Modern Changepoint Analysis

Siddharth Chandrappa

Contents

Introduction **3**

Section 1 **6**

Section 2 : Background - The offline case: inspect and MissInspect **8**
The inspect algorithm 8
The Missinspect algorithm 11

Section 3: The online problem: $'ocd'$ **and** $'MissOcd'$ **12**
Modifications for missingness 15

Section 4: Theoretical analysis **16**
Bounds on the patience 17
Bounds on the response delay 20

Section 5: Numerical Simulations **27**
MCAR missingness 27
Non-MCAR missingness 31

Conclusion and Outlook **33**

Appendices **34**
Code Listings 34

Bibliography **39**

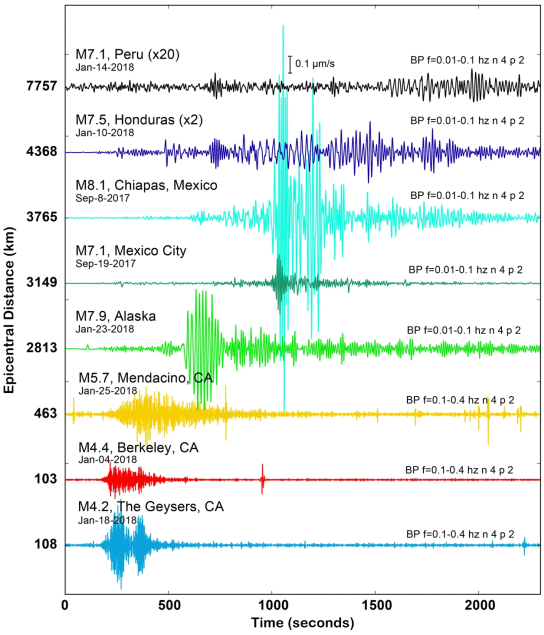
Introduction

With the advent of the information age and the constant advance of modern technology, the necessity for data collection has grown exponentially. When analysing such data, especially in the context of time series, there may be abrupt changes in the data generation process describing the object of interest. These points are collectively called changepoints. There are many natural questions to ask about these changepoints: how and when does such a changepoint occur, did more than one changepoint occur, what implications does the changepoint have on the underlying object and how sure are we that a changepoint did indeed occur? To fully understand the field, we must be able to provide answers to all these questions. At first glance, this seems a particularly broad description of a problem. Indeed, such a broad description yields an equally diverse set of problems and applications, each requiring novel algorithms and consequently, changepoint analysis is a highly active and growing field of research.

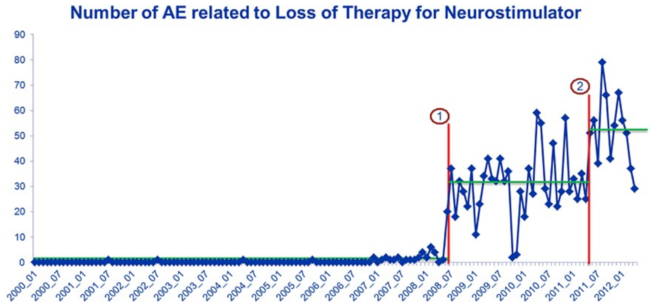
A natural dichotomy that has developed in the field is that of offline and online changepoint detection.

In offline changepoint detection, the entire data set is observed and only then is any attempt made to identify changepoints. The need for this kind of analysis is ubiquitous in medicine, for example, in determining the effectiveness of COVID-19 containment policies [1] and in identifying changes in genome sequences[2]; in climate change research, by identifying sudden changes in global temperature[3] and examples in finance where we analyse historical events that cause sudden changes in stock market prices[4].

In online changepoint analysis, the aim is to detect changepoints as soon as possible after they occur. Applications of online changepoint problems are equally rich, ranging from home sensor data [5]; in robotics, when tackling robot vision problems by analysing changes in object articulation properties[6] and in networks by early detection of Denial of Service (DoS) attacks [7]. Observe Figure 1 for a graphical illustration of examples of two different changepoint problems.



(a) Plots of earthquakes recorded by the Sacramento Dark Fiber DAS array, image taken from [8]



(b) Plot illustrating data collected for a postmarket safety surveillance. The number of AEs (adverse events) due to a loss of therapy from the neurostimulator is shown against time. Data retrieved from the FDA MAUDE database and image is taken from [9]

Figure 1: Plots illustrating examples of offline and online changepoint detection problems

In Figure 1(a), we would like to identify the earthquake as soon as possible after it occurs and so this would be an online changepoint problem. For Figure 1(b), the analysis takes place after data collection of adverse events and is thus an offline changepoint problem.

Another novelty that arises when collecting such large quantities of data in real-time is the phenomenon of missing data. It is not reasonable to expect that data measurement and transfer mechanisms function perfectly and it is often the case that missingness may occur between subsequent iterations. A prominent example is the UDP data transfer protocol, commonly used in data transmission for audio and video transmission when transmitting large quantities of data. Here, the data packet transfer mechanism often leads to packet loss at the host [10] and lots of techniques have been developed to mitigate some of the impacts of this.

Our problem explores missingness within the realm of changepoint analysis, specifically, we attempt to solve a high-dimensional online changepoint problem with missingness. Moreover, our method will be robust enough to provide additional benefits when the changepoint occurs over a sparse set of co-ordinates.

Firstly, we need to carefully define the parameters of our problem. For an online algorithm, we use the definition provided by [11], namely that such an algorithm must have a space complexity that is constant in time, and in particular is not dependent on the number of previous observations. Specifically, quoting [11], any data processing and analysis must be a function of the "number of bits needed to represent the new observation".

The primary model for missingness we consider follows a missing completely at random (MCAR) model; see [12]. Specifically, we will investigate, both theoretically and empirically, the case of row homogeneous missingness, whereby the probability that a given co-ordinate has missing data remains constant in time, but this probability may vary between rows. In particular, it becomes possible to represent missingness through a 'missingness vector' $q = (q_1, q_2, \ldots, q_p)$ where q_i denotes the probability that the 'i^{th}' co-ordinate is missing and p denotes the number of co-ordinates. We will also briefly investigate empirically a non-MCAR model, to illustrate Londschien et al.'s [13] observation that imputation methods perform poorly when missingness follows particular structures by having a higher proclivity to flagging false positives. The example they give and the one we investigate is a simultaneous missingness across multiple co-ordinates due to temporary sensor failure.

The main focus of the essay is our proposal of a new algorithm called '*MissOcd*' to deal with missingness in the above settings. *MissOcd* is based on earlier work by Chen, Wang and Samworth [11], who developed the '*ocd*' (short for online changepoint detection) algorithm to deal with the problem in the absence of missingness. In essence, *ocd* can be thought as the combined effort of:

1. detecting changes concentrated in a single co-ordinate, through likelihood ratio tests across putative changepoints

2. detecting denser changes by aggregating test statistics over several co-ordinates, thereby 'borrowing detection strength' by examining the entire data set

'*MissOcd*' aims to adhere to the same general framework as *ocd* but acts as an extension of it to allow for missing data inputs. A more detailed description of both algorithms will be given in later sections.

This essay will have the following structure. In section 1, we describe formally our problem, giving the data generation process as well as the nature of missingness. We also define relevant quantities of interest that help determine the efficacy of a given method, namely the *patience*, *worst-case response delay* and the *average response delay.* In section 2 we provide background by examining prior work on the offline version of our problem. Specifically, we analyse the *inspect* [14] and *missinspect* algorithms [15]. In section 3, we provide a summary of the ocd algorithm before introducing our proposal '*MissOcd*'. In section 4, we provide theoretical guarantees for $MissOcd'$, which, like *ocd′* referenced in [11], is a slight modification to the original algorithm which yields an easier theoretical analysis. The results we present could be considered as the corresponding bounds to the theoretical bounds for certain quantities of interest as outlined in [11]. In section 5, we run numerical simulations, with comparisons alongside a natural competitor which uses a sliding window data imputation method to handle missingness. Here, we also consider another form of missingness to further emphasise the robustness of '*MissOcd*' and the shortcomings that imputation methods can have. Section 6 is the concluding chapter summarising our work. We also include potential future avenues of research to further extend this work. Section 7 is the appendix which includes our modified code. Section 8 is the bibliography containing all the references we use.

Section 1

We begin by outlining the data generation process, first without missingness and then explain how missingness fits in. Our observed data at time t is denoted as X_t and is distributed as follows:

$$\text{Let } X_t \sim \mathcal{N}(\mu_t, I_p),\ i = 0, 1, 2, \ldots$$
$$\text{where } \mu_t = \begin{cases} \mu^0 & t = 1, 2, \ldots, z \\ \mu^1 & t > z \end{cases}$$

We make a further assumption that there is no temporal dependence so the sequence $X_1, X_2, \ldots$ is independent. Furthermore, $z \in \mathbb{N}^+$ denotes a changepoint when $\mu^1 \neq \mu^0$.

To implement missingness, as mentioned earlier, we have a 'missingness vector' $\mathbf{q} = (\mathbf{q_1}, \mathbf{q_2}, \ldots, \mathbf{q_p})$ such that independently of the prior data generation process, $(X_t)_i$ has a probability q_i of being missing and its value will be given as 'NA'. This probability is also independent of other co-ordinates being missing. We will also work with the 'non-missingness vector' $\mathbf{p} = (\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_p})$ which denotes the probability that a given co-ordinate is not missing, so $q_i = 1 - p_i$. Lastly, we will restrict ourselves to the case where none of the $q_1 = 1$, since this co-ordinate could then be ignored.

We now include a table containing useful terminology and definitions alongside relevant remarks about these for ease of reference, particularly when examining proofs of our theoretical results. Most of these are similar to those found in p.3 of the original paper[11].

Quantity of interest	**Definition**	**Remarks**
θ	$\mu^0 - \mu^1$	This is in general not known. However, we do assume that μ^0 is known. This is a natural assumption since, for example, in the case of a weather event, such as a hurricane, constant monitoring prior to the event will give a good estimate of μ^0, but μ^1 will remain unknown. Henceforth, we work under $\mu^0 = 0$ and $\mu^1 = \theta$.
$\mathbb{P}_{z,\theta}$ and $\mathbb{E}_{z,\theta}$	$\mathbb{P}_{z,\theta}$,θ denotes the joint distribution of X (and $\mathbb{E}_{z,\theta}$ the corresponding expectation)	This allows us to use the notation $\mathbb{P}_{z,0}$, or simply $\mathbb{P}_0$ to denote the distribution when there is no changepoint (we call this the null distribution).
ν,β	$\nu = \|\|\theta\|\|_2$ with the condition that $\nu \geq \beta > 0$	An assumption on $\|\|\theta\|\|_2$ is required since the efficacy of our method will be tied to the signal strength and such a term would be expected in theoretical bounds on the detection delay.
Patience	$\mathbb{E}_{z,0}(N)$	This can be thought of as the average run-length under the null. A control on the patience would control our false-positive detection rate.
Average Response Delay	$\sup_{z\in\mathbb{N}} \mathbb{E}_{z,\theta}((N-z) \vee 0)$	
Worst-case response Delay	$\sup_{z\in\mathbb{N}} ess \sup \mathbb{E}_{z,\theta}((N - z) \vee 0 \| X_1, X_2, \dots X_z)$	This, along with the average response delay measures the effectiveness of our method at detecting a changepoint.
Sparsity of signal	One way of capturing this is $\|\|\theta\|\|_0$	Later, we will define another notion of sparsity, for our theoretical results, which will be related to this natural understanding of sparsity.

We remark that any viable method should be able to control both the response delay and the patience. We can think of the former condition as our method having power to detect a changepoint in the first place and the latter as our method not falsely detecting changepoints under the null.

Section 2 : Background - The offline case: inspect and MissInspect

We will first provide background by investigating and summarising prior work on the corresponding offline problem, specifically, we focus on the inspect[14] and missinspect[15] algorithms which address exactly this. Specifically, as the name suggests, *missinspect* and *inspect* tackle the cases with and without missingness respectively. It is also of interest to compare the methodologies used here with those that will be proposed for the online problem. Throughout the section, I aim to provide additional intuition and explain the motivation behind the algorithms.

The inspect algorithm

We note that the data generation process in the inspect algorithm [14] is the same as in our case (for both the missing and non-missing data cases), with the additions, however, that multiple changepoints are permitted and indeed can be detected. We will also have a finite number of samples, taken to be n.

We will now consider case when there is only one changepoint and will briefly mention how this can be extended to the case of multiple change points.

The field of offline univariate changepoint detection is well studied and enjoys a rich body of literature dating back to the 1950s. So one way to approach this problem would be to somehow leverage this to our benefit, in essence to answer the question "How might we reduce our problem to the problem of detecting a changepoint on univariate time series data?"

The informative sparse projection for estimation of change points (inspect) algorithm helps answer this. It can be thought of as a two-step process, the first step, as its name suggests involves an "informative" projection step; here, the high-dimensional data is projected onto a one-dimensional subspace whilst amplifying (in a way we will make precise) the pre and post-changepoint data. We then use an existing one-dimensional changepoint method on this "informative" univariate time series.

Specifically, given data $X_1, X_2, \dots X_n$ and a unit projection direction '**a**', the distribution for the data $a^T X_i$ is :

$$a^T X_i \sim \begin{cases} \mathcal{N}(0,1) & i \leq z \\ \mathcal{N}(a^t\theta, 1) & i > z \end{cases}$$

It would be natural to select **a** to maximise the difference in the means, $a^t\theta$, which, under the constraint $||a||_2 = 1$, is achieved by taking $a = \dfrac{\theta}{||\theta||_2}$. This of course depends on

θ, which, in general, is unknown. So the problem becomes obtaining an estimate of this direction using the data.

