Towards Explainable Precision Changepoint Detection through Linear Decomposition

Taylor Dinkins
Oregon State University
Corvallis, Oregon, USA
dinkinst@oregonstate.edu

Sharmodeep Bhattacharyya
Oregon State University
Corvallis, Oregon, USA
Sharmodeep.Bhattacharyya@oregonstate.edu

Shirshendu Chatterjee
City University of New York
New York, New York, USA
shirshendu@ccny.cuny.edu

Sabrina Reis
University of Oregon
Eugene, Oregon, USA
sreis@uoregon.edu

Weng-Keen Wong
Oregon State University
Corvallis, Oregon, USA
wongwe@oregonstate.edu

ABSTRACT

We introduce a changepoint detection algorithm which uses a linear decomposition of the precision matrix to identify a change in the partial correlation structure of a time series. Our approach uses likelihood ratio tests to identify clusters of dimensions that are responsible for the change, thus providing more of an explanation as to why the changepoint occurs. We also greatly reduce the number of hypothesis tests needed to identify the relevant group of dimensions causing the change when compared to fully local methods that test each component of the precision matrix individually. We demonstrate the competitive accuracy and run-time of our approach using several simulation studies, and we present three real-world case studies using our approach.

CCS CONCEPTS

• **Mathematics of computing → Probabilistic algorithms**; **Multivariate statistics**.

KEYWORDS

changepoint detection, covariance, precision matrix, correlation, anomaly detection

ACM Reference Format:
Taylor Dinkins, Sharmodeep Bhattacharyya, Shirshendu Chatterjee, Sabrina Reis, and Weng-Keen Wong. 2022. Towards Explainable Precision Changepoint Detection through Linear Decomposition. In *Proceedings of Anomaly and Novelty Detection, Explanation and Accommodation (ANDEA '22).* ACM, New York, NY, USA, 6 pages.

1 INTRODUCTION

The goal of changepoint detection is to identify when a change occurs in the distribution that generates a sequence of data. The majority of changepoint detection methods focus on detecting changes to the mean of the distribution (e.g. see [9]). However, many use cases for real-world datasets require detecting changes in other distributional parameters. For instance, detecting changes in correlation between the prices of stocks is of great interest. The covariance matrix, and by extension its inverse (i.e. the precision matrix), can be used to detect changes in partial correlation, which is the focus of our work.

In order to apply precision changepoint detection to real-world data, two key issues need to be addressed. First, many types of real-world data are high-dimensional, which makes precision changepoint detection much more difficult. The number of entries in the precision matrix increases quadratically with increasing dimensionality, causing computational challenges in terms of running time and memory usage. High-dimensional precision matrices also introduce statistical challenges, including estimation [16] and hypothesis testing [12]. A common strategy for dealing with high-dimensional precision matrices is to assume that the precision matrix is sparse. This sparsity assumption has a natural consequence because a 0 entry at row i and column j of the precision matrix indicates that the random variables $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ are conditionally independent given all of the other variables.

A second issue involves explainability as changepoint detection algorithms typically only report the point in time when the change happens (referred to as *localization in time*). In the case of a covariance or precision changepoint, a more informative result is to also perform *localization in dimensions*. This additional form of localization reports which features (i.e. dimensions) are responsible for this change, thereby providing more of an explanation regarding the changepoint. This explanation is especially useful in domains involving scientific discovery, public safety, and finance.

Our work seeks to address these two issues by introducing a computationally efficient precision changepoint detection algorithm for high-dimensional data that can localize the change in both time and features. The key to our approach is to decompose the precision matrix into smaller blocks of related dimensions. In doing so, we can monitor these blocks for changes in precision, and thus make the changepoint detection more scalable and explainable.

2 RELATED WORK

Our discussion of related work will focus on changepoint detection methods that detect changes in the covariance or precision of a distribution. Some of the earliest methods focus on detecting changes to the variance of a univariate time series (e.g. [28, 17, 13]). For

ANDEA '22, August 14-18 2022, Washington DC

multivariate data, many approaches for covariance changepoint detection monitor the cumulative sum (CUSUM) statistic (e.g. [7, 2]), which accumulates the difference between the covariance matrix entries and a reference value representing the in-control state for the corresponding matrix entry. These techniques, however, do not scale to high-dimensions and a variety of strategies are needed to deal with large high-dimensional covariance matrices, including removing small entries of the covariance matrix [15], sparse Gaussian graphical models [31] and independent one-dimensional projections [27]. Another category of techniques look at changes in factor models, where these changes can be in the factor loadings or the number of factors [14, 4, 5]. Recent methods for precision changepoint detection are based on de-sparsified estimators [3] or sparse Gaussian graphical models [19, 18]. Finally, another closely related area is correlation changepoint detection, with a recent method using graph neural networks [32].

This work is also connected to hypothesis testing of covariance matrices [24, 25, 20, 10, 21], differential correlation matrices [11] and differential precision matrices [30]. As described in [12], *global* tests identify overall structural differences in the matrices while *local* tests look for differences in individual matrix entries, but are computationally expensive due to extensive hypothesis testing. Many global methods (e.g. [30, 18]) inspect each matrix entry but report the maximum value of the test statistic over the entries; these types of global tests can be easily converted to local tests by having them return the corresponding entry for the maximum-valued test statistic, but they still incur a large computational expense. None of the existing approaches address both localization in feature space as well as improved computational efficiency. Our work will fill this gap by addressing these two issues.

