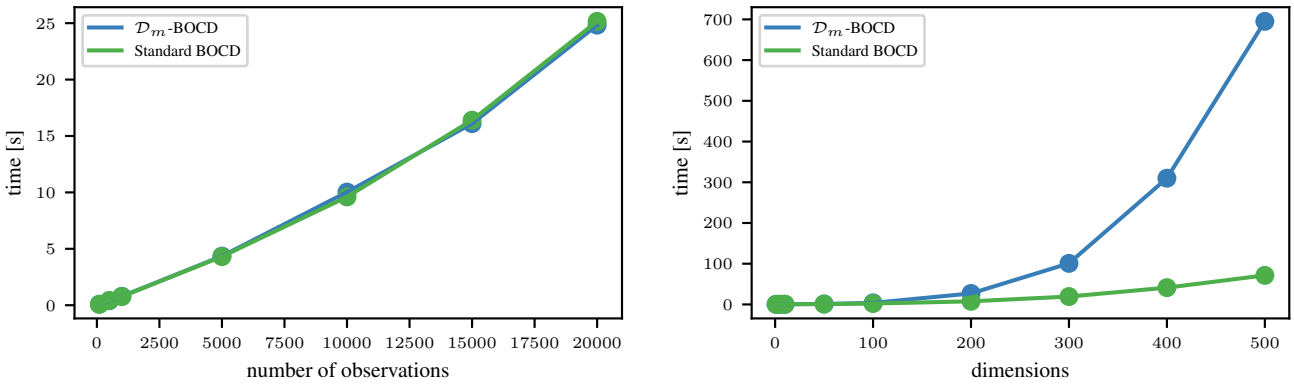
In this section we give additional details on the numerical experiments of Section 4. We provide the exact prior and ω used in each experiment. Moreover, we present an extra numerical experiment to compare the computational complexity of standard BOCD and $\mathcal{D}_m$-BOCD in Appendix D.1.

D.1. Computational complexity

The computational complexity of both standard BOCD and $\mathcal{D}_m$-BOCD is linear in the number of data points. To verify that this theoretical complexity mirrors the practical computational overhead, we generate samples from a Gaussian distribution with 1 CP where the mean varies. We vary the sample size from $T = 100$ up to $T = 20000$. We fit a Gaussian distribution

with the correct variance taken as fixed in both the $\mathcal{D}_m$-BOCD and the standard BOCD. As shown in Figure 5a, both methods are equally fast for any number of observations.

Although the computational complexity of $\mathcal{D}_m$-BOCD is linear in the data, it is quadratic in the dimension of the observations, in particular, is $\mathcal{O}(T(p^2 + d^2))$. To observe this in practice, we consider the same settings as before, but now we fix the sample size to $T = 100$ and vary the data dimensions from $d = 1$ up to $d = 500$. Figure 5b shows that both methods take practically the same time when d is less than 100.



(a) Overall time in seconds versus number of observations. We observe that both methods are equally fast for any number of observations.

(b) Overall time in seconds versus the dimension of the observations. We observe that both methods are practically equally fast when the dimension of the observations is less than 100.

Figure 5. Comparison between $\mathcal{D}_m$-BOCD and the standard BOCD. **blue** line for $\mathcal{D}_m$-BOCD, and **green** line for standard BOCD.

The previous setting is where $\mathcal{D}_m$-BOCD clearly shows its advantage over the β-BOCD due to its completely closed form, making it nearly equivalent, in terms of computational overhead, to standard BOCD. In more complex settings, the posterior predictive may not be available in closed form; hence, we approximate it by sampling from $\pi_\omega^{\mathcal{D}_m}$. It might not be obvious how this is a significant advantage over the β-BOCD framework. To demonstrate that $\mathcal{D}_m$-BOCD is faster than β-BOCD, we generate samples from a Gaussian distribution with 1 CP, where the mean and the variance change. We vary the sample size from $T = 100$ to $T = 20000$ and fit a Gaussian distribution in the $\mathcal{D}_m$-BOCD, β-BOCD and the standard BOCD. In Figure 6, we observe that although the standard BOCD is faster than $\mathcal{D}_m$-BOCD, our method is considerably faster than the β-BOCD for any number of observations.