Ideally, our method should be able to exploit both the sparsity in the signal and the temporal structure of the data; this leads to the well-known CUSUM transformation. Here, we apply a univariate CUSUM transformation row-by-row before exploiting sparsity. The CUSUM transformation is defined as:

$$\mathcal{T}_{p,n} : \mathbb{R}^{n \times p} \to \mathbb{R}^{n \times (p-1)}$$

$$(\mathcal{T}_{p,n}(M))_{j,t} := \sqrt{\frac{t(n-t)}{n}} \left(\underbrace{\frac{1}{n-t} \sum_{r=t+1}^{n} M_{j,r}}_{\text{mean of data after putative changepoint}} - \underbrace{\frac{1}{t} \sum_{r=1}^{t} M_{j,r}}_{\text{mean of data prior to putative changepoint}} \right)$$

We remark that the co-ordinate $\mathcal{T}_{j,t}(M)$ can alternately be characterised as a generalised likelihood ratio test on the j^{th} co-ordinate of the null that μ^0 for all times against the alternate that $\mu^0 = 0$ for $t \leq z$ and $\mu^1 \neq 0$ for $t > z$. So, $\mathcal{T}_{p,n}(M)_{j,t}$ represents the generalised likelihood ratio statistic in the j^{th} co-ordinate at a putative changepoint time t.

The features of the CUSUM transformation that make it particularly useful are linearity in the input matrix M and that it respects and makes use of the temporal structure of the data.

Importantly, linearity allows us to form the decomposition

$$T = A + E$$

where $T = \mathcal{T}(X), A = \mathcal{T}(\mu)$ and $T = \mathcal{T}(\epsilon)$

Where X, μ, ϵ are the matrices formed whose columns are $(X_1, X_2, \ldots, X_n)$, $(\mu_1, \mu_2, \ldots, \mu_n)$ and $\epsilon_1, \epsilon_2, \ldots, \epsilon_n \overset{\text{iid}}{\sim} \mathcal{N}(0, I_p)$, so T can be thought of as the CUSUM transformation of the mean perturbed by E.

In the single changepoint case, explicit calculation of A reveals that

$$A = \theta v^T$$

for a particular vector $v = v(n, z)$ (this can be written explicitly although this is not required for our discussion).

Consequently, θ is the leading left singular vector of A so to recover this, we seek

$$\boxed{\underset{x \in S^{p-1}(k)}{\operatorname{argmax}} ||T^T x||_2}$$

However, as the authors note, solving this problem is thought to be NP-hard due to the sparsity condition, so we might wish to simplify it by relaxing some of the assumptions.

So, note:

$$\begin{aligned}
\underset{x\in S^{p-1}(k)}{\text{argmax}}\,||T^Tx||_2 &= \underset{x\in S^{p-1}(k),y\in S^{n-2}}{\text{argmax}}\; x^TTy \\
&= \underset{x\in S^{p-1}(k),y\in S^{n-2}}{\text{argmax}}\; (xy^T)\cdot T \text{ where } \cdot \text{ denotes the trace inner product} \\
&= \underset{M\in\mathcal{M}}{\text{argmax}}\, M\cdot T
\end{aligned}$$

Where

1. $||x||_0 = k \implies M$ has at least k rows which consist entirely of zeros
2. Considering the singular-value decomposition of $xy^T = UDV^T \implies (xy^T)^2 = xx^T = UD^2U^T$ so $rank(xy^t) = 1 \iff ||\sigma(M)||_0 = 1$ where $\sigma(M)$ is the vector of non-zero diagonal elements of D, i.e. the singular values
3. Also, since xx^T has rank 1, and x is an eigenvector with eigenvalue 1 since $(xx^T)x = ||x||_2x = x$, from the above calculation we deduce further that $||\sigma(M)||_2 = 1$

So our set

$$\begin{aligned}
\mathcal{M} := \{M \in \mathbb{R}^{p\times(n-1)} :&||\sigma(M)||_0 = 1, ||\sigma(M)||_2 = 1, \\
&M \text{ has at least } k \text{ rows which consist entirely of zeros}\}
\end{aligned}$$

With view of a convex optimisation in mind, we can combine $||\sigma(M)||_0 = 1, ||\sigma(M)||_2 = 1$ and relax the assumptions to form

$$\mathcal{M}_1 = \{M \in \mathbb{R}^{p\times(n-1)} : ||M||_2 \leq 1\}$$

and attach an l_1 penalty on M to account for row sparsity. The resulting problem can now be handled with tools of convex optimisation, moreover the dual optimisation problem is strikingly simple and its solution can be expressed in terms of the soft-thresholding function:

$$(\underset{M\in\mathcal{M}_1}{\text{argmax}}\, M\cdot T - \lambda||M||_1)_{ij} = \frac{SFT(T,\lambda)}{||SFT(T,\lambda)||_2}$$

where $SFT(A,t)_{i,j} = sgn(A_{ij})(|A_{ij}| - \lambda)_+$

Remark: The original paper proposed 2 different relaxation sets, the other was less enlarged, specifically it was

$$\mathcal{M}_2 := \{M \in \mathbb{R}^{p\times(n-1)} : \sum_{i=1}^{\min(p,n-1)} \sigma_i(A) \leq 1\}$$

However solving the dual problem involved using the alternating direction method of multipliers (ADMM) which is less efficient for practical use and the authors recommend using $\mathcal{M}_1$.

Once we have identified the optimising M^*, we proceed with univariate changepoint detection, in particular we take the maximum of the projected CUSUM statistics over all times i.e. we take $\text{argmax}_{t\in\{1,2,\ldots,n-1\}}(|(v^TT)_t|)$ as our changepoint estimate.

The Missinspect algorithm

We now contrast this with the MissInspect algorithm. Here, we proceed similarly, whilst making some necessary modifications.

We first modify the CUSUM transformation with the MissCUSUM transformation :

$$(\mathcal{T}^{miss}_{p,n}(M))_{j,t} := \sqrt{\frac{\tilde{t}_{j,t}(\tilde{n}_{j,t}-\tilde{t}_{j,t})}{\tilde{n}_{j,t}}}\left(\frac{1}{\tilde{n}_{j,t}-\tilde{t}_{j,t}}\sum_{r=t+1}^{n}\tilde{M}_{j,r}-\frac{1}{\tilde{t}_{j,t}}\sum_{r=1}^{t}\tilde{M}_{j,r}\right)$$

where

1. $\tilde{M}$ is the matrix with each missing element (denoted as 'NA') in M is replaced with 0
2. $\tilde{t}_{j,t} = \sum_{r=1}^{t} 1_{\{(X_r)_j \text{ is not missing}\}}$
3. $\tilde{n}_{j,t} = \sum_{r=1}^{n} 1_{\{(X_r)_j \text{ is not missing}\}}$

Writing it this way is particularly revealing and natural since it is essentially applying the same row-by-row likelihood ratio test as before, but simply ignoring the missing data points.

Now, a similar decomposition property of the CUSUM transformation holds as before with the exception that the matrices μ, ϵ have entries corresponding to missing data points. So, we have

$$\mathcal{T}^{miss}_{p,n}(X) = \mathcal{T}^{miss}_{p,n}(\mu) + \mathcal{T}^{miss}_{p,n}(\epsilon)$$

As before, we would want analyse the leading left singular vector of $A_{miss} := \mathcal{T}^{miss}_{p,n}(\mu)$. However, since A^{miss} is no longer rank 1, too much information is lost if we rely on our previous relaxation.

Instead, it can be shown that A_{miss} can be approximated by the rank 1 approximation $(\theta \circ \sqrt{q})v^T$, with $v = v(n, z)$, where $\circ$ denotes the Hadamard product, so the projection direction we seek in this case would be $(\theta \circ \sqrt{q})$. Here, a different relaxation optimisation problem is used instead, which can be solved in a similar vein as before. Specifically the problem is:

$$\underset{x\in\mathcal{X},y\in\mathcal{Y}}{\text{argmax}}\{(xy^T)\cdot\mathcal{T}^{miss}_{p,n}(X) - \lambda||x||_1\}$$

with $\mathcal{X} = \{x \in \mathbb{R}^p : ||x||_2 \leq 1\}$ and $\mathcal{Y} = \{y \in \mathbb{R}^{n-1} : ||x||_2 \leq 1\}$. The penalty term once again controls for the sparsity and $\mathcal{X}, \mathcal{Y}$ represent natural relaxation sets. This is a biconvex optimisation problem, so can once again be solved using existing tools in convex optimisation.

As before, the final univariate changepoint algorithm is done by taking the maximum of the projected CUSUM statistics.

We also not that both Inspect and MissInspect come with theoretical guarantees, both on the angle between the estimated and ideal projection directions and on the location of the changepoint itself.

An important point to note here is the transition from the case without missingness to the case with missingness. The result is a modification of the original algorithm, however the essential fundamentals of the inspect algorithm remain in MissInspect. As we will see, this shares many similarities with our own proposal, which once again naturally builds upon existing methods. Together, they complete the analysis of the problem in both online and offline settings.

Section 3: The online problem: $'ocd'$ and $'MissOcd'$

We first provide a description of the *ocd* algorithm to allow us to understand our proposal for missingness. Indeed, our method is a modified version of the 'ocd' algorithm [11] which is the special case of our problem when q is the zero vector (no-missingness).

To make sense of the algorithm, we ask ourselves, what characteristics must any good algorithm possess? Well for one, we use must be sophisticated enough to perceive changes when all or most of the signal is in one co-ordinate alone. Secondly, such a method must also be able to adapt to the case where the signal is denser and more spread out over several co-ordinates. Here, examining just a single co-ordinate alone would not be powerful enough to quickly detect a changepoint.

With the first objective in mind, $'ocd'$ makes use of the so called 'diagonal statistics'. Here, we assume the signal has size $||\theta||_2 = b \neq 0$ and test against the null that $\mu_0 = 0$ for all times against the alternative that $\mu_t = \begin{cases} 0 & t = 1, 2, \ldots, z \\ b & t > z \end{cases}$. Importantly, the time of the changepoint is not known. We do this by performing a likelihood ratio test of the null against the alternate assuming a changepoint at a time $t = h$ and taking the maximal statistic over all such putative changepoints $h \in \{0, 1, 2, \ldots, n\}$. This leads to test statistic $R_{n,b}$ and potential changepoint locations $t_{n,b}$ defined by:

$$R^j_{n,b} = \max_{0 \leq h \leq n} \sum_{i=n-h+1}^{n} b((X_i)_j - b/2)$$

$$t^j_{n,b} = s \operatorname*{argmax}_{0 \leq h \leq n} \sum_{i=n-h+1}^{n} b((X_i)_j - b/2)$$

Compare this with the CUSUM transformation from the inspect algorithm where we assumed the post-changepoint mean was 'b'. In the inspect algorithm we used a generalised likelihood ratio test. This gets to the key feature of the above statistics, indeed, it can be rewritten inductively as

$$\boxed{R^j_{n,b} = \max\left(R^j_{n-1,b} + b((X_m)_j - b/2), 0\right)}$$

so this can be calculated with an **online algorithm**. On the other hand, if we were instead to take the maximum along rows of the CUSUM statistics as in 'inspect', the awkward rescaling factor which would change with time renders such an online calculation impossible. For the same reason, other common statistics, such as the Shiryayev statistic constitute **sequential algorithms** and are not suitable in our setting.

So, assuming there has indeed been a changepoint, we can think of $t^j_{n,b}$ as the most likely time at which the changepoint occurred if we are to focus strictly on data along the 'j^{th}' co-ordinate alone.

To tackle the second objective, we construct 'off-diagonal statistics'. Here, given a co-ordinate j and its corresponding most likely putative changepoint '$t^j_{n,b}$', we sum the effects of the signal change across the other co-ordinates and then aggregate their normalised effect as follows:

$$A^{j',j}_{n,b} = \sum_{i=n-t^j_{n,b}+1}^{n} (X_i)_{j'}$$

Which upon aggregation:

$$Q^j_b = \sum_{j' \neq j} \frac{(A^{j',j}_{n,b})^2}{\max(t^j_{n,b}, 1)}$$

Essentially, this uses the diagonal statistic to provide an idea for the changepoint location and then forms the statistic borrowing strength across all the different co-ordinates to detect sparser changes.

Of course, since θ is unknown, we will need to perform this for various values of 'b', with the largest value across all co-ordinates lying in $[\nu/\sqrt{p}, \nu]$ (a simple consequence of the pigeonhole principle). Consequently, b is chosen to range through the set $\mathcal{B} \cup \mathcal{B}^0$ (see the pseudocode for the algorithms).

We present the pseudocode for the original '*ocd*' method alongside '*MissOcd*' in Algorithm 1 and Algorithm 2 and we highlight the modifications that we have made. Red denotes new additions to the algorithm and blue modifications to existing parts of the *ocd* algorithm.