3 METHODOLOGY

3.1 Definitions

Let $\tilde{\boldsymbol{X}} = \{\tilde{\boldsymbol{X}}_0, ..., \tilde{\boldsymbol{X}}_T\}$, be a time series where each observation $\tilde{\boldsymbol{X}}_t \in \mathbb{R}^p$. First, we perform a series of preprocessing steps on $\tilde{\boldsymbol{X}}$ - specific to the dataset - such that we treat the result $\boldsymbol{X} = \{\boldsymbol{X}_0, ..., \boldsymbol{X}_T\}$ as an temporal ordering of independent, identically distributed, approximately Gaussian samples, where $\boldsymbol{X}_t \sim \mathcal{N}(\boldsymbol{0}, \Sigma_t), \forall t = 0, ..., T$. Let $\mathrm{S} = \frac{1}{T}\sum_{t=0}^{T} \boldsymbol{X}_t \boldsymbol{X}_t^\top$ be the sample covariance matrix for $\boldsymbol{X}$.

We aim to identify changepoints in time and localize these changepoints in dimensions of $\boldsymbol{X}$ in which the structure of the inverse covariance matrix (i.e. the precision matrix) $\boldsymbol{\Theta} = \Sigma^{-1}$ changes. More formally, a time k is regarded as a changepoint if

$$\boldsymbol{\Theta}_0 = ... = \boldsymbol{\Theta}_{k-1} \neq \boldsymbol{\Theta}_k = ... = \boldsymbol{\Theta}_T; \;\; 0 < k < T \tag{1}$$

The main idea behind our work is to use a linear precision model [1, 33] for $\boldsymbol{\Theta}$, which is given by

$$\Theta = \sum_{i=0}^{M} \alpha_i \cdot \boldsymbol{H}_i \tag{2}$$

Here, each basis matrix $\boldsymbol{H}_i \in \mathbb{R}^{p \times p}$ is symmetric. The set of basis matrices $\{\boldsymbol{H}_0, ..., \boldsymbol{H}_M\}$ are linearly independent, and there is assumed to be a set of scalar coefficients $\boldsymbol{\alpha} = \{\alpha_0, ..., \alpha_M\}$ such that $\sum_{i=0}^{M} \alpha_i \cdot \boldsymbol{H}_i$ is positive definite; here, we use the operator $\cdot$ to indicate an element-wise multiplication of the scalar α_i with each entry of the matrix $\boldsymbol{H_i}$. We additionally restrict $\{\boldsymbol{H}_0, ..., \boldsymbol{H}_M\}$ to be orthogonal for the purpose of interpretable localization of changes. Under these assumptions, specification of $\{\boldsymbol{H}_0, ..., \boldsymbol{H}_M\}$ in our approach remains somewhat flexible, and should be representative of the "stable" behavior of the data - such as sensors that operate together or stocks of the same sector. We assume that there is an early section in time of training data that can be used for informing the choice of each $\boldsymbol{H}_i$. Formally, $\exists k$ such that $\boldsymbol{\Theta}_0 = \boldsymbol{\Theta}_1 = ... = \boldsymbol{\Theta}_k$ where $0 < k < T$, which implies that the training window $\boldsymbol{X}_{train} = \{\boldsymbol{X}_0, ..., \boldsymbol{X}_k\}$ contains no changepoint.

3.2 Construction of Basis Matrices

We focus on a construction of $\{\boldsymbol{H}_0, ..., \boldsymbol{H}_M\}$ from data such that each $\boldsymbol{H}_i$ captures the strength of the partial correlation relationships present over specific clusters of dimensions in $\boldsymbol{X}$.

To specify these matrices, we first apply Graphical Lasso [22] to the training window $\boldsymbol{X}_{train}$, which minimizes

$$f(\boldsymbol{\Theta}) = tr(\mathrm{S}_{train}\boldsymbol{\Theta}) - log(det(\boldsymbol{\Theta})) + \lambda \sum_{j \neq k} det(\Theta_{jk}) \tag{3}$$

We convert the precision matrix estimate $\boldsymbol{\Theta}_{GL}$ from Graphical Lasso into a distance matrix $\boldsymbol{D}$ based on the magnitudes of these values, which is then used for single linkage clustering. This conversion is as follows, where $abs(\boldsymbol{A})$ indicates the elementwise absolute value of $\boldsymbol{A}$.

$$\begin{aligned} \boldsymbol{D} &= abs(\boldsymbol{\Theta}_{GL}) - max(abs(\boldsymbol{\Theta}_{GL})) \\ \mathrm{diag}(\boldsymbol{D}) &= \boldsymbol{0} \\ \boldsymbol{K} &= \text{single-linkage}(\boldsymbol{D}) \end{aligned} \tag{4}$$

Where $\boldsymbol{K}$ is cluster assignments for each dimension, produced by single linkage clustering. Finally, we use $\boldsymbol{K}$ and partition the dimensions into $M > 1$ distinct clusters. These partitions are then used to construct each $\boldsymbol{H}_i, i = 0, ..., M$, by populating $\boldsymbol{H}_i$ with all the values of $\boldsymbol{\Theta}_{GL}$ that correspond to the dimensions in the relevant cluster. Let $\boldsymbol{c}_i$ be a set containing the indices of dimensions in cluster i

$$\boldsymbol{H}_i[j,k] = \boldsymbol{\Theta}[j,k]; \;\; \forall j \in \boldsymbol{c}_i, \forall k \in \boldsymbol{c}_i \tag{5}$$

Using this approach, we can optionally perform further modifications of $\{\boldsymbol{H}_0, ..., \boldsymbol{H}_M\}$, such as by collecting all diagonal terms into a single diagonal matrices for isolating variances, adding extra matrices to model between cluster interactions, or adding $\epsilon > 0$ entries to capture induced changes for zeroed-out entries of $\boldsymbol{\Theta}_{GL}$. Overall, this framework provides a great deal of flexibility as to what types of changes can be monitored.