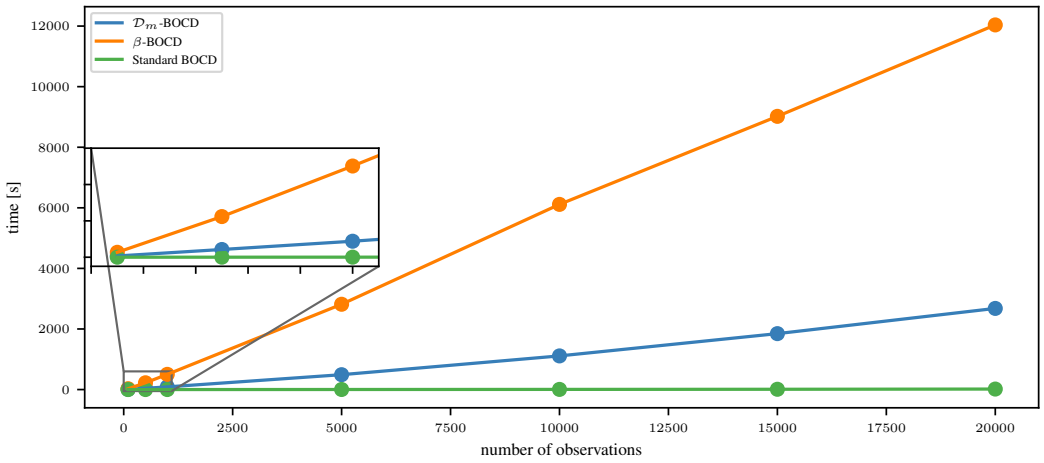


Figure 6. Overall time in seconds versus the number of observations. **blue** line for $\mathcal{D}_m$-BOCD, **orange** line for β-BOCD, and **green** line for standard BOCD. We observe that our method is faster than β-BOCD but is slower than the standard BOCD.

D.2. Varying m, k, and outliers intensity.

We have demonstrated that our method is insensitive to the scale of the outliers. To do so, we compare the performance of the standard BOCD with the $\mathcal{D}_m$-BOCD in different contaminated datasets: while everything else stays the same, we scale the contamination points. We generate 600 samples with 6 CPs at $T \in \{200, 400\}$, and 3 contaminated point at $T \in \{100, 300, 500\}$. We contaminate the data by adding (or subtracting) 0, 5, 10, 20, and 40. For the $\mathcal{D}_m$-BOCD, we choose two different m functions: m as proposed to assure robustness, and $m = I_d$. In Figure 7, we observe that both Standard BOCD and $\mathcal{D}_m$-BOCD with $m = I_d$ mistakenly label outliers as CPs, while $\mathcal{D}_m$-BOCD with m robust is robust and identifies lasting changes.