Algorithm 1 Pseudocode for *ocd*

Input
1. Observations $X_1, X_2, \cdots \in \mathbb{R}^p$
2. β - recall we assume $||\theta||_2 \geq \beta > 0$
3. sparsity threshold parameter $a \geq 0$
4. Statistic thresholds $T^d > 0, T^o > 0$

Define
1. $b_k = \left\{ \pm \frac{\beta}{\sqrt{2^k \log_2(2p)}} \right\}$
2. $\mathcal{B} = \bigcup_{k=1}^{\lfloor \log_2(p) \rfloor} b_k$
3. $\mathcal{B}^0 = \left\{ b_{1+\lfloor \log_2(p) \rfloor} \right\}$

Initialise
1. $A_b = 0 \in \mathbb{R}^{p \times p}$ for $b \in \mathcal{B} \cup \mathcal{B}^0$
2. $t_b = 0 \in \mathbb{R}^p$ for $b \in \mathcal{B} \cup \mathcal{B}^0$
3. $n = 0$

repeat
  $n \leftarrow n + 1$
  Read in current data X_n
  for $(j, b) \in [1:p] \times (\mathcal{B} \cup \mathcal{B}^0)$ **do**
    $t_b^j \leftarrow t_b^j + 1$
    $A_b^{(\cdot,j)} \leftarrow A_b^{(\cdot,j)} + X_n$
    if $bA_b^{(j,j)} - b^2 \frac{t_b^j}{2} \leq 0$ **then**
      $(A_b^{(\cdot,j)}, t_b^j) \leftarrow (0, 0)$
    end if
    Set $Q_b^j = \sum_{j' \neq j} \frac{(A_b^{j',j})^2 \mathbb{1}\left\{ |A_b^{j',j}| \geq a\sqrt{t_b^j} \right\}}{\max(t_b^j, 1)}$
  end for
  Set $S^d \leftarrow \max_{(j,b) \in [1:p] \times (\mathcal{B} \cup \mathcal{B}^0)} (bA_b^{(j,j)} - b^2 \frac{t_b^j}{2})$
  Set $S^o \leftarrow \max_{(j,b) \in [1:p] \times \mathcal{B}} (Q_b^j)$
until $S^d \geq T^d$ or $S^o \geq T^o$
Return $N = n$

Algorithm 2 Pseudocode for *MissOcd*

Input
1. Observations $X_1, X_2, \cdots \in \mathbb{R}^p$
2. β - recall we assume $||\theta||_2 \geq \beta > 0$
3. sparsity threshold parameter $a > 0$
4. Statistic thresholds $T^d > 0, T^o > 0$

Define
1. $b_k = \left\{ \pm \frac{\beta}{\sqrt{2^k \log_2(2p)}} \right\}$
2. $\mathcal{B} = \bigcup_{k=1}^{\lfloor \log_2(p) \rfloor} b_k$
3. $\mathcal{B}^0 = \left\{ b_{1+\lfloor \log_2(p) \rfloor} \right\}$

Initialise
1. $A_b = 0 \in \mathbb{R}^{p \times p}$ for $b \in \mathcal{B} \cup \mathcal{B}^0$
2. $\tilde{t}_b^{(\cdot,\cdot)} = 0 \in \mathbb{R}^{p \times p}$ for $b \in \mathcal{B} \cup \mathcal{B}^0$
3. $n = 0$

repeat
  $n \leftarrow n + 1$
  Read in current data X_n
  Let $\tilde{X}_n$ be X_n with missing data replaced with 0
  Set $\omega \in \mathbb{R}^p$, $\omega_i = \begin{cases} 1 & (X_n)_i \text{ is not missing} \\ 0 & (X_n)_i \text{ is missing} \end{cases}$
  for $(j, b) \in [1:p] \times (\mathcal{B} \cup \mathcal{B}^0)$ **do**
    $\tilde{t}_b^{(\cdot,j)} \leftarrow \tilde{t}_b^{(\cdot,j)} + \omega$
    $A_b^{(\cdot,j)} \leftarrow A_b^{(\cdot,j)} + \tilde{X}_n$
    if $bA_b^{(j,j)} - b^2 \frac{\tilde{t}_b^{(j,j)}}{2} \leq 0$ **then**
      $(A_b, \tilde{t}_b^{(\cdot,j)}) \leftarrow (0, 0)$
    end if
    Set $Q_b^j = \sum_{j' \neq j} \frac{(A_b^{j',j})^2 \mathbb{1}\left\{ |A_b^{j',j}| \geq a\sqrt{\tilde{t}_b^{(j',j)}} \right\}}{\max(\tilde{t}_b^{j',j}, 1)}$
  end for
  Set $S^d \leftarrow \max_{(j,b) \in [1:p] \times (\mathcal{B} \cup \mathcal{B}^0)} (bA_b^{(j,j)} - b^2 \frac{\tilde{t}_b^{(j,j)}}{2})$
  Set $S^o \leftarrow \max_{(j,b) \in [1:p] \times \mathcal{B}} (Q_b^j)$
until $S^d \geq T^d$ or $S^o \geq T^o$
Return $N = n$

At this stage we also define the theoretical quantities N^d and N^o. Here, we assume the algorithm is allowed to run until both thresholds have been exceeded. The corresponding times when this occurs are N^o, N^d.

Modifications for missingness

The method '*MissOcd*' is a very natural generalisation of the original 'ocd' method. Much like 'missinspect', '*MissOcd*' follows the earlier notions, specifically of 'diagonal' and 'off-diagonal' statistics.

When calculating diagonal statistics, we proceed similarly but ignore missing data, so we would perform the hypothesis tests on a 'contracted data set' on the observed data only. Consequently, we store $\tilde{t}^{(j,j)}_{n,b}$ which represents this position within the 'contracted data set' (the reason for the 2 co-ordinates will become clear when we demonstrate how off-diagonal statistics are calculated). For ease of analysis, we also define a function $t^{(j,j)}_{n,b} := TrueTime(\tilde{t}^{(j,j)}_{n,b})$ that returns the corresponding time when including missing values. This is illustrated in Figure 2 below which demonstrates the procedure:

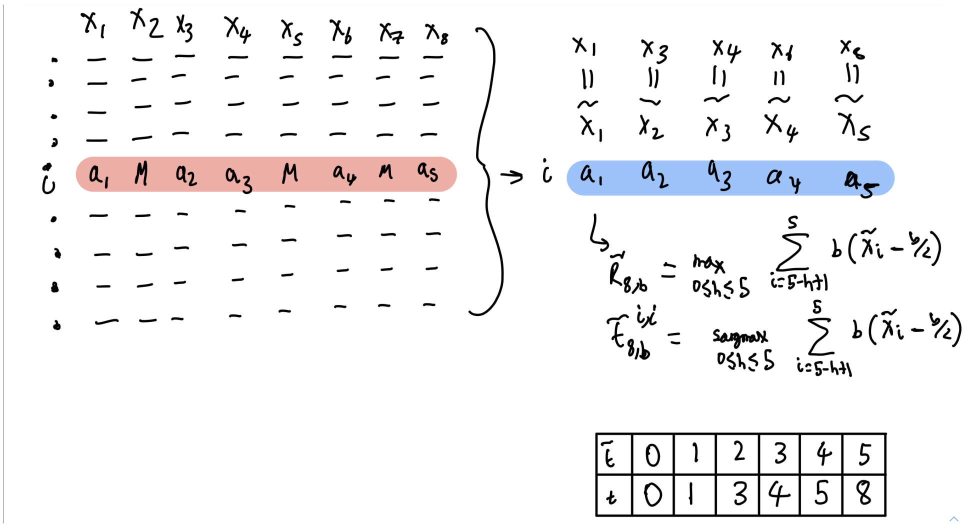


Figure 2: Illustration demonstrating the calculation of diagonal statistics for *MissOcd*

Note the close similarity of this modified step with the MissCUSUM transformation which also performs the corresponding CUSUM likelihood ratio test but again considers only the observed data points.

As before, from this we can then form diagonal statistics.

Now, to calculate off-diagonal statistics, if our putative changepoint for the 'j-th' (from the diagonal statistics) co-ordinate is at $t = TrueTime(\tilde{t}^{(j,j)}_{n,b}) = t^{(j,j)}_{n,b}$, then we sum the contributions in the other co-ordinates for '$t > t^{(j,j)}_{n,b}$', whilst ignoring data that is missing, so that:

$$A^{j',j}_{n,b} = \sum_{i=n-t^{(j,j)}_{n,b}+1}^{n} (\tilde{X}_i)_{j'}$$

Where $\tilde{X}$ is the matrix of observations with missing data entries ('NA') replaced with 0.

Importantly, however, previously the normalisation factor was $\max(t^j_{n,b}, 1)$. Now, to account for missingness, we ought to use $\tilde{t}^{(j',j)}_{n,b}$, the number of non-missing values in the 'j^{th}' co-ordinate for $t > t^{(j,j)}_{n,b}$ which yields the factor $\max(\tilde{t}^{(j',j)}_{n,b}, 1)$. Our corresponding off-diagonal statistic then becomes:

$$Q^j_{n,b} = \sum_{j' \neq j} \frac{(A^{j',j}_{n,b})^2}{\max(\tilde{t}^{(j',j)}_{n,b}, 1)}$$

A final remark we make is that we can attach to Q a thresholding term $\{|A^{j',j}_{b,n}| \geq a\sqrt{\tilde{t}^{(j',j)}_{b,n}}\}$. When we know the signal has some sort of sparsity structure, we limit our aggregation to exclude excessive noise, only considering the more significant co-ordinates. Note we can include this in the *ocd* algorithm as well.

If sparsity is unknown, it might be best to perform what the authors call the 'adaptive' procedure, by calculating two (or more) off diagonal statistics for varying levels of the sparsity parameter 'a' and flagging a changepoint when any one of them exceeds the threshold value (note $a = 0$ corresponds to the completely dense case).

Section 4: Theoretical analysis

Now, alongside '$MissOcd$' just as in the original paper which additionaly outlined *ocd*′, we present a minor modification of the algorithm which allows us to provide theoretical guarantees, 'namely' $MissOcd'$ (see Algorithm 3 for the pseudocode of $MissOcd'$). The main difference is the introduction of the τ variable. The reason becomes clear when we look at

$$A^{j',j}_{n,b} = \sum_{i=n-t^{(j,j)}_{n,b}+1}^{n} (\tilde{X}_i)_{j'}$$

If $t^{j,j}_{n,b} > n - z$, the time since the true changepoint, then A^j_b will be aggregating **both pre and post-changepoint data** and so the distribution of the off diagonal statistic would become hard to control. This becomes an issue if we wanted to bound the worst-case response delay, since we take the maximum over all possible changepoint locations and values of the pre-change data $(X_1, X_2, \ldots, X_z)$. This could skew the test statistics. To remedy this, we construct $\tau^{j,j}_{n,b}$ so that only a fraction (between 1/2 and 3/4 in the case of *ocd*′) of this tail is used for aggregation. In this case, we can guarantee that only post-changepoint data is used for aggregation.

Many of the results used in the proof for theoretical analysis are built on the proof framework of the original paper [11]. We will explain all the modifications we make to this proofs for our setting.

Algorithm 3 Pseudocode for $MissOcd'$

Input Observations $X_1, X_2, \dots \in \mathbb{R}^p, \beta > 0, a > 0, T^d > 0, T^o > 0$

Define $b_k = \left\{ \pm \frac{\beta}{\sqrt{2^k \log_2(2p)}} \right\} \quad \mathcal{B} = \bigcup_{k=1}^{\lfloor \log_2(p) \rfloor} b_k, \quad \mathcal{B}^0 = \left\{ b_{1+\lfloor \log_2(p) \rfloor} \right\}$

Initialise $n = 0,\ A_b = \Lambda_b = \tilde{\Lambda}_b = 0 \in \mathbb{R}^{p \times p}$ and $\tilde{t}_b^{(\cdot,\cdot)} = \tilde{\tau}_b^{(\cdot,\cdot)} = \tilde{\tau}_b^{*(\cdot,\cdot)} = 0 \in \mathbb{R}^{p \times p}$ for $b \in \mathcal{B} \cup \mathcal{B}^0$

repeat

  $n \leftarrow n + 1$

  Read in current data X_n

  Let $\tilde{X}_n$ be X_n with missing data replaced with 0

  Set $\omega \in \mathbb{R}^p$, $\omega_i = \begin{cases} 1 & (X_n)_i \text{ is not missing} \\ 0 & (X_n)_i \text{ is missing} \end{cases}$

  for $(j, b) \in [1:p] \times (\mathcal{B} \cup \mathcal{B}^0)$ **do**

    $\tilde{t}_b^{(\cdot,j)} \leftarrow \tilde{t}_b^{(\cdot,j)} + \omega$

    $A_b^{(\cdot,j)} \leftarrow A_b^{(\cdot,j)} + \tilde{X}_n$

    Let $\delta = 0$ if $\tilde{t}_b^{(j,j)}$ is a power of 2 and $\delta = 1$ if not

    $\tilde{\tau}_b^{(\cdot,j)} \leftarrow \tilde{\tau}_b^{(\cdot,j)}\delta + \tilde{\tau}_b^{*(\cdot,j)}(1-\delta) + \omega$ and $\Lambda_b^{(\cdot,j)} \leftarrow \Lambda_b^{(\cdot,j)}\delta + \tilde{\Lambda}_b^{(\cdot,j)}(1-\delta) + \tilde{X}_n$

    $\tilde{\tau}_b^{*(\cdot,j)} \leftarrow (\tilde{\tau}_b^{*(\cdot,j)} + \omega)\delta$ and $\tilde{\Lambda}^{(\cdot,j)} \leftarrow (\tilde{\Lambda}^{(\cdot,j)} + \tilde{X}_n)\delta$

    if $bA_b^{(j,j)} - b^2 \frac{\tilde{t}_b^{(j,j)}}{2} \le 0$ **then**

      $(\tilde{t}_b^{(\cdot,j)}, \tilde{\tau}_b^{(\cdot,j)}, \tilde{\tau}_b^{*(\cdot,j)}) \leftarrow (0,0,0)$

      $(A_b^{(\cdot,j)}, \Lambda_b^{(\cdot,j)}, \tilde{\Lambda}_b^{(\cdot,j)}) \leftarrow (0,0,0)$

    end if

    Generate For $k, j \in [1:p]$ $Z_{k,j} \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$

    Set $Q_b^j = \sum_{j' \neq j} \frac{(\Lambda_b^{j',j})^2 + Z_{j',j}^2 \mathbb{1}\{\Lambda_b^{j',j} = 0\}}{\max(\tilde{\tau}_b^{j',j}, 1)} \mathbb{1}\left\{ |\Lambda_b^{j',j}| \ge a\sqrt{\tilde{\tau}_b^{(j',j)}} \right\}$

  end for

  Set $S^d \leftarrow \max_{(j,b) \in [1:p] \times (\mathcal{B} \cup \mathcal{B}^0)} \left(bA_b^{(j,j)} - b^2 \frac{\tilde{t}_b^{(j,j)}}{2}\right)$

  Set $S^o \leftarrow \max_{(j,b) \in [1:p] \times \mathcal{B}} (Q_b^j)$

until $S^d \ge T^d$ or $S^o \ge T^o$

Return $N = n$

Bounds on the patience

The first question we must ask when deriving bounds on the patience is "If we use the same theoretical thresholds as in the analysis of the *ocd′* algorithm, what would happen to the patience?"