3.3 Estimation

Given the matrices $\{\boldsymbol{H}_0, ..., \boldsymbol{H}_M\}$, the Maximum Likelihood Estimates for the coefficients $\boldsymbol{\alpha}$ in the linear precision model $\boldsymbol{\Theta} = \sum_{i=0}^{M} \alpha_i \cdot \boldsymbol{H}_i$ can be found following several approaches [1, 33, 26, 23], using the log-likelihood function

$$\frac{2}{T} log(L(\boldsymbol{\alpha})) = tr[(\sum_{i=0}^{M} \alpha_i \cdot \boldsymbol{H}_i)\mathrm{S}] - log(det(\sum_{i=0}^{M} \alpha_i \cdot \boldsymbol{H}_i)) \tag{6}$$

For our purposes, we continue to use the lasso regularized objective under the assumption of sparsity in $\Theta = \sum_{i=0}^{M} \alpha_i \cdot \boldsymbol{H}_i$, with a λ value chosen high enough for stability on the data, and consistent

with the λ used in Equation 3.

$$f(\boldsymbol{X}, \boldsymbol{\alpha}) = \frac{2}{T} log(L(\boldsymbol{X}, \boldsymbol{\alpha})) = \\ tr[(\sum_{i=0}^{M} \alpha_i \cdot \boldsymbol{H_i})\mathrm{S}] - log(det(\sum_{i=0}^{M} \alpha_i \cdot \boldsymbol{H_i})) + \lambda \sum_{j \neq k} det(\Theta_{jk}) \tag{7}$$

We include a log-barrier term [8] in a simple iterative, first-order, sub-derivative solution to Equation 7 to encourage only solutions of $\boldsymbol{\alpha}$ where $\sum_{i=0}^{M} \alpha_i \cdot \boldsymbol{H_i}$ is positive definite. Note that by the construction of each H_i in Equations 4 and 5, a valid positive definite solution is one where $\boldsymbol{\alpha} = \mathbf{1} \in \mathbb{R}^M$, which is where we start the first iteration. Our learning rate is fixed at $\beta = 0.005$ over 100 iterations.

3.4 Changepoint Identification

We introduce our testing procedure for a setting of two distinct samples of data, where the separation between the two samples functions as a candidate changepoint in time. We test for changepoints as characterized by Equation 1 and we use the linear modeling assumption for $\boldsymbol{\Theta}$ in Equation 2. Then, we define a single *global* changepoint in $\boldsymbol{\Theta}$ by a change in *any* of the α_i coefficients of Equation 2. Let $\boldsymbol{\alpha}^t = \{\alpha_0^t, \ldots, \alpha_M^t\}$ be the set of coefficients at time t. Formally, k is a global changepoint if

$$\boldsymbol{\alpha}^0 = ... = \boldsymbol{\alpha}^{k-1} \neq \boldsymbol{\alpha}^k = ... = \boldsymbol{\alpha}^T \tag{8}$$

Instead of testing for global changepoints, we aim to test for *localized* changes in the clusters of dimensions corresponding to the specification of each $\boldsymbol{H_i}$ matrix, which we approach with multiple hypothesis tests, one for each coefficient α_i. For a given candidate point k and two temporally ordered samples of data $\boldsymbol{X}_{0:k}, \boldsymbol{X}_{k+1:T}$ we define the null hypothesis as

$$\mathbb{H}_0 : \ \boldsymbol{\alpha}^0 = ... = \boldsymbol{\alpha}^{k-1} = \boldsymbol{\alpha}^k = ... = \boldsymbol{\alpha}^T \tag{9}$$

We propose testing multiple alternative hypotheses of the form

$$\mathbb{H}_{A_i} : \ \alpha_i^0 = ... = \alpha_i^{k-1} \neq \alpha_i^k = ... = \alpha_i^T; \ \ i = 0, ..., M \tag{10}$$

In short, we test for whether a single coefficient α_i^A (over "All" the data $\boldsymbol{X}_{1:T}$), fits the data sufficiently when compared to two different coefficients α_i^L (for the "Left" part $\boldsymbol{X}_{0:k-1}$), and α_i^R (for the "Right" part $\boldsymbol{X}_{k:T}$).

In contrast to a global test on precision matrices of the form $\boldsymbol{\Theta_0} = \boldsymbol{\Theta_1}$, our tests localize changes to clusters of dimensions that interact with a single scalar α_i. While this may seem a restrictive assumption when compared to a fully local test of the form $\Theta_{i,j,1} = \Theta_{i,j,2}$, for all $0 \leq i \leq j \leq p$, we greatly reduce the number of hypothesis tests, and provide easier interpretability in higher dimensional settings where the user has access to the set of dimensions in each cluster. Our approach lies somewhere in between the fully global and fully local regimes, and relies on testing the Maximum Likelihood Estimates of this reduced number of coefficients $M << p$, instead of the p regression coefficients for each dimension on each sample[30].