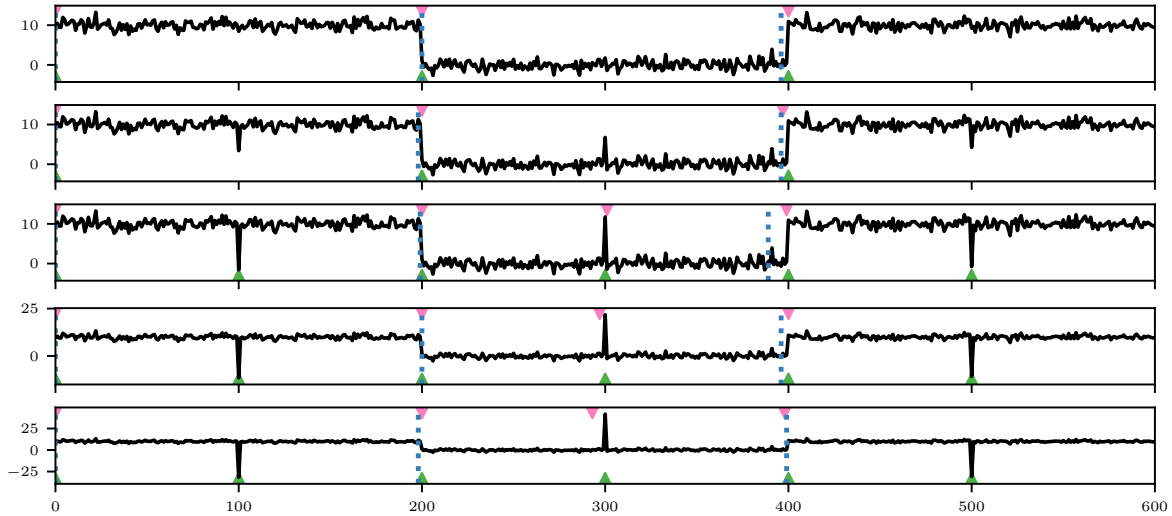


Figure 7. Contamination point scaled by 0, 5, 10, 20, and 40, from top to bottom. MAP segmentation indicated by **blue** dashed lines for $\mathcal{D}_m$-BOCD with m robust, ▼ for $\mathcal{D}_m$-BOCD with $m = I_d$, and ▲ for standard BOCD. Both Standard BOCD and $\mathcal{D}_m$-BOCD with $m = I_d$ mistakenly label outliers as CPs, while DSM_m-BOCD with m robust is robust and identifies lasting changes.

Finally, for contamination scaled by 10, we compared the performance and wall-clock time for $k \in \{1, 50, 600\}$. In Figure 8, we observe that for $k = 1$ the method cannot detect any changepoint, while for $k = 50$ and $k = 600$, the method successfully identifies the changepoints. However, in Table 2 for $k = 600$, the method takes 6x more time. As we discussed in the main paper, k is a parameter in all variants of BOCD and will make all algorithms more expensive if increased.

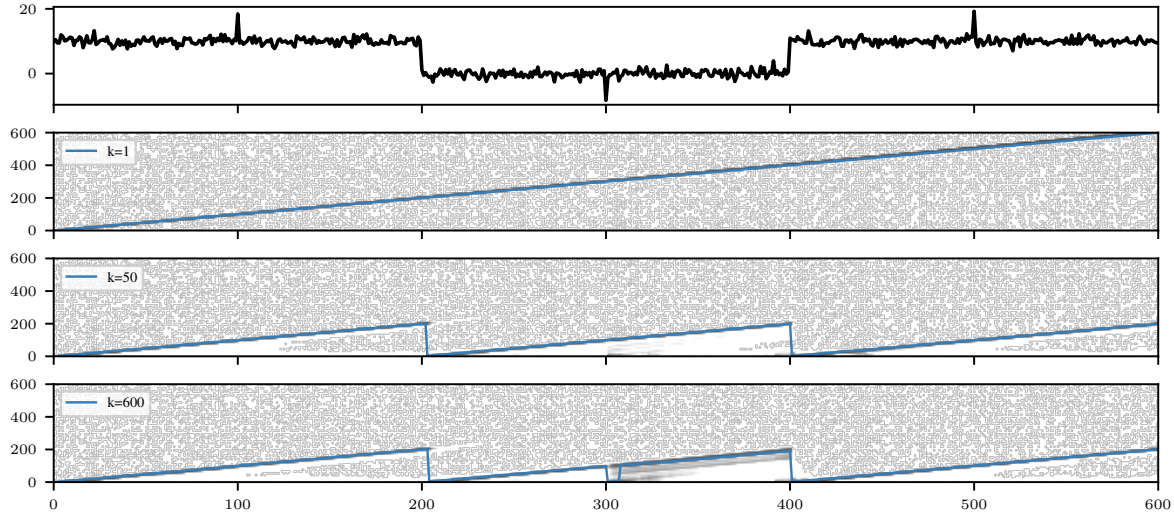


Figure 8. MAP segmentation indicated by **blue** dashed lines for $\mathcal{D}_m$-BOCD with m robust. For $k = 1$ the method cannot detect any changepoint, while for $k = 50$ and $k = 600$, the method successfully identifies the changepoints