The answer comes in two parts. Firstly ,recall that the diagonal statistics are formed from hypothesis test on a 'contracted data set' of the observed data points only. Now if in co-ordinate j at time n only $m \le n$ data points are observed, we can cast this in the setting of the original 'ocd' method, but with only time $t = m$ having passed. The probability that the diagonal statistic has not flagged a changepoint will be $\mathbb{P}_{missocd'}(N^d \le$

$n) = \mathbb{P}_{ocd'}(N^d \leq m) \leq \mathbb{P}_{ocd'}(N^d \leq n)$, where $\mathbb{P}_{missocd'}$ and $\mathbb{P}_{ocd'}$ denote probabilities in the cases with and without missingness. From this, we expect N^d to be on average larger than in the *ocd′* algorithm. This will be made precise in the theorem which provides a sharper bound in the case of missingness.

For the off-diagonal statistics, as we will see in the proof of this theorem, under no missingness the statistic of interest

$$Q^j_{n,b}|\tau^j_b \sim \chi^2_{p-1}\mathbb{1}\{\tau^j_b > 0\} \text{ under the null}$$

whereas with missingness

$$\tilde{Q}^j_{n,b}|(\tau^{(\cdot,j)}_b, \tilde{\tau}^{(\cdot,j)}_b) \sim \sum_{r=1,r\neq j}^{p} Z_r^2 \mathbb{1}\{\tilde{\tau}^{(r,j)}_b > 0\} \text{ where } Z_1, Z_2, \ldots, Z_p \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$$

If $\tau^j_b = 0$ (or $\tilde{\tau}^{j,j}_b = 0$ in the case of $MissOcd'$) , both statistics are 0 but if $\tau^j_b > 0$ ($\tilde{\tau}^{j,j}_b > 0$ for $MissOcd'$), $Q^j_b|\tau^j_b$ follows a χ^2_{p-1} distribution whilst there is still a chance that $\tilde{\tau}^{r,j}_b = 0$, since all values in a given co-ordinate for $t > \tau^{j,j}_b$ could be missing. So, this distribution has fewer degrees of freedom, will be stochastically dominated and less likely to flag a changepoint. We note, however, making this precise would require knowledge of τ^j_b and the amount of missingness in each co-ordinate which becomes more difficult to analyse and our method, just as in the original proof, ignores this.

The theorem below gives a bound on the patience of $MissOcd'$:

Theorem 0.1. *Letting N be the output from $Missocd'$ with parameters $a = 0, T^d = \log(16p\gamma\log_2(4p)), T^o = \psi(\tilde{T}^o), \tilde{T}^o = 2\log(16p\gamma\log_2(2p))$ where $\psi(x) = p - 1 + x + \sqrt{2(p-1)x}$ and suppose that $\gamma \geq 1$ and $m = \lfloor 2\gamma \rfloor \geq \frac{1}{\min_{j\in[1:p]} p_j}$*
Then

$$\mathbb{E}_0(N) \geq \frac{\gamma}{2}(3 - \frac{\sum_{i=1}^p p_i}{p})$$

Proof. Let $m := \lfloor 2\gamma \rfloor$
We derive a sufficient condition for bounding the patience:

$$\begin{aligned}
\mathbb{E}_0(N) = \mathbb{E}_0(\min\{N^o, N^d\}) &\geq \mathbb{E}_0(\mathbb{1}\{\min(N^o, N^d)\}\min(N^o, N^d)) && (1)\\
&\geq 2\gamma\mathbb{P}_0(\min(N^o, N^d) > 2\gamma) && (2)\\
&\geq 2\gamma(1 - \mathbb{P}_0(\{N^o \leq 2\gamma\} - \{N^d \leq 2\gamma\})) && (3)
\end{aligned}$$

So, we need only upper bound $\mathbb{P}_0(N^o \leq 2\gamma)$ and $\mathbb{P}_0(N^d \leq 2\gamma)$

$$\begin{aligned}
\mathbb{P}_0(N^o \leq m) \leq \mathbb{P}_0(\bigcup_{n\in[1:m],j\in[1:p],b\in\mathcal{B}} Q^j_{n,b} \geq T^o) &\leq \sum_{n\in[1:m],j\in[1:p],b\in\mathcal{B}} \mathbb{P}_0(Q^j_{n,b} \geq T^o) && (4)\\
&= \sum_{n\in[1:m],j\in[1:p],b\in\mathcal{B}} \mathbb{E}_0\left[\mathbb{P}_0(Q^j_{n,b} \geq T^o|\tilde{\tau}^{(\cdot,j)}_{n,b})\right] && (5)
\end{aligned}$$

Under the null, $\Lambda_{n,b}^{(k,j)}|\tilde{\tau}_{n,b}^{(\cdot,j)} \sim \mathcal{N}(0, \tilde{\tau}_{n,b}^{(\cdot,j)})$
So

$$Q_{n,b}^{(k,j)}|\tilde{\tau}_{n,b}^{(\cdot,j)} = \sum_{r=1, r\neq j}^{p} Z_r^2 \mathbb{1}\{\tilde{\tau}_b^{(j,r)} > 0\} \text{ where } Z_1, Z_2, \ldots, Z_p \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$$

Note previously all the $\tilde{\tau}_b^{r,j}$ would be equal to $\tau_b^{j,j} := TrueTime(\tilde{\tau}_b^{j,j})$ so the above distribution would simply be $\chi_{p-1}^2 \mathbb{1}\{\tau_b^{j,j} > 0\}$
Note

$$Q_{n,b}^{(k,j)}|\tilde{\tau}_{n,b}^{(\cdot,j)} \leq \sum_{r=1, r\neq j}^{p} Z_r^2 \sim \chi_{p-1}^2 \implies \mathbb{P}_0(Q_{n,b}^{(k,j)} \geq T^o|\tilde{\tau}_{n,b}^{(\cdot,j)}) \leq \mathbb{P}_0(\sum_{r=1, r\neq j}^{p} Z_r^2 \geq T^o) \tag{6}$$

The quantity $\mathbb{P}_0(\sum_{r=1, r\neq j}^{p} Z_r^2 \geq T^o)$ is the same bound as per the original proof. There, a tail bound on the chi-squared distribution was used and it was shown that $\mathbb{P}_0(\sum_{r=1, r\neq j}^{p} Z_r^2 \geq T^o) \leq \exp(-\tilde{T}^o/2)$ and from this point we can proceed identically, combining (5) and (6) to obtain

$$\mathbb{P}_0(N^o) \leq |\mathcal{B}|mp\exp(-\tilde{T}^o/2) \leq \frac{1}{4}$$

Turning our attention to N^d, by independence of the rows, we can derive a bound on $N_b^j := \inf\{R_{n,b}^j \geq T^d\}$, and then use independence to derive a bound on N^d.
It would be nice if we could use the result for no missingness which showed that

$$\mathbb{P}_{0,ocd'}(N_b^j \leq k) \leq 1 - (1 - \exp(-T^d))^k \tag{7}$$

To do this, we would just need to condition on the number of non-missing observations. Let $\mathbb{P}_{0,ocd'}$ denote the corresponding probabilities under no missingness and let $M_m^j \sim Bin(m, p_j)$ be the number of non-missing observations in the j^{th} co-ordinate.

$\mathbb{P}_{0,ocd'}(N_b^j \leq i)$

$$\begin{aligned}
\mathbb{P}_0(N_b^j \leq m) &= \sum_{i=0}^{m} \mathbb{P}_{0,ocd'}(N_b^j \leq i)\mathbb{P}_0(M_m^j = i) \\
\text{By (7)} \leftarrow &\leq \sum_{i=0}^{m}(1 - (1 - \exp(-T^d))^i)\mathbb{P}_0(M_m^j = i) \\
&= 1 - \sum_{i=0}^{m}(1 - \exp(-T^d))^i\mathbb{P}_0(M_m^j = i) \\
\text{By convexity of } \alpha^x \text{ for } \alpha \in [0,1] \leftarrow &\leq 1 - (1 - \exp(-T^d))^{\sum_{i=0}^{m} i \times \mathbb{P}_0(M_m^j = i)} \\
&= 1 - (1 - \exp(-T^d))^{\mathbb{E}_0(M_m^j)} \\
&= 1 - (1 - \exp(-T^d))^{mp_j}
\end{aligned}$$

so

$$\mathbb{P}_0(N_b > m) = \mathbb{P}_0(\bigcap_{j\in[1:p], b\in\mathcal{B}\cup\mathcal{B}_0} \{N_b^j > m\}) = \prod_{j\in[1:p]} \left[1 - \mathbb{P}_0(N_b^j \leq m)\right]$$

$$\geq \prod_{j\in[1:p]} \left[1 - \underbrace{|\mathcal{B}\cup\mathcal{B}_0|(1-(1-\exp(-T^d))^{mp_j})}_{\alpha}\right]$$

We now use a lemma from chapter 3, page 71 of [16] which states that:

Lemma 0.2. *For $x_k \in (0,1), a_k \geq 1, k \in [1:n]$ Then:*

$$\prod_{i=1}^{n}(1-x_k)^{a_k} > 1 - \sum_{i=1}^{n} a_k x_k$$

Applying lemma 0.2 on α (which we can verify lies in $[0,1]$):

$$\geq 1 - |\mathcal{B}\cup\mathcal{B}_0| \sum_{j=1}^{p}(1-(1-\exp(-T^d))^{mp_j}) = 1 - p|\mathcal{B}\cup\mathcal{B}_0| + |\mathcal{B}\cup\mathcal{B}_0| \sum_{j=1}^{p}(1-\exp(-T^d))^{mp_j})$$

$$\geq 1 - p|\mathcal{B}\cup\mathcal{B}_0| + |\mathcal{B}\cup\mathcal{B}_0| \sum_{j=1}^{p} mp_j(1-\exp(-T^d))) \text{ applying lemma 0.2 again}$$

$$= 1 - mp|\mathcal{B}\cup\mathcal{B}_0| \sum_{j=1}^{p} \left[\frac{p_j}{p}\right] \exp(-T^d) \geq 1 - \sum_{j=1}^{p} \left[\frac{p_j}{p}\right] /4$$

Note we use the bound on m when applying Lemma 0.2 in the first line. Consequently

$$\mathbb{E}_0(N) \geq 2\gamma(1 - 1/4 - \sum_{j=1}^{p} \left[\frac{p_j}{p}\right] /4) = \gamma\left(\frac{3}{2} - \sum_{j=1}^{p} \left[\frac{p_j}{p}\right] /2\right)$$

Note that in the case of no-missingness, this becomes γ as before. □

Bounds on the response delay

As in the original proof, the non-negativity of the bound allows us to restrict ourselves to times after the changepoint, considering only $\{N > z\}$. The main idea used in this proof is to get similar bounds for analogous quantities as in the original proof. For this, we have to consider separately 'good' and 'bad' events, whereby on 'good' events, a sufficient amount of data is observed so we can apply a similar argument to derive similar bounds as before.

On the other hand, 'bad' events arise through a high proportion of missing data, and this no longer applies. However, we also show that such events are sufficiently unlikely. Our eventual bound, therefore, will consist of terms that resemble the original bound and additional terms to account for 'bad' events that occur due to missingness.