We make use of the likelihood ratio as our test statistic, in accordance with Wilks' theorem[29]. The likelihood ratio for a single window is given by

$$W_i^k = \frac{\sup_{\boldsymbol{\alpha}^A \in \mathbb{R}^M} \prod_{t=0}^{T} f(\boldsymbol{X}_t, \boldsymbol{\alpha}^A)}{\sup_{\alpha_i^L, \alpha_i^R \in \mathbb{R}^M} \prod_{t=0}^{k} f(\boldsymbol{X}_t, \boldsymbol{\alpha}_{-i}^A \cup \alpha_i^L) \prod_{t=k+1}^{T} f(\boldsymbol{X}_t, \boldsymbol{\alpha}_{-i}^A \cup \alpha_i^R)} \tag{11}$$

Where $f(\boldsymbol{X}_t, \boldsymbol{\alpha})$ is given in Equation 7, and $\boldsymbol{\alpha}_{-i}^A$ indicates the exclusion of the ith coefficient of $\boldsymbol{\alpha}^A$ from the set. Our test statistic is calculated as

$$T_i^k = -2ln(W_i^k) \tag{12}$$

To identify candidate changepoints, we take a simple sliding window approach with a window size w and stepsize s, and repeatedly perform this testing process in a temporally ordered manner to collect test statistics W_i^k for all $i = 0, ..., M$ and all $k = 0, ..., T$ on the full data $\boldsymbol{X} = \{\boldsymbol{X}_0, ..., \boldsymbol{X}_T\}$. Our algorithm can be applied in an online or offline setting, under the assumption that the changepoint lies within the range $[0 + w, T - w]$. In both settings, we would identify a change if $T_i^k \geq \tau$, for some threshold τ.

Following Wilks' theorem, we use the chi-squared approximation to compute empirical p-values for all i. Due to multiple hypothesis testing, we adjust these M p-values at each candidate point k using the Benjamini-Hochberg[6] False Discovery Rate procedure (FDR-BH) and perform the thresholding based on the desired FDR-BH significance level α_{FDR-BH}. In the offline setting, assuming a single changepoint is present in any given window partitioning of the data, we can simply check the maximal values of the test statistic curve, spaced at least $2w$ apart. For an online application, we slide the window forward in time and compute the test statistic each time, but we return a detected changepoint if ever we exceed the significance level. See Algorithm 1 for an example application of the method to data with no returns at detected changepoints.

Algorithm 1:

Input : $\boldsymbol{X} = \{\boldsymbol{X}_0, ..., \boldsymbol{X}_T\}, \lambda, M, w, s, \alpha_{FDR-BH}$
Output: (T_i^k, p_i^k); $(0 \leq i \leq M;\ 0 + w \leq k \leq T - w)$
Collect $\hat{\boldsymbol{\Theta}}_{GL}$ - (Equation 3);
Construct each $\boldsymbol{H_i}$ - (Equations 4 and 5);
for $0 + w \leq k \leq T - w$, *with step-size* s **do**
  Partition $\boldsymbol{X}^L = \boldsymbol{X}_{(k-w):k}$, $\boldsymbol{X}^R = \boldsymbol{X}_{(k+1):(k+w)}$;
  Estimate $\boldsymbol{\alpha}^A$ from $\boldsymbol{X}_{(k-w):(k+w)}$ - (Equation 11);
  for $0 \leq i \leq M$ **do**
    Estimate α_i^L, α_i^R from $\boldsymbol{X}^L, \boldsymbol{X}^R$ - (Equation 11);
    Calculate T_i^k - (Equations 11 and 12);
    Calculate p_i^k as $\chi^2(T_i^k, 2)$;
  end
  Adjust $\boldsymbol{p^k}$ with FDR-BH(α_{FDR-BH},$\boldsymbol{p^k}$);
end

4 RESULTS AND DISCUSSION

4.1 Simulations

To assess the performance of our algorithm, we generate a suite of simulated changepoints and compare to the differential network

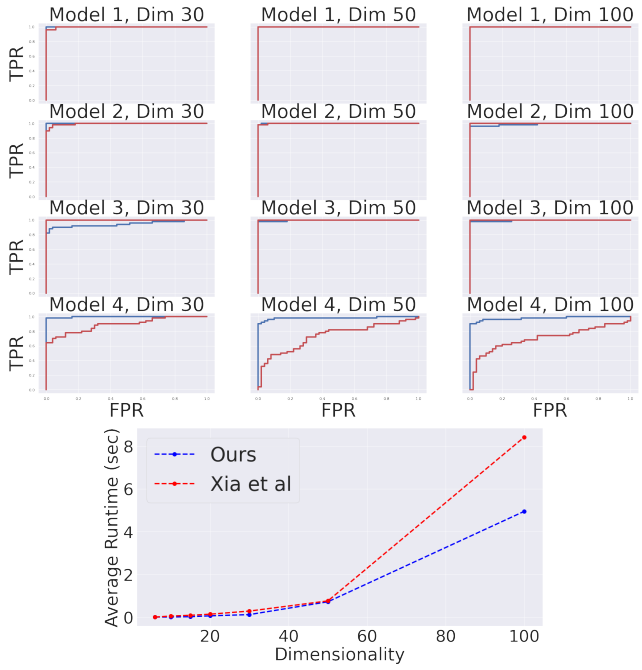

Figure 1: ROC curves on simulated matrix models (top) and empirical Runtime Comparison (bottom).

change detection algorithm from [30][12]. Both implementations are in Python, and we make use of RPY2[1] to interface with SCALREG[2] in R for estimation in [30]. We utilize two matrix models from [30], the Cholesky decomposition, and simulations from the linear precision model given by Equation 2. These models are summarized as follows. First, let $\boldsymbol{D}_{p\times p} = \text{diag}(\text{Uniform}(0.5, 2.5))$.