k	time [s]
1	**1**
50	38
600	236

Table 2. Wall-clock time for varying k

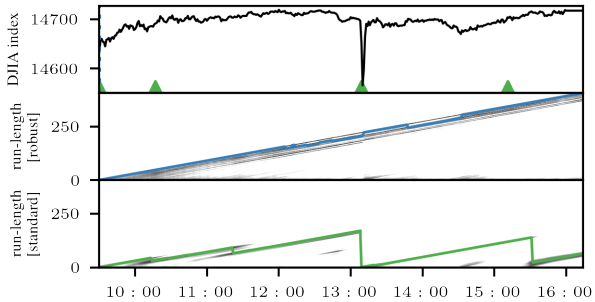


Figure 9. Twitter Flash Crash. The run-length is the time since the last changepoint (CP). *Top:* Jow Dones Index with Maximum a posteriori CPs detected by standard BOCD marked as ▲. *Middle & Bottom:* run-length posteriors of $\mathcal{D}_m$-BOCD with most likely run-length in **blue** and of standard BOCD in **green**. Standard BOCD incorrectly detects a CP, $\mathcal{D}_m$-BOCD does not.

D.3. Accuracy and detection delay

We quantify the method's performance advantage by comparing detection delay and accuracy on artificially generated data with outliers. In particular, We generate 600 samples with 2% of outliers and 6 CPs; then, we report the positive predictive value (PPV), true positive rate (TPR), and the detection delays. These metrics are given by

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \qquad \text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

where TP, FP, and FN are true positives, false positives, and false negatives, respectively. We say the method detects a TP changepoint when the changepoint detected for the method is in a small neighbourhood of the true CP. In order to measure how far the predicted CP is from the true CP, we measure the time difference between both and call it detection delay. For both PPV and TPR, the nearest to 1, the better. For delays, the lower, the better. We run the experiment 10 times, varying the position of the outliers. The following table shows the mean and standard deviation for each metric:

Method	PPV	TPR	Delays
$\mathcal{D}_m$-BOCD	**0.907±0.154**	**0.883±0.13**	1.643±0.475
Standard BOCD	0.6±0.128	0.833±0.149	**1.05±1.545**

Table 3. Performance indices-mean and standard deviation-using the positive predictive value (PPV), true positive rate (TPR), and the detection delays over 10 realisations. For PPV and TPR, the nearest to 1, the better. For delays, the lower, the better.

In Table 3, we observe $\mathcal{D}_m$-BOCD has a significantly better performance with respect to PPV, meaning that the rate of false positives is lower than standard BOCD. This is a consequence of the robustness of our method. Moreover, we see a similar result concerning TPR, which measures the amount of changepoint not detected for the methods. Overall, this means that our method detects the same amount of TP as the standard BOCD while not detecting many FP, showing the strength of our method. Lastly, the detection delay shows that in spite of being robust to outliers, $\mathcal{D}_m$-BOCD does not cause any delay in the detection of CP.

D.4. Twitter flash crash & Cryptocrash

A robust CP detection algorithm must not to be fooled by outliers while detecting CP correctly. We show that $\mathcal{D}_m$-BOCD has this capability on two real-world examples: the first is the Dow Jones Industrial Average (DJIA) index every minute on 17/04/2013, the day of the *Twitter flash crash*. The data is publicly available on FirstRate Data.[2] That day, the Associated