Theorem 0.3. *If N is the output from the algorithm Missocd′ with parameter values $a = 0, T^d = \log(16p\gamma\log_2(4p)), T^o = \psi(\tilde{T}^o), \tilde{T}^o = 2\log(16p\gamma\log_2(2p))$, then there exists a universal constant C such that for $s \geq 2$*

$$\mathbb{E}_\theta^{wc}(N) \leq C\left\{\max\{1, (\frac{s\log(ep\gamma)\log(ep)}{\beta^2})^{4/3}, \frac{p(\log(ep))^2}{\nu^4}, p\sum_{j=1}^{p}\frac{1}{q_j^{36/7}}, (1+\frac{\log(ep)}{\beta^2})\frac{1}{\min_{j\in[1:p]} p_j}\}\right\}$$

and for $s = 1$:

$$\mathbb{E}_\theta^{wc}(N) \leq C(\max\{1, \frac{\log(ep\gamma)\log(ep)}{\beta\nu}, \frac{\log(ep\gamma)^2\log(ep)^2}{\beta^2\nu^2}, \frac{(\log(ep))^2}{\nu^4}, \min_{j\in[1:p]}\frac{1}{p_j^{8/3}}\})$$

Proof. We begin by providing a table of relevant quantities that will be used throughout the proof

Quantity of interest/Result	**Remarks**
Effective support and effective sparsity, specifically, there exists a smallest $s(\theta) \in \{2^0, 2^1, \dots, 2^{\lfloor\log_2 p\rfloor}\}$ such that for $S(\theta) := \left\{j \in [1:p] : \theta_j \geq \frac{\|\theta\|_2}{\sqrt{s(\theta\log_2(2p)}}\right\}$ we have $\|S(\theta)\| \geq s(\theta)$	$S(\theta), s(\theta)$ are known as the effective support and effective sparsity of θ respectively. This result is Lemma 17 in [11]. *Note this also fits with our previous notion of sparsity since* $\|\theta\|_0 \leq s \implies \|s(\theta)\| \leq s$.
$\exists b_* \in \{\frac{-\beta}{\sqrt{s(\theta\log_2(2p)}}, \frac{\beta}{\sqrt{s(\theta\log_2(2p)}}\}$ such that $\mathcal{J} := \{j \in [1:p] : \frac{\theta_j}{b_*} \geq 1, \|\theta_j\| \leq \frac{\nu}{\sqrt{2}}\}$ satisfies $\|\mathcal{J}\| \geq s(\theta)/2$	This follows from the result that the set $\{j \in [1:p] : \|\theta_j\| > \frac{\nu}{\sqrt{2}}\}$ has cardinality at most 1 since otherwise $\|\theta\|_2^2 > \frac{\nu^2}{2} + \frac{\nu^2}{2} = \nu^2$
$q(\alpha; X_{[1:z]}) := \inf\{y \in \mathbb{R} : \|\{j \in \mathcal{J} : \tilde{t}^{j,j}_{z,b_*} \leq y\}\| \geq \alpha\|\mathcal{J}\|\}$.	We just defined the value of b_*. **Importantly, the tail $\tilde{t}^{j,j}_{z,b_*}$ is obtained from using threshold values equal to ∞ and sparsity parameter $a = 0$**
$\mathcal{J}_\alpha := \{j \in \mathcal{J} : \tilde{t}^{j,j}_{z,b_*} \leq q(\alpha)\}$	We get that $\|\mathcal{J}_\alpha\| \geq \alpha\|\mathcal{J}\| \geq \alpha s/2$. This follows directly from the definition of $q(\alpha)$ as well as our previous bound $\|\mathcal{J}\| \geq s$

We now mention a bound related to $q(\alpha)$. We will denote the corresponding quantities defined in *ocd* with a '*'. Under the *ocd′* algorithm, we had the definition $q^*(\alpha; X_{[1:z]}) := \inf\{y \in \mathbb{R} : |\{j \in \mathcal{J} : t^j_{z,b_*} \leq y\}| \geq \alpha|\mathcal{J}|\}$ with the following bounds: By Proposition 8 of [11] , on the event $\{N > z\}$:

$$q^*(1, X_{1:z}, \theta) \leq \frac{8T^d s\log_2(2p)}{\beta^2}$$

Now importantly, this bound is independent of z. As we have done before, we rely on the fact that $\tilde{t}^{j,j}_b$ is the "contracted version" of the tail.

Specifically, for proposition 8, we can use the same proof as per [11], replacing $t^j_{z,b} \to \tilde{t}^{j,j}_{z,b}$ and replacing $z \to k$ where k is the number of non-missing data observations in the j^{th} co-ordinate up to the changepoint. This allows us to derive the exact same bound on q.

We are now ready to proceed with the main part of the proof: We split our analysis into 2 cases. Let $R(r) \in \mathbb{R}^p$ denote the number of non-missing observations at time r **after the changepoint**. Note that $R_j = R_j(r) \sim Bin(r, p_j)$. Consider the events for $j \in \mathcal{J}_\alpha$

$$\Omega^j_{R_j} := \{\tilde{t}^{j,j}_{z+r,b_*} > \frac{2R_j}{3}\}$$

In the first case, we consider when all the tails are "small". Here, Lemmas 11 and Lemma 15(b) from [11] applied to $\min(\tilde{t}^{j,j}_{z+r,b_*}, R_j)$ for $j \in \mathcal{J}_\alpha$ gives that:

$$\mathbb{P}_0\left(\bigcap_{j\in\mathcal{J}_\alpha}(\Omega^j_{R_j})^c|X_{1:z}, R\right) \le \prod_{j\in\mathcal{J}_\alpha}\exp\left(-\frac{b_*^2 R_j}{12}\right) = \exp\left(-\frac{b_*^2}{12}\left[\sum_{j\in\mathcal{J}_\alpha} R_j\right]\right) \tag{8}$$

We now focus on the event $\Omega^j_{R_j} \cap A_1$ where $A_1 = \{R_j \ge 2\} \cap \{R_j \ge 3q(\alpha)\}$ for some $j \in \mathcal{J}_\alpha$ This allows us to apply Lemma 18 from [11] since $\tilde{t}^{j,j}_{z+r,b_*} > \lceil\frac{2R_j}{3}\rceil > 2$ Applying the lemma we get that:

$$\frac{R_j}{3} \le \frac{\tilde{t}^{j,j}_{z+r,b_*}}{2} \le \tilde{\tau}^{j,j}_{z+r,b_*} \le 3\frac{\tilde{t}^{j,j}_{z+r,b_*}}{4} \le 3\frac{\tilde{t}^{j,j}_{k,b_*} + R_j}{4} \le \frac{3(q(\alpha) + R_j)}{4} \le R_j$$

In particular we have

$$\frac{R_j}{3} < \tilde{\tau}^{j,j}_{z+r,b_*} \le R_j$$

which means that the tail $\tilde{\tau}^{j,j}_{z+r,b_*}$ consists only of post-changepoint observations.
Note also that

$$\tilde{\tau}^{k,j}_{z+r,b_*}|\tilde{\tau}^{j,j}_{z+r,b_*} := R^{k,j} \sim Bin(\tilde{\tau}^{j,j}_{z+r,b_*}, p_k)$$

So that for $k \in [1:p]\backslash\{j\}$

$$\Lambda^{k,j}_{z+r,b_*}|\tilde{\tau}^{k,j}_{z+r,b_*}, X_{1:z} = x_{1:z} \overset{\text{ind}}{\sim} \mathcal{N}(\theta_k\tilde{\tau}^{k,j}_{z+r,b_*}, \tilde{\tau}^{k,j}_{z+r,b_*})$$

Then $\displaystyle\sum_{k=1,k\ne j}^{p}\frac{(\Lambda^{k,j})^2 + Z^2_{k,j}\mathbb{1}\{\Lambda^{k,j} = 0\}}{\max(\tilde{\tau}^{k,j}_{z+r,b_*}, 1)}$ has a non-central chi-squared distribution with $p-1$ degrees of freedom and non-centrality parameter $\displaystyle\sum_{k\ne j}(\theta^k)^2\tilde{\tau}^{k,j}_{z+r,b_*}$ Fix $c_1, c_2 \in [0,1]$ and work on the events $A_2 := \{R_j \ge (r)^{1-c_1}\}$ and $A_3 := \displaystyle\bigcap_{k\ne j}\{\frac{R^{k,j}}{3} \ge \frac{1}{3}(r)^{1-c_1-c_2}\}$. Note that at this stage, we could have instead used any sets A_2 and A_3 such that the proportion of observations tends to 0 as $r \to \infty$, we have just used these for simplicity (we are just demonstrating convergence at a given rate, not the optimal rate). We have

$$\tilde{\tau}^{k,j}_{z+r,b_*} \ge \frac{1}{3}(r)^{1-c_1} \implies R^{k,j}\text{is stochastically larger than } Bin(\frac{1}{3}(r)^{1-c_1}, p_k) =: \tilde{R}^{k,j}$$

as well as

$$\text{Non centrality parameter} \geq \sum_{k\neq j}(\theta^k)^2\frac{1}{3}(r)^{1-c_1-c_2} \geq \frac{\nu^2}{6}r^{1-c_1-c_2}$$

where the last line follows since $j \in \mathcal{J} \implies |\theta_j| \leq \frac{\nu}{\sqrt{2}}$

Now define

$$E_r^j := \{\sum_{k=1,k\neq j}^{p}\frac{(\Lambda^{k,j})^2 + Z_{k,j}^2\mathbb{1}\{\Lambda^{k,j}=0\}}{\max(\tilde{\tau}^{k,j}_{z+r,b_*},1)} < T^o\}$$

As in the original proof, we use Birgé (2001 Lemma 8.1) [17] to bound the quantiles of such a non-central chi-squared distribution:

$$\mathbb{P}_{z,\theta}(E_r^j \cap (\Omega_r^j \cap A_1 \cap A_2 \cap A_3)|\tilde{\tau}^{j,j}_{z+r,b_*}, X_{1:z}=x_{1:z}) \leq \exp\left(-\frac{\left(\frac{\nu^2}{6}r^{1-c_1-c_2} - \tilde{T}^o - \sqrt{2(p-1)\tilde{T}^o}\right)^2}{4(p-1+\frac{\nu^2}{3}r^{1-c_1-c_2})}\right)$$

$$\leq \exp\left(-\frac{\frac{\nu^4}{6}(r)^{2(1-c_1-c_2)}}{576(p-1+\frac{\nu^2}{3}r^{1-c_1-c_2})}\right)$$

Where the last line holds when $r \geq \left[\frac{12(\tilde{T}^o + \sqrt{2(p-1)\tilde{T}^o})}{\nu^2}\right]^{\frac{1}{1-c_1-c_2}}$.

Combining all this together

$$\mathbb{P}_{z,\theta}(N > z+r|X_{1:z}) \leq \mathbb{E}\left[\mathbb{P}_{z,\theta}\left(\bigcap_{j\in\mathcal{J}_\alpha}(\Omega^j_{R_j})^c|X_{1:z},R\right)\right]$$
$$+\mathbb{P}_{z,\theta}(E_r^j \cap (\Omega_r^j \cap A_1 \cap A_2 \cap A_3)|\tilde{\tau}^{j,j}_{z+r,b_*}, X_{1:z}=x_{1:z}) + \mathbb{P}_{z,\theta}(A_1^c \cup A_2^c \cup A_3^c)$$
$$\leq \mathbb{E}\exp\left(-\frac{b_*^2}{12}\left[\sum_{j\in\mathcal{J}_\alpha}R_j\right]\right)$$
$$+\,p\exp\left(-\frac{\frac{\nu^4}{6}(r)^{2(1-c_1-c_2)}}{576(p-1+\frac{\nu^2}{3}r^{1-c_1-c_2})}\right)$$
$$+\,\mathbb{P}_{z,\theta}(A_1^c) + \mathbb{P}_{z,b}(A_2^c) + \mathbb{P}_{z,b}(A_3^c)$$
$$:= 1 + 2 + 3 + 4 + 5$$

Defining $r_0 := (3q(\alpha))^{\frac{1}{1-c_1}} + 2^{\frac{1}{1-c_1}} + \left[\frac{12(\tilde{T}^o - \sqrt{2(p-1)\tilde{T}^o})}{\nu^2}\right]^{\frac{1}{1-c_1-c_2}}$ and

noting

$$\mathbb{E}_{z,b}(\max(N-z,0)|X_{1:z}) = \int_0^\infty \mathbb{P}_{z,b}(N > z+u|X_{1:z})du \leq r_0 + \int_{r_0-1}^\infty \mathbb{P}_{z,b}(N > z+\lfloor u\rfloor|X_{1:z})du$$

we can bound the integrals for each of 1 : 5.

We begin with term 1

$$
\begin{aligned}
\leq \int_{r_0-1}^{\infty} \mathbb{E}\exp\left(-\frac{b_*^2}{12}\left[\sum_{j\in\mathcal{J}_\alpha} R_j(\lfloor u\rfloor)\right]\right) du &\leq \int_{r_0-1}^{\infty} \mathbb{E}\exp\left(-\frac{b_*^2}{12}\left[\sum_{j\in\mathcal{J}_\alpha} R_j(\lfloor u\rfloor)\right]\right) du && (9)\\
&= \int_{r_0-1}^{\infty}\left(\prod_{j\in\mathcal{J}_\alpha}\left[(1-p_j)+p_j\exp(\frac{-b_*^2}{12})\right]\right)^{\lfloor u\rfloor} du && (10)\\
&\leq \int_{0}^{\infty}\left(\prod_{j\in\mathcal{J}_\alpha}\left[(1-p_j)+p_j\exp(\frac{-b_*^2}{12})\right]\right)^{u} du && (11)\\
&= -\frac{1}{\sum_{j\in\mathcal{J}_\alpha}\log\left((1-p_j)+p_j\exp(\frac{-b_*^2}{12})\right)} && (12)\\
&\leq \frac{1}{(1-\exp(\frac{-b_*^2}{12}))\sum_{j\in\mathcal{J}_\alpha} p_j} && (13)\\
&\leq \left(\frac{12}{b_*^2}+1\right)\frac{1}{\sum_{j\in\mathcal{J}_\alpha} p_j} \leq \left(\frac{12}{b_*^2}+1\right)\frac{2}{s\alpha}\frac{1}{\min\limits_{j\in[1:p]} p_j} && (14)
\end{aligned}
$$

Where (10) follows from the moment generating function of the binomial and the last line uses our bound $|\mathcal{J}_\alpha| \leq \frac{s\alpha}{2}$.

Note that this bound uses $\min\limits_{j\in[1:p]} p_j$ since different co-ordinates have different amounts of missingness and we are unable to relate $j\in\mathcal{J}_\alpha$ to this.