Model 1. $\boldsymbol{\Theta}^*_{p\times p} = \theta^*_{i,j}$ where $\theta^*_{i,i} = 1; \theta^*_{i,i+1} = \theta^*_{i+1,i} = 0.6; \theta^*_{i,i+2} = \theta^*_{i+2,i} = 0.3$; and $\theta^*_{i,j} = 0$ otherwise. Let $\boldsymbol{\Theta}_1 = \boldsymbol{D}^{\frac{1}{2}}\boldsymbol{\Theta}^*\boldsymbol{D}^{\frac{1}{2}}$.

Model 2. $\boldsymbol{\Theta}^*_{p\times p} = \theta^*_{i,j}$ where $\theta^*_{i,i} = 1; \theta^*_{i,j} = 0.8 \times \text{Bernoulli}(1, 0.05)$ for $i < j$; and $\theta^*_{j,i} = \theta^*_{i,j}$. Let $\boldsymbol{\Theta}_2 = \boldsymbol{D}^{\frac{1}{2}}(\boldsymbol{\Theta}^* + \delta\boldsymbol{I})/(1+\delta)\boldsymbol{D}^{\frac{1}{2}}$, where $\delta = |\lambda_{min}(\boldsymbol{\Theta}^*)| + 0.05$.

To simulate a change in Models 1 and 2, we follow the procedure in [30] and create $\boldsymbol{U}_{p\times p}$ as an upper triangular matrix with 4 unique non-zero entries randomly sampled from above the diagonal, and reflected for symmetry. The magnitude of each of these values is randomly and uniformly sampled from the ranges

$$[-2\omega log(\frac{p}{n})^{\frac{1}{2}}, -\omega log(\frac{p}{n})^{\frac{1}{2}}] \cup [\omega log(\frac{p}{n})^{\frac{1}{2}}, 2\omega log(\frac{p}{n})^{\frac{1}{2}}] \quad (13)$$

Where $\omega = \max(\text{diag}(\boldsymbol{\Theta}))$. We create $\boldsymbol{\Theta}^1 = \boldsymbol{\Theta} + \delta I, \boldsymbol{\Theta}^2 = \boldsymbol{\Theta} + \boldsymbol{U} + \delta\boldsymbol{I}$, with $\delta = |\lambda_{min}(\boldsymbol{\Theta}^*)| + 0.05$. Then, we sample data from $\mathcal{N}(\boldsymbol{0}, (\boldsymbol{\Theta}^1)^{-1})$ and $\mathcal{N}(\boldsymbol{0}, (\boldsymbol{\Theta}^2)^{-1})$.

Model 3. $\boldsymbol{L}_{p\times p} = \text{tril}(\text{random pos-def matrix})$, where $\text{tril}(\boldsymbol{A})$ indicates the lower-triangle of $\boldsymbol{A}$. Let $\boldsymbol{\Theta}^1 = \boldsymbol{L}\boldsymbol{L}^\top$. We randomly sample 2 unique entries in the lower triangle of $\boldsymbol{L}$ and add a magnitude τ to the entries sampled as in Equation 13 with $n = 50$, to create $\tilde{\boldsymbol{L}}$; $\boldsymbol{\Theta}^2 = \tilde{\boldsymbol{L}}\tilde{\boldsymbol{L}}^\top$. Likewise, we sample data from $\mathcal{N}(\boldsymbol{0}, (\boldsymbol{\Theta}^1)^{-1})$ and $\mathcal{N}(\boldsymbol{0}, (\boldsymbol{\Theta}^2)^{-1})$.

Model 4. Using the model $\boldsymbol{\Theta} = \sum_{i=0}^{M} \alpha_i \cdot \boldsymbol{H}_i$, we randomly sample $\boldsymbol{\alpha} = \text{Uniform}(0, 1, M)$, specify $\{\boldsymbol{H}_0, ..., \boldsymbol{H}_M\}$ by sampling $\boldsymbol{A}_{p\times p} =$ (spd matrix), and randomly populate each $\boldsymbol{H}_i$ with $\lfloor \frac{p}{M} \rfloor$ entries each, in the same locations as $\boldsymbol{A}$, yielding $\boldsymbol{\Theta}^1$. Then, we increase or decrease a randomly selected α_i by the scale 0.8, yielding $\boldsymbol{\Theta}^2$. We sample data from $\mathcal{N}(\boldsymbol{0}, (\boldsymbol{\Theta}^1)^{-1})$ and $\mathcal{N}(\boldsymbol{0}, (\boldsymbol{\Theta}^2)^{-1})$. We let M scale with the dimensionality, with $M = [5, 10, 15]$.

In all simulations, we perform window-to-window comparisons of size $w = 100$ averaged over 50 seeds to calculate ROC curves, shown in Figure 1 (top), and define a true positive for our algorithm by taking the maximum value of Equation 12 over all i, which we compare to the global test from [30]. We simulate dimensionality $p = [30, 50, 100]$ for ROC, and $p = [6, 10, 20, 30, 50, 100]$ for runtime, each with 50 unique seeds, and perform an identical simulation process for data containing no changepoints from each model. In our algorithm, for models Model 1, 2, and 3, we set $M = [10, 15, 20]$ in scaling with the dimensionality. For Model 4, we set M as specified, but with no access to the simulation settings of $\boldsymbol{\alpha}$ or $\{\boldsymbol{H}_0, ..., \boldsymbol{H}_M\}$. We let $\lambda = 0.1$ for the stability of lasso. Lastly, we include a runtime comparison in Figure 1 (bottom) of a single window-to-window comparison averaged over the seeds at each dimension, highlighting the relative performance of our method.