[2]https://firstratedata.com/free-intraday-data

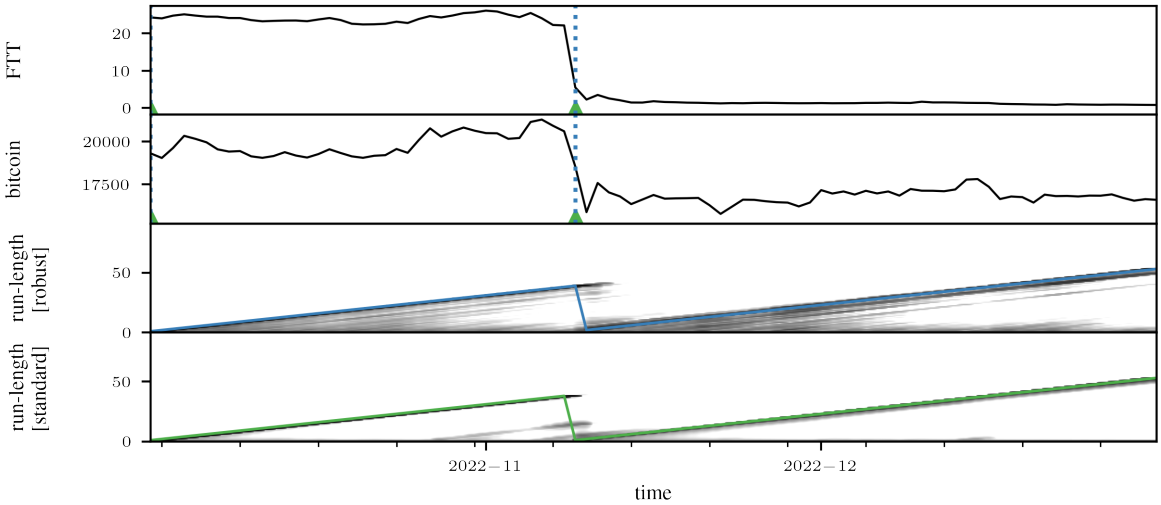


Figure 10. Maximum a posteriori (MAP) segmentation. In **blue** $\mathcal{D}$-BOCD segmentation using dashed lines. In **green** Standard BOCD segmentation using ▲ marks. In addition we plot the run-length posteriors of robust $\mathcal{D}_m$-BOCD algorithm with most likely run-length in **blue** and of standard BOCD in **green**. We observe that robustness does not lead to increased CP detection latency: both methods detect the CP at the same time.

Press' Twitter account was hacked and falsely tweeted that explosions at the White House had injured then-president Barack Obama. In response, the DJIA dropped by 150 points in a matter of seconds before bouncing back. As Figure 9 shows, this is a clear outlier. Modelling the time series with a Gaussian, the plot shows that $\mathcal{D}_m$-BOCD successfully ignores this blip, while standard BOCD incorrectly labels it as a CP. The second example tracks the average daily value of FTT and Bitcoin between 10/2022 and 12/2022, data which is publicly available on Yahoo finance.[3] FTT was the token issued by FTX, one of the biggest crypto-exchanges before it failed due to a liquidity crisis on November 11th 2022. The ensuing collapse of FTX marked a crash in the value of various crypto-currencies, including Bitcoin. Using a two-dimensional Gaussian distribution for both $\mathcal{D}_m$-BOCD and standard BOCD, Figure 10 shows that both methods correctly detect the CP. Figure 10 in Appendix D also displays the run-length posteriors, and shows that robustness does not lead to increased CP detection latency.

In the Twitter flash crash experiment, we model the data with a Gaussian distribution, modelling its natural parameters. For $\mathcal{D}_m$-BOCD, we use a conjugate squared exponential prior r with parameters $\mu = (0, 1)^\top$, and Σ a diagonal matrix so that $\mathrm{diag}(\Sigma) = (10, 1)$ for the natural parameters of said Gaussian. For standard BOCD, we use a Normal-inverse gamma prior with parameters $\mu_0 = 0$, $\nu = 1$, $\alpha = 2$ and $\beta = 10$. We use the first 50 observations to select ω as in Section 3.4, with an obtained value of $\omega^\star \approx 0.0001$

In the Cryptocrash experiment, we model the data with a multivariate Gaussian distribution modelling its natural parameters, and use conjugate squared exponential prior with parameters $\mu = (0, 1, 0, 1)^\top$, and Σ diagonal matrix such that $\mathrm{diag}(\Sigma) = (2, 1, 2, 1)$ for the $\mathcal{D}_m$-BOCD. For standard BOCD, we model mean and variance instead, and use a normal-inverse-Wishart prior with parameters $\nu = 0$, $\kappa = 1$, $\mu = (0, 0)$ and Ψ diagonal matrix such that $\mathrm{diag}(\Psi) = (1, 1)$. we manually fix $\omega = 0.01$. Figure 10 shows the run-length posteriors and the Maximum A Posteriori (MAP) segmentations produced by each method. Since the most likely run-lengths are virtually identical, the below plot also shows that in spite of being robust to outliers, $\mathcal{D}_m$-BOCD does not cause any delay in the detection of CPs.