For terms 3, 4, 5, note that

$$r > (2^{1/(1-c_1)} + 3q(\alpha)^{1/(1-c_1)}) \implies \mathbb{P}(A_1^c) \leq \mathbb{P}(A_2^c)$$

and also that bounding $\mathbb{P}(A_2^c)$ and $\mathbb{P}(A_3^c)$ amounts to finding tail bounds for the binomial distribution. Specifically for $\mathbb{P}(A_2^c)$

$$\mathbb{P}(R_j(r) < r^{1-c_1}) \leq \begin{cases} \exp(-\frac{r}{2p_j}\left(p_j - \frac{1}{r^{c_1}}\right)^2) & r \geq \frac{1}{p_j^{1/c_1}} \\ 1-\exp\left(-r\frac{\left(r^{-c_1}-p_j\right)^2}{2(p_j+\frac{r^{-c_1}-p_j}{3})}\right) & r < \frac{1}{p_j^{1/c_1}} \end{cases}$$

which follows from the Chernoff inequalities in Theorem 4 of [18] So

$$\begin{aligned}
\int_{r_0}^{\infty} \mathbb{P}(R_j < r^{1-c_1})dr &\leq \int_{1/p_j^{1/c_1}}^{\infty} \exp(-\frac{r}{2p_j}\left(p_j - \frac{1}{r^{c_1}}\right)^2)dr + \frac{1}{p_j^{1/c_1}} \\
&\leq \int_{1/p_j^{1/c_1}}^{\infty} \exp(-rp_j^3\,(r^{c_1} - p_j)^2)dr + p_j^{1/c_1} \\
&\leq \int_0^{\infty} \exp(-(u^{c_1} + p_j)^{1/c_1} p_j^3\left(u^{2c_1}\right))du + \frac{1}{p_j^{1/c_1}} \\
&\overset{\text{AM}-\text{GM}}{\leq} \int_0^{\infty} \exp(-(4u^{c_1}p_j)^{1/2c_1} p_j^3\left(u^{2c_1}\right))du + \frac{1}{p_j^{1/c_1}} \\
&= \frac{\Gamma(\frac{1}{2c_1+1/2})}{2c_1+1/2} \times \frac{1}{2^{1/[c_1(2c_1+1/2)]}} \times \frac{1}{p_j^{(3+\frac{1}{2c_1})(\frac{1}{2c_1+1/2})}} + \frac{1}{p_j^{1/c_1}}
\end{aligned}$$

Similarly we get that:

$$\begin{aligned}
&\int_{r_0}^{\infty} \mathbb{P}(R^{k,j} < r^{1-c_1-c_2})dr \\
\leq\ & \frac{\Gamma(\frac{1}{2c_2+(1-c_1)/2})}{2c_2+(1-c_1)/2} \times \frac{1}{2^{(1-c_1)/[c_2(2c_2+(1-c_1)/2)]}} \times \frac{1}{p_j^{(3+\frac{1-c_1}{2c_2})(\frac{1}{2c_2+(1-c_1)/2})}} + \frac{1}{p_j^{1/c_2}}
\end{aligned}$$

Finally, we consider 2 which is bounded by

$$\int_0^{\infty} \min(p\exp(-\frac{\nu^2}{384}u^{1-c_1-c_2}), 1)du + \int_0^{\infty} \min(p\exp(-\frac{\nu^4}{1152(p-1)}u^{2(1-c_1-c_2)}), 1)du$$

At this stage, although possible, the terms become rather long if we stick to using arbitrary c_1, c_2. We do this since the purpose of this proof is to demonstrate convergence and to give a minimal rate of convergence, rather than the optimal rate. We take $c_1 = c_2 = \frac{1}{4}$ and then the integral can be calculated exactly to be:

$$(\frac{384}{\nu^2})^2(\log(ep)^2 + 1) + \frac{1152(p-1)}{\nu^4}\log(ep)$$

For more general values of c_1, c_2, we can obtain similar bounds using Proposition 2.10 from [19] which provides upper bounds on the incomplete Gamma function.

We can now combine these results together to form our bound:

$$\begin{aligned}
\mathbb{E}_{z,\theta}(\max(N-z,0)|X_{1:z}) &\leq r_0 + \left(\frac{12}{b_*^2} + 1\right)\frac{2}{s\alpha}\frac{1}{\min_{j\in[1:p]} p_j} \\
&+ (\frac{384}{\nu^2})^2(\log(ep)^2+1) + \frac{1152(p-1)}{\nu^4}\log(ep) \\
&+ \sum_{j=1}^{p}\left[\frac{1}{2}\frac{1}{p_j^5} + \frac{2}{p_j^2}\right] + (p-1)\left(\frac{8\Gamma(\frac{8}{7})}{7} \times \frac{1}{2^{24/7}}\sum_{j=1}^{p}\frac{1}{p_j^{36/7}} + \sum_{j=1}^{p}\frac{1}{p_j^4}\right)
\end{aligned}$$

Since $\log(ep) \leq \tilde{T}^o$ and substituting for r_0, we can bound this by

$$K_1\left(1+q(\alpha)^{4/3}+\frac{1}{\nu^4}\left[(\tilde{T}^o)^2+p\tilde{T}^o+\sqrt{p}(\tilde{T}^o)^{3/2}+1\right]\right)+L_1(p\sum_{j=1}^{p}\frac{1}{p_j^{36/7}})$$
$$+(\frac{2}{s\alpha}+\frac{24\log_2(2p)}{\beta^2\alpha})\frac{1}{\min\limits_{j\in[1:p]}p_j}$$

For universal constants K_1, L_1
Substituting $\tilde{T}^o$ and taking $\alpha = 1$ and using the bound on $q(\alpha)$ (also substituting T^d), this becomes:

$$M_2\left(1+(\frac{s\log(ep\gamma)\log(ep)}{\beta^2})^{4/3}+\frac{p(\log(ep))^2}{\nu^4}+p\sum_{j=1}^{p}\frac{1}{p_j^{36/7}}+(1+\frac{\log(ep)}{\beta^2})\frac{1}{\min\limits_{j\in[1:p]}p_j}\right)$$

for a universal constant M_2, which proves the result for $s \geq 2$. The reason the proof above is only valid for $s \geq 2$ lies in the fact our bound that $|\mathcal{J}| \geq s/2$ can only be used when $s \geq 2$.

For the case $s = 1$ we proceed separately. We can begin with the original proof by noting that there exists $j_* \in [1:p]$ such that $\theta_{j_*} \leq \frac{\nu}{\sqrt{\log_2(2p)}}$. The proof then proceeds using only properties of the diagonal co-ordinates of the j_* co-ordinate and arrives at the bound:

$$\mathbb{P}_{z,\theta,ocd'}(N > z+n|X_{1:z}=x_{1:z}) \leq \frac{1}{2}\exp\left(-n\frac{(\theta_{j_*})^2}{32}\right)$$

for $n \geq \lceil 4T^d/(b_*\theta_{j_*}\rceil$ To account for missingness, we proceed similarly as before first defining:

$$n^*(n) \sim Bin(n, p_{j_*})$$
$$n \geq \left\lceil\frac{4T^d}{(b_*\theta_{j_*}}\right\rceil^2 := n_0$$
$$\omega := \left\{n^*(n) \geq \left\lceil\frac{4T^d}{b_*\theta_{j_*}}\right\rceil\right\}$$

where n^* denotes the number of non-missing observations. Then

$$\begin{aligned}\mathbb{E}_{z,\theta}(\max(N-z,0)|X_{1:z}=x_{1:z}) &= \sum_{n=0}^{\infty}\mathbb{P}_{z,\theta}(N>z+n|X_{1:z}=x_{1:z})\\ &\leq n_0+\frac{1}{2}\sum_{n=n_0}^{\infty}\exp\left(-\sqrt{n}\frac{(\theta_{j_*})^2}{32}\right)+\int_{n=n_0}^{\infty}\mathbb{P}(n^*(n)<n^{1/2})dn\\ &\leq n_0+\frac{1}{2}\int_0^{\infty}\exp(-\sqrt{n}\frac{(\theta_{j_*})^2}{32})dn+\int_{n=n_0}^{\infty}\mathbb{P}(n^*(n)<n^{1/2})dn\\ &\leq (1+\frac{4T^d}{b^*\theta_{j_*}})^2+(\frac{32^2}{\theta_{j_*}^4})+\frac{A_1}{p_{j_*}^{8/3}}\end{aligned}$$

where A_1 is a universal constant and the bound on the last term follows from the proof when we bounded terms 3, 4, 5. Substituting for T^d, θ_{j_*}, b_* yields for a universal constant A

$$A_2(1 + \frac{\log(ep\gamma)\log(ep)}{\beta\nu} + \frac{\log(ep\gamma)^2\log(ep)^2}{\beta^2\nu^2} + \frac{\log(ep)^2}{\nu^4} + \min_{j\in[1:p]} \frac{1}{p_j^{8/3}})$$

$\square$

Remarks:

We now make a few remarks about our theoretical analysis. The bound for the worst-case response delay is by no means optimal. Determining an optimal bound was not the purpose of this essay since we just wanted to demonstrate that such a bound could indeed be reached in the case of missingness. Moreover, unlike for the *ocd* algorithm, we do not have an improved bound for the average response delay. To do this, we would need to determine bounds for the moments of $q(\alpha)$ and could not just rely on the bound on its expectation found in proposition 9 in [11].

Section 5: Numerical Simulations

MCAR missingness

We test the performance of 'MissOcd' against a natural alternative method which we have labelled 'imputeocd'. 'Imputeocd' is a two-step procedure that uses a fixed-length window of length '$w-1$' which stores the data from the last w observations and uses these values to impute any missing data points for the most recent observation. This enables us to apply the original 'ocd' algorithm using a completed data set. See Figure 3 for a visual demonstration of how mean imputation works.

The imputation step allows us a lot of different scope for the precise way we impute missing values. Here, we have consulted the existing literature on real-time data imputation techniques [20] and list some of these below. For our simulations, however we only use the first of these and label the algorithm as 'imputeocd'.

We also make a note that we use the adaptive procedure throughout here using the recommended sparsity parameters $a = 0, a = \sqrt{8\log(p-1)}$.

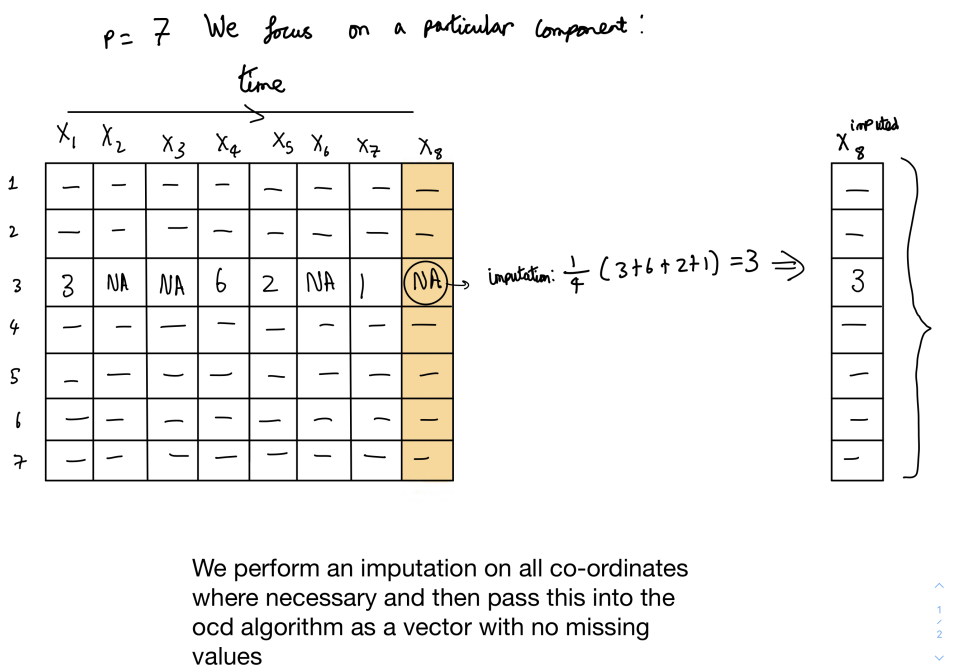


Figure 3: Illustration demonstrating mean imputation

Naïve mean imputation:

Here, missing data in the new observation in say, co-ordinate ‘j’ will be replaced by the mean of the previous ‘$w - 1$’observations (the mean is taken only on observed data, so missing data here will not be counted as an observation).

Weighted mean imputation/exponential method:

Our data is observed sequentially, so in the case where the window contains both pre- and post-changepoint data, our newly observed data would be a post-changepoint observation. Therefore, we would prefer that the more recent (post-changepoint) data is given more weight and ‘older’ data is weighted less.

Weighted) K-nearest Neighbours($k - NN$):

Given our partially observed data point at a given time, $k - NN$ finds the k vectors in our window, which are closest (in the l_2 norm sense) to our data point. Our vector is then imputed by taking a (weighted mean) of the k vectors. Issues will arise in co-ordinates where there is missing data and as per the suggestion in [20], we initially use mean imputation to fill in missing values and then perform k-NN on the imputed vectors.

A general feature of the imputation methods will be a bias-variance trade-off, whereby very small window sizes would lead to high variances in the estimates and consequently, to ensure a given patience, thresholds would have to prohibitively high, leading to large response delays. On the other hand, for very large window sizes the window would still contain pre-changepoint data for a large time after the changepoint has occurred. This leads to a persistent bias in estimates long after the changepoint has passed, once again leading to increased response delays.

Secondly, and perhaps more importantly, we note that the imputation methods will perform best when the noise due to variance in the estimates is less significant compared to the size of the signal. To guarantee this, the window size must be intimately dependent on our lower bound β for the signal strength. However, this would no longer be an online method due to our stringent definition that an online method permitted dependence only on the number of bits required to store the new observation. Forcing it to be an online method by arbitrarily fixing the window length will lead to a situation where there exists a signal strength such that the noise is simply too large to allow for any changepoint detection at all. We will observe this effect in our numerical simulations below.