4.2 Real World Case Studies

We apply our algorithm to three real-world datasets as described below. We make the assumptions of a single changepoint in any window-to-window comparison, and that thresholded (zero) entries of $\boldsymbol{\Theta}_{GL}$ indicate interactions that we do not wish to explicitly model.

Earthquake Detection. UNAVCO[3] streams real-time Global Navigation Satellite System (GNSS) data which can be used to detect travelling ionosphere disturbances due to earthquakes. We include results from two case studies. The first contains transmissions of detrended total electron content (dTEC) from 5 stations on the Aleutian Islands to a passing satellite, and are used to detect the occurrence of the Alaskan earthquake on January 23, 2018. The second contains readings from 6 stations in Tohoku, Japan for the earthquake that took place on March 11, 2011. Although these datasets are low dimensional, they serve to illustrate the localization aspects of our algorithm. We verified our results with domain experts for this data.

We present test statistic results for our algorithm applied to the Alaskan earthquake with $M = 2, \lambda = 0.1, w = 50, s = 1$, and the training window comprising the first 10% of the data in Figure 2 (top). For this event, we separate an additional matrix $\boldsymbol{H}_2$, which contains all the diagonal entries, to illustrate the isolation of the (inverse) variance changes. One set of two stations (those in Cluster 1) are close in spatial location and thus do not experience a partial correlation change during the earthquake, as these two stations are affected in tandem. In contrast, the 3 sensors in Cluster 0 experience such a change. Our results for the Tohoku earthquake with $M = 3, \lambda = 0.1, w = 50, s = 1$, and the training window comprising the

[1] https://rpy2.github.io/
[2] https://cran.r-project.org/web/packages/scalreg/
[3] https://www.unavco.org/data/gps-gnss/real-time/real-time.html

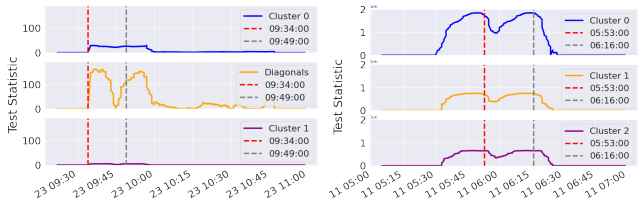

Figure 2: Test Statistics for Alaska 2018 (left) and Tohoku 2011 (right) earthquakes with earthquake start/end times. Two peaks occur approximately when the earthquake begins and ends.

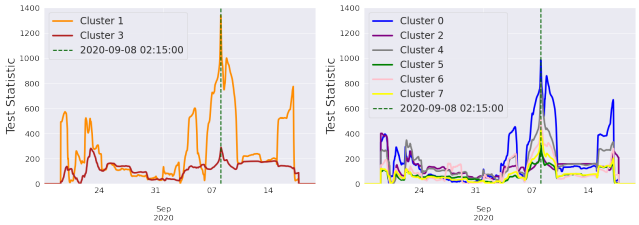

Figure 3: Holiday Farm Fire test statistics for two clusters of sensors (left) and remaining clusters (right). The highest peak generally occurs at the fire onset.

first 25% of the data are similarly shown in Figure 2 (bottom). Here, we do not separate the diagonal entries.

Wildfire Detection. The PurpleAir sensor network[4] uses laser counters to measure airborne particulate matter. We use the mass concentrations of $PM_{2.5}$ registered by the sensors to detect wildfire presence. While wildfires cause a sudden shift in the mean of the $PM_{2.5}$, we observe empirically that wildfires also cause sudden, abnormal correlations between the sensors present in the smoke fallout across spatial distance.

We focus on the Holiday Farm fire that took place east of Eugene, OR in September, 2020. Our data consists of the date range January 1, 2020 - December 31, 2021, with a sampling rate of 15 minutes, with 14 sensors that provided consistent readings through this region; we did not use data from other sensors that had severe data quality issues. We use the period up through July 2020 as our training data for preprocessing and specification of $\{\boldsymbol{H}_0, ..., \boldsymbol{H}_M\}$. We process the data by linearly interpolating missing sensor readings, adding a small ϵ value to readings of $PM_{2.5} = 0$, and taking the log-transformation of the original data. We scale this result to have mean 0, variance 1 using the summary statistics from the training window. Overall, clusters for $\{\boldsymbol{H}_0, ..., \boldsymbol{H}_M\}$ correspond to spatial locations. We isolate the summer period for generating test curves using the sliding window approach, without breaks. For this dataset, we choose $M = 8$, $\lambda = 0.05$, $w = 100$, $s = 1$.

Figure 3 plots the test statistic curves from two distinct clusters of sensors. Cluster 1 consists of three sensors - two of which are in close proximity (Crescent Park) - and one a slight distance away (Springfield HS). Cluster 2 consists of two sensors (Office and Office B). The peak in the statistic for Cluster 1 is far more pronounced, and aligns with the major onset of smoke drifting into the region. We hypothesize that Cluster 1 is able to capture this change due to the delay in smoke arrival, which modifies the pairwise, off-diagonal entries of $\boldsymbol{H}_1$ between the two sensors located farther north, and the single sensor in this cluster which is farther south.