D.5. Well-log

We model the data with a Gaussian distribution, modelling its natural parameters and using a conjugate squared exponential prior with parameters $\mu = (0, 10)^\top$, and Σ diagonal matrix such that $\mathrm{diag}(\Sigma) = (100, 100)$. Figure 11 shows the run-length and the segmentation of each method. We use the first 200 observations to select ω as in Section 3.4, with an obtained value of $\omega^\star \approx 0.0004$

[3] https://finance.yahoo.com/

D.6. Multivariate synthetic data

We generate 1000 samples from a time series with CPs at t = 250, 750. Conditional on the CPs, the data is generated independently from an exponential in the first a and Gaussian distribution in the second dimension. Since both dimensions are exponential family members, their joint distribution is too. On the natural parameters of this joint distribution, we place a conjugate squared exponential prior with parameters $\mu = (1, 0, 0.5)^\top$, and Σ diagonal matrix such that $\mathrm{diag}(\Sigma) = (1, 1, 0.2)$. As we do not compare against BOCD on this data set, we simply fix $\omega^\star = 0.15$.

D.7. UK 10 year government bond yield

We model the data with a Gamma distribution and use a conjugate squared exponential prior with parameters $\mu = (0, 1)^\top$, and Σ diagonal matrix such that $\mathrm{diag}(\Sigma) = (50, 3)$ on its natural parameters. We use the first 100 observations to select ω as in 3.4. The obtained value is $\omega^\star \approx 0.05$.

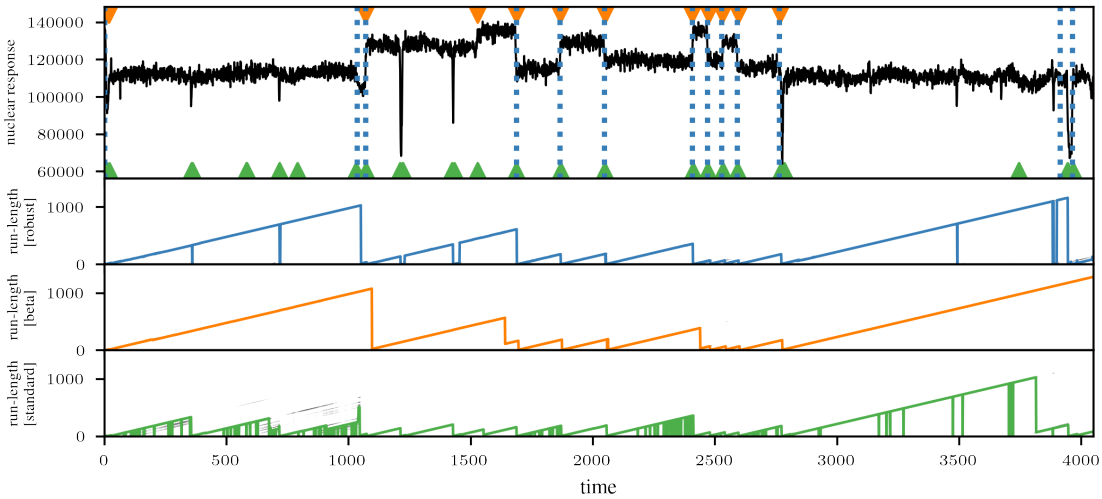


Figure 11. AP segmentation for the well-log indicated by **blue** dashed lines for $\mathcal{D}_m$-BOCD, ▼ for β-BOCD, and ▲ for standard BOCD. In addition we plot the run-length posteriors of robust $\mathcal{D}_m$-BOCD algorithm with most likely run-length in **blue**, β-BOCD in **orange**, and of standard BOCD in **green**. We observe that our method is more robust to outliers than the standard BOCD, but is more sensitive than β-BOCD.