Remark: To provide numerical simulations, we modify the R package 'ocd' provided by the original authors, to implement '*MissOcd*' and to allow for missing data inputs and the parts of the codes that are modified are provided in the appendices. The data imputed form with mean imputation has also been implemented as 'imputeocd'. The scripts we use are included in the appendices.

Selection of thresholds

As referenced in the original paper, in practise, the theoretical thresholds are often loose and precision is lost. To remedy this, we will use Monte Carlo simulations, full details can be found in section 4.1 of [11]. In essence, the argument uses the renewal nature of the process. We can then argue that the detection delay will be exponentially distributed and use this to determine thresholds for a given patience. Again, from Tables 1 and 2 we provide empirical evidence to demonstrate that we do indeed have approximate patience control. Note that $\mathbb{E}(X|X < 2500) = 483.04$ for $X \sim exp(1/500)$

Method	**Response delay**
imputeocd	513.62
missocd	485.59

Table 1: We ran the algorithm under the null with dimension 10, patience (for Monte-Carlo) 500, a missingness vector that was a realisation of a $Beta(9, 1)$ distribution and a window size of 20 for 'imputeocd'. We average delay over 500 repetitions and restrict ourselves to delays of length at most 2500.

Method	**Response delay**
imputeocd	466.91
missocd	503.63

Table 2: We ran the algorithm under the null with dimension 100, patience (for Monte-Carlo) 500, a missingness vector that was 0.3 in each co-ordinate and a window size of 20 for 'imputeocd'. We average delay over 500 repetitions and restrict ourselves to delays of length at most 2500.

Now, Table 3 compares the imputation method '*imputeocd*' with '*MissOcd*', varying several of our parameters. The original OCD with full data (data is generated and then removed) is provided as an additional benchmark.

q	Sparsity	ν	ocd	imputeocd	missocd
$q_i = 0.1$	5	1	33.68	36.59	**35.45**
$q_i = 0.1$	5	2	10.32	12.26	**10.41**
$q_i = 0.1$	10	1	40.56	**42.59**	43.08
$q_i = 0.1$	10	2	12.08	12.99	**12.45**
$q_i = 0.1$	100	1	62.54	**51.08**	57.37
$q_i = 0.1$	100	2	16.54	18.34	**17.51**
$q_i = 0.4$	5	1	33.45	58.53	**46.88**
$q_i = 0.4$	5	2	10.16	22.62	**15.64**
$q_i = 0.4$	10	1	39.5	65.98	**55.99**
$q_i = 0.4$	10	2	11.88	24.74	**16.51**
$q_i = 0.4$	100	1	53.74	**73.56**	74.67
$q_i = 0.4$	100	2	14.94	30.37	**21.97**
$q_i = 0.7$	5	1	41.15	144.28	**129.24**
$q_i = 0.7$	5	2	11.55	49.51	**35.10**
$q_i = 0.7$	10	1	48.37	162.38	**138.68**
$q_i = 0.7$	10	2	13.98	58.43	**39.83**
$q_i = 0.7$	100	1	72.29	188.30	**187.15**
$q_i = 0.7$	100	2	17.99	65.89	**49.8**
$Unif[0,1]$	5	1	35.49	211.53	**54.63**
$Unif[0,1]$	5	2	10.45	114.7	**17.35**
$Unif[0,1]$	10	1	40.28	220.44	**66.96**
$Unif[0,1]$	10	2	12.31	85.27	**20.97**
$Unif[0,1]$	100	1	57.25	262.18	**83.88**
$Unif[0,1]$	100	2	15.16	85.1	**27.71**
$beta(1,9)$	5	1	38.92	39.91	**35.23**
$beta(1,9)$	5	2	11.18	12.43	**10.97**
$beta(1,9)$	10	1	37.04	39.26	**36.04**
$beta(1,9)$	10	2	12.12	13.94	**12.07**
$beta(1,9)$	100	1	57.18	53.22	**51.01**
$beta(1,9)$	100	2	17.38	20.11	**15.63**
$beta(5,5)$	5	1	34.66	94.60	**60.26**
$beta(5,5)$	5	2	10.97	36.14	**19.92**
$beta(5,5)$	10	1	40.68	95.96	**65.98**
$beta(5,5)$	10	2	12.29	38.96	**21.92**
$beta(5,5)$	100	1	55.37	103.52	**81.95**
$beta(5,5)$	100	2	42.41	20.11	**26.75**

Table 3: Table comparing the efficacy of 'missocd' and 'ocd' for a range of types of missingness. For easier comparison, both algorithms were performed on the same datasets each time

The dimension was fixed at 100. The changepoint occurs at $z = 0$, sparsity takes values in (5,10,100) and $\nu \in \{1, 2\}$. The missingness types considered was was (a) the same in all co-ordinates with $q_i \in \{0.1, 0.4, 0.7\}$ (b) realisations of a $Uniform(0,1)$ random variable and (c) realisations of a $beta(10v, 10(1-v))$ for $v \in \{0.1,\ 0.5\}$, patience = 2000. The patience for Monte-Carlo was set to 500 for all except the case when $q_i = 0.7$ where it was set to 2000. The window length for *imputeocd* was set to 20. The response delays were measured over 100 iterations. I was unable to use higher values of patience and more parameters due to computational limitations (I was not in a position to use parallel processing).

Now, as we controlled for the patience, comparisons are made by examining the response delay. Here, we can see $MissOcd$ generally outperforms *imputeocd* in the wide range of scenarios. As expected, performance is similar for low values of missingness but becomes more pronounced for higher values of missingness. This is especially noticeable for th $uniform[0,1]$ and $beta(5,5)$ cases. This represents the case we mentioned earlier when variance in the imputed value adversely affect detection strength to the extent where it becomes impossible.

Non-MCAR missingness

Here, we examine another form of missingness, as discussed earlier. Specifically, we analyse the observation from Lonsdchien et al.'s [13] where they mentioned the fallibility of imputation techniques when multiple co-ordinates become missing simultaneously. This could be explained by a sensor failure for example.

We examine the case when we have a window size of length 30. Moreover, at each point in time there is a 0.01 probability that 'block missingness' is triggered whereby half of the co-ordinates are missing for a duration of 25 time periods. We also chose to run the procedure with $\beta = 2$ and dimension $p = 100$.

In the plots below, we examine the pre-changepoint regime (which is used in the Monte-Carlo simulation to determine thresholds). Figure 4 denotes the mean imputed *ocd* algorithm whilst Figure 5 denotes the $MissOcd$ algorithm. In each of the plots, we show the diagonal, off-diagonal dense and off-diagonal sparse test statistics and mark with dashed vertical lines the point at which block missingness was triggered.

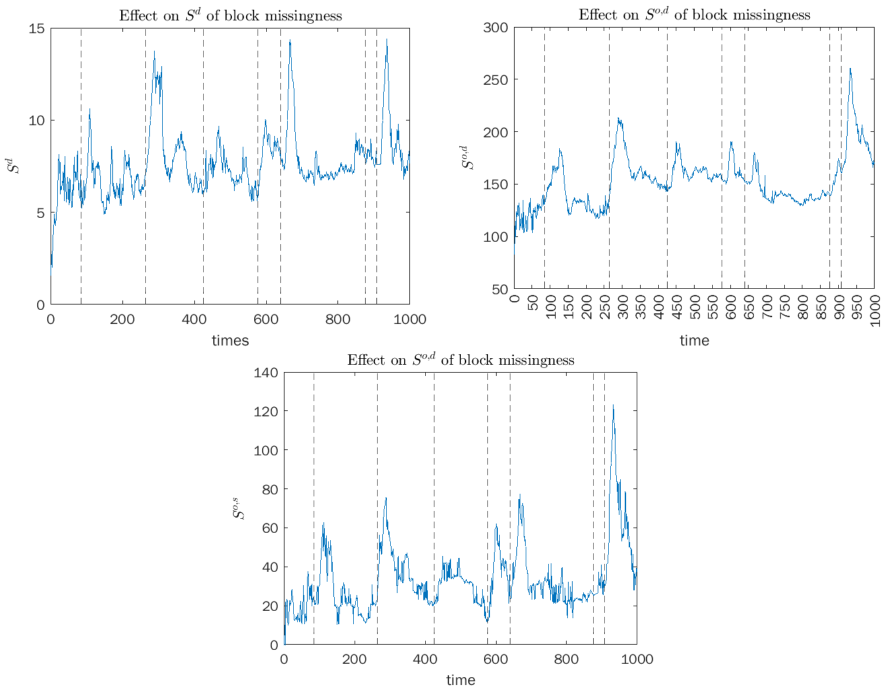


Figure 4: Plots illustrating the effects of block missingness on each of the three statistics in *imputeocd*

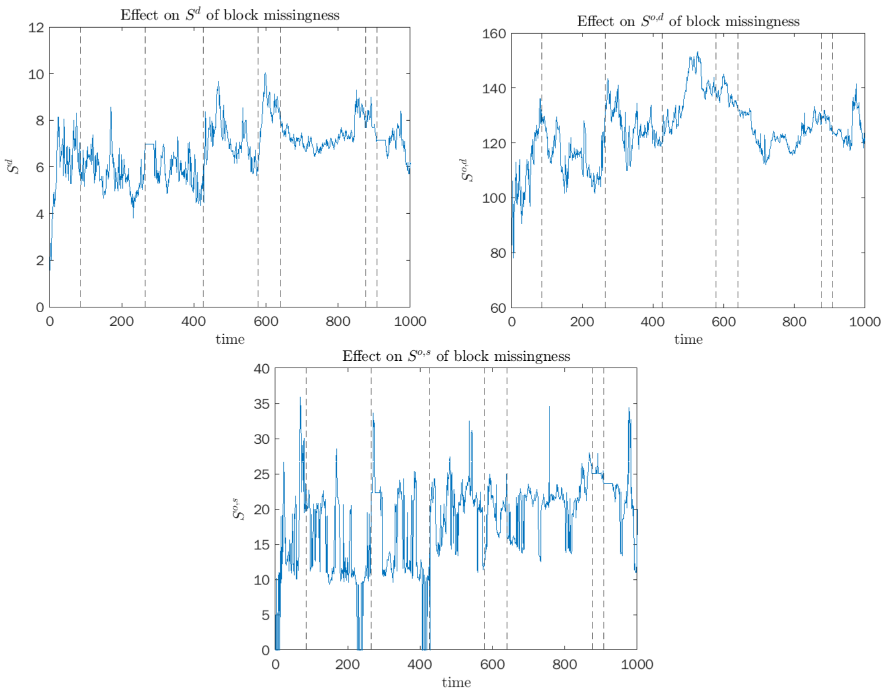


Figure 5: Plots illustrating the effects of block missingness on each of the three statistics in $MissOcd$

As we see, for *imputeocd*, all the large 'spikes' in the test statistics are preceded by the triggering of block missingness. This, in turn, results in the Monte-Carlo simulation requiring particularly high thresholds to ensure a given patience. This has the effect that it takes a particularly long time for certain signals to be detected in this setup.

Contrasting this with $MissOcd$, observe there are no significant spikes following periods of block missingness and as we would prefer prior to the changepoint, the statistics appear to perturb around a mean that is constant in time.

In summary, the $Missocd$ algorithm demonstrates its robustness in tackling other forms of missingness, which prove troublesome for imputation techniques. In fact, it appears that we could even obtain theoretical results for $MissOcd$ in this form of missingness as well.

Conclusion and Outlook

Throughout this essay, we were tasked with the problem of high-dimensional changepoint analysis, allowing for sparsity and missingness. We began with the *inspect* and *missinpect* where we tackled the offline problem. Here, we saw how we were able to compare the two algorithms, which gave us an insight on how we might handle the case of missing data. Benefiting from this intuition, we were able to extend this to the online case leading to the algorithm *ocd* and its extension *missocd*. Together, the algorithms when viewed as a whole, address the problem in all four cases (online, offline, missingness and no-missingness).

We were also able to demonstrate theoretical performance guarantees for $MissOcd$ whilst also demonstrating its efficacy on a selection of parameter values, as well as robustness to other forms of missingness.

Natural questions we might ask at this stage is whether we could relax some of our assumptions on the data generation process. We summarise some of these in the following table:

Modification/Extension	Remarks
Non-Gaussianity	As pointed out in [11], we can modify our arguments using the suggestions made in the original paper to obtain bounds in the case of sub-Gaussian random variables as well.
Spatial Dependence	This covers cases where the covariance matrix of X_i is no longer the identity. Here, we can use the pre-changepoint data to obtain an estimate for this and transform our data back into our previous setting.
Non-MCAR missingness	We briefly examined block missingness earlier, but in general, missingness can exhibit itself in a plethora of different forms. Examples include MAR (Missing at random) and MNAR (Missing Not at Random) forms of randomness outlined in [12]. Another form, discussed in [15] was in the case sensors were unable to detect particularly large values, so that values above, say 1, are undetectable. Investigating the performance and potentially modifying our algorithms in these settings presents an opportunity for further analysis.