Detection of the COVID-19 Pandemic from Stock Prices. We conclude with an example of our algorithm applied to a (nearly) global changepoint problem, to highlight that we both detect the pandemic, and are able to isolate interesting behaviors of individual dimensions, without the need for fully local methods. Here, we collect a set of closing prices of 41 stocks across various industries: ['AAL', 'AAPL', 'ALK', 'AMZN', 'ASML', 'AXP', 'BA', 'CLX', 'CSCO', 'DB', 'DIS', 'ETSY', 'FB', 'GIS', 'GOOG', 'INTC', 'JPM', 'K', 'LUV', 'MA', 'MCHP', 'MS', 'MSFT', 'NCLH', 'NFLX', 'NVDA', 'NXPI', 'ON', 'ORCL', 'PEP', 'PFE', 'PG', 'PYPL', 'RCL', 'SPY', 'STM', 'TSLA', 'TSM', 'TWTR', 'VZ', 'WFC'].

This data is collected from February 22, 2017 - February 19, 2022, at daily intervals, using parameter settings $M = 10, \lambda = 0.5, w = 100$, and $s = 1$. We preprocess the data by taking the logarithm of the daily stock returns, followed by standard mean 0 scaling. We use the first 40% of the data as the training data window.

Figure 4 shows a comparison of two test statistic curves, one containing two clusters, and the other containing all others. Nearly all curves have a pronounced peak in at January 8, 2020. This aligns with the fact that January 8, 2020 was the first date the Center for Disease Control announced a health advisory regarding COVID-19. In Figure 4 (left), the lower curve in red ('CLX') does not have as dramatic a shift on January 8 as the blue curve ('MSFT'). The Clorox company ('CLX') was placed in its own cluster during construction of $\{\boldsymbol{H}_0, ..., \boldsymbol{H}_M\}$. Our algorithm's results suggest it was less affected by the COVID-19 pandemic overall for the $w = 100$ window positioning on January 8, 2020. Figure 5 draws a vertical line for this date and illustrates the greater degree of spread for the log-returns of 'MSFT' (Microsoft) immediately after January 8 compared to 'CLX'.

To summarize, our algorithm was able to detect both the general pattern of consistent changes present at the start of the COVID-19 pandemic, and our cluster localization process provided a convenient and interpretable format for identifying a surprising pattern in this data. This highlights one strength of our approach to localization when analyzing changepoints of this nature.

5 CONCLUSION

We have shown that a linear decomposition of the precision matrix can be used to identify groups of features that are responsible for the change in a precision matrix. In addition, we have shown that the running time of our approach is faster than a fully local method that inspects all entries of the precision matrix. For future work, we will improve the localization by drilling down further into the clusters of features to identify smaller subsets of features responsible for the change, perform detection time analysis over different thresholds, collect additional high dimensional real-world datasets for comparison with [30] and [18], and we will improve the running time and estimation method of our approach.

[4] https://www2.purpleair.com/

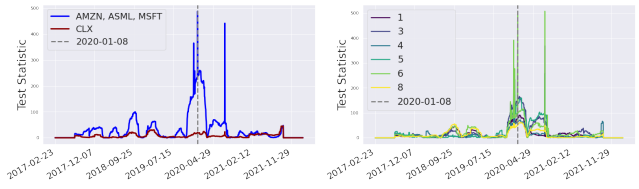

Figure 4: Test statistics for two stock clusters (left) and the remaining clusters (right). Notice that the singular cluster CLX in (left), unlike the other clusters, does not have a pronounced spike on January 8, 2020 that would be consistent with a change in precision.

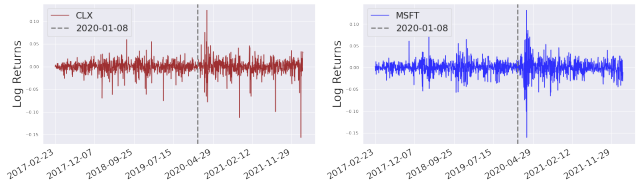

Figure 5: Log Returns of CLX (left) and MSFT (right), with the most common peak of January 8, 2020.

ACKNOWLEDGMENTS

This research was partially funded through NSF grant CNS-1941892 and the Industry-University Cooperative Research Center on Pervasive Personalized Intelligence. We thank Rahul Khanna, Giuseppe Raffa and Jose Lopez for their guidance and constructive comments. We also thank Fiona Luhrmann and Jihye Park for their expertise in analyzing the real-time GNSS data.

REFERENCES

[1] T. W. Anderson. 1970. Estimation of covariance matrices which are linear combinations or whose inverses are linear combinations of given matrices. In *Essays in probability and statistics*. University of North Carolina Press Chapel Hill, NC, 1–24.

[2] A. Aue, S. Hörmann, L. Horváth, and M. Reimherr. 2009. Break detection in the covariance structure of multivariate time series models. *The Annals of Statistics*, 37, 6B, (Dec. 2009), 4046–4087. DOI: 10.1214/09-AOS707.

[3] V. Avanesov and N. Buzun. 2018. Change-point detection in high-dimensional covariance structure. *Electronic Journal of Statistics*, 12, 2, 3254–3294.

[4] J. Bai, X. Han, and Y. Shi. 2020. Estimation and inference of structural changes in high dimensional factor models. *Journal of Econometrics*, 219, 1, 66–100.