Appendices

Code Listings

We include here the primary modifications we make. The following code is the step function for the *missocd* algorithm.

```
ocd_updateMISS <- function(x_new, A, tail, tail_real, tail_full, beta, sparsity){
  p <- length(x_new)    # dimension of data
  a <- sqrt(2*log(p))  # hard threshold parameter
  L <- floor(log2(p))  # number of different scales
  tmp <- beta/sqrt(2^(0:(L+1)) * log2(2*p))
  B <- c(tmp, -tmp) # diadic grid of univariate alternative values
  unique_tail <- sort(unique(as.vector(tail_real)))
  unique_tail_fake <- sort(unique(as.vector(tail)))
  tail_loc <- matrix(match(as.vector(tail_real), unique_tail), p)
  x_toz<-x_new
  x_toz[is.na(x_toz)]<-0
  A <- A + x_toz
  tail <- tail + !is.na(x_new)
  tail_real <- tail_real + 1
  A_expand <- matrix(A[cbind(rep(1:p, length(B)), as.vector(tail_loc))], p)
```

```r
R <- t(t(A_expand) * B - t(tail) * B^2 / 2)
 tail <- tail * (R > 0)
 tail_real<-tail_real*(R>0)
 unique_tail_real_new <- sort(unique(as.vector(tail_real)))
 tail_real_loc <-match(as.vector(tail_real), unique_tail_real_new)
 unique_tail_new<-list()
 for(i in 1:p){
   temp<-as.vector(tail_real[i,])
   temp2<-as.vector(tail[i,])
   unique_tail_new[[i]]<-rep(1,length(unique_tail_real_new))
   temp_tail_loc<-match(temp,unique_tail_real_new)
   for(j in 1:length(temp2)){
     unique_tail_new[[i]][temp_tail_loc[j]]<-temp2[j]
   }
 }
 if (unique_tail_real_new[1] == 0){
   A <- A[, match(unique_tail_real_new[-1] - 1, unique_tail)]
   tail_full <- tail_full[, match(unique_tail_real_new[-1] - 1, unique_tail)]
   A <- cbind(0, A)
   tail_full <- cbind(0, tail_full)
   tail_full<-tail_full+!is.na(x_new)
 } else {
   A <- A[, match(unique_tail_real_new - 1, unique_tail)]
   tail_full <- tail_full[, match(unique_tail_real_new - 1, unique_tail)]
   tail_full<-tail_full+!is.na(x_new)
 }

G <- A^2/matrix(pmax(1,tail_full),p)
G0 <- G * (G > a^2)
colsum_dense <- colSums(G)
colsum_sparse <- colSums(G0)

 for (i in seq_along(unique_tail_real_new)) {
   colsum_dense[i] <- colsum_dense[i] - min(G[(rowSums(tail_real == unique_tail_real_new[i]) > 0), i])
   colsum_sparse[i] <- colsum_sparse[i] - min(G0[(rowSums(tail_real == unique_tail_real_new[i])) > 0, i])
 }
S_diag <- max(R, 0)
S_dense <- max(colsum_dense)
S_sparse <- max(colsum_sparse)
```

```
  stat <- setNames(c(S_diag, S_dense, S_sparse), c('diag', 'off_d', 'off_s'))
  if (sparsity=='dense') stat <- stat[-3]
  if (sparsity=='sparse') stat <- stat[-2]

  return(list(stat=stat, A=A, tail=tail,tail_real=tail_real,tail_full=tail_full))
}
```

The following codes show how we implement monte-carlo simulations for the different forms of missingness and our different solutions to tackling them:

```
MC_ocdMISS <- function(dim, patience, beta, sparsity, MC_reps, prob){
  peak_stat <- matrix(0, MC_reps, 3)
  colnames(peak_stat) <- c('diag','off_d','off_s')
  if (sparsity == 'sparse') peak_stat <- peak_stat[,-2]
  if (sparsity == 'dense') peak_stat <- peak_stat[,-3] S_diag, S_{off,d} and S_{off,s}
  for (rep in 1:MC_reps){
    A <- matrix(0, dim, 1)
    tail_full<-matrix(0, dim, 1)
    tail_real <- matrix(0, dim, floor(log2(dim))*2+4)
    tail <- matrix(0, dim, floor(log2(dim))*2+4)
    for (i in 1:patience){
      x_new <- rnorm(dim)
      num<-rep(1,dim)
      for(coords in 1:dim){
        num[coords]<-rbinom(1,1,prob[coords])
      }
      for(j in 1:length(num)){
        if(num[j]==1){
          x_new[j]<-NA
        }
      }
      ret <- ocd_updateMISS(x_new, A, tail, tail_real, tail_full, beta, sparsity)
      A <- ret$A; tail_real <- ret$tail_real; tail <- ret$tail; tail_full<-ret$tail_full
      peak_stat[rep,] <- pmax(peak_stat[rep,], ret$stat)
    }
  }
  thresh_est <- function(v) quantile(sort(v), exp(-1))
  th_individual <- apply(peak_stat, 2, thresh_est)
  th_multiplier <- thresh_est(apply(t(peak_stat)/th_individual, 2, max))
  th <- th_individual * th_multiplier
```

```r
  names(th) <- colnames(peak_stat)
  return(th)
}
```

```r
MC_ocdMI <- function(dim, patience, beta, sparsity, MC_reps,p,
    windowsize){
  peak_stat <- matrix(0, MC_reps, 3)
  colnames(peak_stat) <- c('diag','off_d','off_s')
  if (sparsity == 'sparse') peak_stat <- peak_stat[,-2]
  if (sparsity == 'dense') peak_stat <- peak_stat[,-3]
  for (rep in 1:MC_reps){
    A <- matrix(0, dim, 1)
    tail <- matrix(0, dim, floor(log2(dim))*2+4)
    for(i in 1:windowsize){
      x_new <- rnorm(dim)
      num<-rep(1,dim)
      for(coords in 1:dim){
        num[coords]<-rbinom(1,1,p[coords])
      }
      if(i<2){
        x_new <- rnorm(dim)
        window<-matrix(x_new,ncol=1)
      }else{
        for(j in 1:length(num)){
          if(num[j]==1){
            x_new[j]<-NA
          }
        }
        window<-cbind(window,c(x_new))
      }
    }
    winrow<-length(window[,1])
    winlen<-length(window[1,])
    lastaverages<-numeric(winrow)
    for(i in 1:winrow){
      lastaverages[i]<-mean(window[i,1:winlen-1],na.rm=TRUE)
    }
    for (i in 1:patience){
        x_new <- rnorm(dim)
        num<-rep(1,dim)
        for(coords in 1:dim){
          num[coords]<-rbinom(1,1,p[coords])
        }

        for(j in 1:length(num)){
          if(num[j]==1){
            x_new[j]<-NA
```

```
          }
        }
        window<-cbind(window[,2:windowsize],matrix(c(x_new)))
        imp <- meanImpute(window,lastaverages)
        x_new<-imp$imputed
        lastaverages<-imp$lastaverages
      ret <- ocd_update(x_new, A, tail, beta, sparsity)
      A <- ret$A; tail <- ret$tail
      peak_stat[rep,] <- pmax(peak_stat[rep,], ret$stat)
    }
  }
  thresh_est <- function(v) quantile(sort(v), exp(-1))
  th_individual <- apply(peak_stat, 2, thresh_est)
  th_multiplier <- thresh_est(apply(t(peak_stat)/th_individual,
    2, max))
  th <- th_individual * th_multiplier
  names(th) <- colnames(peak_stat)
  return(th)
}
```

```
MC_ocdblock <- function(dim, patience, beta, sparsity, MC_reps,p
    ,windowsize){
  peak_stat <- matrix(0, MC_reps, 3)
  colnames(peak_stat) <- c('diag','off_d','off_s')
  if (sparsity == 'sparse') peak_stat <- peak_stat[,-2]
  if (sparsity == 'dense') peak_stat <- peak_stat[,-3]
  S_diag, S_{off,d} and S_{off,s}
  for (rep in 1:MC_reps){
    A <- matrix(0, dim, 1)
    tail <- matrix(0, dim, floor(log2(dim))*2+4)
    for(i in 1:windowsize){
      x_new <- rnorm(dim)
      if(i<2){
        x_new <- rnorm(dim)
        window<-matrix(x_new,ncol=1)
      }else{
        window<-cbind(window,c(x_new))
      }
    }
    winrow<-length(window[,1])
    winlen<-length(window[1,])
    lastaverages<-numeric(winrow)
    for(i in 1:winrow){
      lastaverages[i]<-mean(window[i,1:winlen-1],na.rm=TRUE)
    }
    timer=0
    for (i in 1:patience){
```

```
      x_new <- rnorm(dim)

      btest<-rbinom(1,1,p)
      if(btest==1&&timer==0){
        btest<-rbinom(1,1,0.2)
        timer<-10
      }

      if(timer!=0){
        x_new[1:dim/2]<-NA
      }
      window<-cbind(window[,2:windowsize],matrix(c(x_new)))
      imp <- meanImpute(window,lastaverages)
      x_new<-imp$imputed
      lastaverages<-imp$lastaverages
      ret <- ocd_update(x_new, A, tail, beta, sparsity)
      A <- ret$A; tail <- ret$tail
      peak_stat[rep,] <- pmax(peak_stat[rep,], ret$stat)
      if(timer!=0){
        timer<-timer-1
      }
    }
  }
  thresh_est <- function(v) quantile(sort(v), exp(-1))
  th_individual <- apply(peak_stat, 2, thresh_est)
  th_multiplier <- thresh_est(apply(t(peak_stat)/th_individual,
     2, max))
  th <- th_individual * th_multiplier
  names(th) <- colnames(peak_stat)
  return(th)
}
```

Bibliography

References

[1] Jonas Dehning et al. "Inferring change points in the spread of COVID-19 reveals the effectiveness of interventions". In: *Science* 369.6500 (2020), eabb9789. DOI: 10.1126/science.abb9789. eprint: https://www.science.org/doi/pdf/10.1126/science.abb9789. URL: https://www.science.org/doi/abs/10.1126/science.abb9789.

[2] Vito M. R. Muggeo and Giada Adelfio. "Efficient change point detection for genomic sequences of continuous measurements". In: *Bioinformatics* 27.2 (Nov. 2010), pp. 161–166.

[3] Michelle Yu and Eric Ruggieri. "Change point analysis of global temperature records". In: *International Journal of Climatology* 39.8 (2019), pp. 3679–3688. DOI: https://doi.org/10.1002/joc.6042. URL: https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/joc.6042.

[4] Sayantan Banerjee and Kousik Guhathakurta. *Change-point Analysis in Financial Networks*. 2019. DOI: 10.48550/ARXIV.1911.05952. URL: https://arxiv.org/abs/1911.05952.

[5] Samaneh Aminikhanghahi, Tinghui Wang, and Diane J. Cook. "Real-Time Change Point Detection with Application to Smart Home Time Series Data". In: *IEEE Transactions on Knowledge and Data Engineering* 31.5 (2019), pp. 1010–1023. DOI: 10.1109/TKDE.2018.2850347.

[6] Scott Niekum et al. "Online Bayesian changepoint detection for articulated motion models". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 1468–1475. DOI: 10.1109/ICRA.2015.7139383.

[7] A.G. Tartakovsky et al. "A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods". In: *IEEE Transactions on Signal Processing* 54.9 (2006), pp. 3372–3382. DOI: 10.1109/TSP.2006.879308.

[8] Ajo-Franklin J.B., Dou S., and Lindsey N.J. et al. "Distributed Acoustic Sensing Using Dark Fiber for Near-Surface Characterization and Broadband Seismic Event Detection". In: *Scientific Reports* 9.1328 (2019). DOI: https://doi.org/10.1038/s41598-018-36675-8.

[9] Xu Z et al. "Signal detection using change point analysis in postmarket surveillance". In: *Pharmacoepidemiol Drug Saf* 24.6 (2019), pp. 663–668. DOI: doi:10.1002/pds.3783.

[10] H. Zheng and J. Boyce. "An improved UDP protocol for video transmission over Internet-to-wireless networks". In: *IEEE Transactions on Multimedia* 3.3 (2001), pp. 356–365. DOI: 10.1109/6046.944478.

[11] Chen Y, Wang T, and Samworth R J. "High-dimensional, multiscale online changepoint detection". In: *J. Roy. Statist. Soc., Ser. B, to appear* (2021+).

[12] Christina Mack, Zhaohui Su, and Daniel Westreich. *Managing Missing Data in Patient Registries: Addendum to Registries for Evaluating Patient Outcomes: A User's Guide*. 2018.

[13] Malte Londschien, Solt Kovács, and Peter Bühlmann. "Change-Point Detection for Graphical Models in the Presence of Missing Values". In: *Journal of Computational and Graphical Statistics* 30.3 (2021), pp. 768–779. DOI: 10.1080/10618600.2020.1853549. URL: https://doi.org/10.1080/10618600.2020.1853549.

[14] Wang T. and Samworth R. J. "High dimensional change point estimation via sparse projection". In: *J. Roy. Statist. Soc., Ser. B* 80.1 (2018), pp. 57–83.

[15] Follain B., Wang T., and R. J. Samworth. *High-dimensional changepoint estimation with heterogeneous missingness*. 2021. URL: https://arxiv.org/abs/2108.01525.

[16] D. S. Mitrinović, J. E. Pečarić, and A. M. Fink. *Classical and New Inequalities*. 1993.

[17] Lucien Birgé. "An Alternative Point of View on Lepski's Method". In: *Lecture Notes-Monograph Series* 36 (2001), pp. 113–133. ISSN: 07492170. URL: http://www.jstor.org/stable/4356108 (visited on 04/29/2022).

[18] Fan Chung and Linyuan Lu. “Concentration Inequalities and Martingale Inequalities: A Survey”. In: *Internet Mathematics* 3.1 (Jan. 1, 2006). DOI: 10.1080/15427951.2006.10129115.

[19] Iosif Pinelis. “Exact lower and upper bounds on the incomplete gamma function”. In: *Mathematical Inequalities & Applications* 23.4 (2020). DOI: dx.doi.org/10.7153/mia-2020-23-95.

[20] Christian Velasco-Gallego and Iraklis Lazakis. “Real-time data-driven missing data imputation for short-term sensor data of marine systems. A comparative study”. In: *Ocean Engineering* 218 (2020). DOI: https://doi.org/10.1016/j.oceaneng.2020.108261.