[5] M. Barigozzi, H. Cho, and P. Fryzlewicz. 2018. Simultaneous multiple change-point and factor analysis for high-dimensional time series. *Journal of Econometrics*, 206, 1, 187–225. DOI: 10.1016/j.jeconom.2018.05.003.

[6] Y. Benjamini and Y. Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57, 1, 289–300.

[7] O. Bodnar and W. Schmid. 2017. Cusum control schemes for monitoring the covariance matrix of multivariate time series. *Statistics*, 51, 4, 722–744. DOI: 10.1080/02331888.2016.1268616.

[8] S. P. Boyd and L. Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

[9] E. Brodsky and B. S. Darkhovsky. 2013. *Nonparametric methods in change point problems*. Vol. 243. Springer Science & Business Media.

[10] T. Cai, W. Liu, and Y. Xia. 2013. Two-sample covariance matrix testing and support recovery in high-dimensional and sparse settings. *Journal of the American Statistical Association*, 108, 501, 265–277.

[11] T. T. Cai and A. Zhang. 2016. Inference on high-dimensional differential correlation matrices. *Journal Multivariate Analysis*, 143, 107–126.

[12] T. Tony Cai. 2017. Global Testing and Large-Scale Multiple Testing for High-Dimensional Covariance Structures. *Annual Review of Statistics and Its Application*, 4, 1, 423–446. DOI: 10.1146/annurev-statistics-060116-053754.

[13] J. Chen and A. K. Gupta. 1997. Testing and locating variance changepoints with application to stock prices. *Journal of the American Statistical Association*, 92, 438, 739–747.

[14] X. Cheng, Z. Liao, and F. Schorfheide. 2016. Shrinkage estimation of high-dimensional factor models with structural instabilities. *The Review of Economic Studies*, 83, 1511–1543.

[15] H. Dette, G. Pan, and Q. Yang. 2020. Estimating a Change Point in a Sequence of Very High-Dimensional Covariance Matrices. *Journal of the American Statistical Association*, 0, 0, (July 2020), 1–11.

[16] J. Fan, Y. Liao, and H. Liu. 2016. An overview of the estimation of large covariance and precision matrices. *The Econometrics Journal*, 19, 1, C1–C32.

[17] Carla Inclan and George C. Tiao. 1994. Use of cumulative sums of squares for retrospective detection of changes of variance. *Journal of the American Statistical Association*, 89, 427, 913–923.

[18] Hossein Keshavarz and George Michailidis. 2020. Online detection of local abrupt changes in high-dimensional gaussian graphical models. *arXiv preprint arXiv:2003.06961*.

[19] Hossein Keshavarz, George Michailidis, and Yves Atchadé. 2018. Sequential change-point detection in high-dimensional gaussian graphical models. *arXiv preprint arXiv:1806.07870*.

[20] Jun Li and Song Xi Chen. 2012. Two sample tests for high-dimensional covariance matrices. *The Annals of Statistics*, 40, 2, 908–940.

[21] Weiming Li and Yingli Qin. 2014. Hypothesis testing for high-dimensional covariance matrices. *Journal of Multivariate Analysis*, 128, 108–119.

[22] Rahul Mazumder and Trevor Hastie. 2011. The graphical lasso: new insights and alternatives. (2011). DOI: 10.48550/ARXIV.1111.5479.

[23] D. B. Rubin and T. H. Szatrowski. 1982. Finding maximum likelihood estimates of patterned covariance matrices by the em algorithm. *Biometrika*, 69, 3, 657–660.

[24] J. R. Schott. 2007. A test for the equality of covariance matrices when the dimension is large relative to the sample sizes. *Computational Statistics and Data Analysis*, 51, 12, 6535–6542.

[25] M. S. Srivastava and H. Yanagihara. 2010. Testing the equality of several covariance matrices with fewer observations than the dimension. *Journal of Multivariate Analysis*, 101, 6, 1319–1329.

[26] B. Sturmfels, S. Timme, and P. Zwiernik. 2020. Estimating linear covariance models with numerical nonlinear algebra. *Algebraic Statistics*, 11, 1, 31–52.

[27] D. Wang, Y. Yu, and A. Rinaldo. 2021. Optimal covariance change point localization in high dimensions. *Bernoulli*, 27, 1, 554–575. Retrieved Sept. 5, 2021 from.

[28] D. W. Wichern, R. B. Miller, and D-A. Hsu. 1976. Changes of variance in first-order autoregressive time series models-with an application. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 25, 3, 248–256.

[29] S. S Wilks. 1938. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The annals of mathematical statistics*, 9, 1, 60–62.

[30] Y. Xia, T. Cai, and T. T. Cai. 2015. Testing differential networks with applications to detecting gene-by-gene interactions. *Biometrika*, 102, 247–266.

[31] X. Xuan and K. Murphy. 2007. Modeling changing dependency structure in multivariate time series. In *Proceedings of the 24th international conference on Machine learning*. Association for Computing Machinery, 1055–1062.

[32] R. Zhang, Y. Hao, D. Yu, W-C. Chang, G. Lai, and Y. Yang. 2020. Correlation-Aware Change-Point Detection via Graph Neural Networks. In *Neural Information Processing* (Lecture Notes in Computer Science). H. Yang, K. Pasupa, A. C-S. Leung, J. T. Kwok, J. H. Chan, and I. King, (Eds.) Springer International Publishing, 555–567.

[33] P. Zwiernik, C. Uhler, and D. Richards. 2017. Maximum likelihood estimation for linear gaussian covariance models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79, 4, 1269–1292.