Monitoring network changes in social media

Cathy Yi-Hsuan Chen [*] Yarema Okhrin [†] Tengyao Wang [‡]

July 27, 2021

Abstract

Econometricians are increasingly working with high-dimensional networks and their dynamics. Econometricians, however, are often confronted with unforeseen changes in network dynamics. In this paper, we develop a method and the corresponding algorithm for monitoring changes in dynamic networks. We characterize two types of changes, edge-initiated and node-initiated, to feature the complexity of networks. The proposed approach accounts for three potential challenges in the analysis of networks. First, networks are high-dimensional objects causing the standard statistical tools to suffer from the curse of dimensionality. Second, any potential changes in social networks are likely driven by a few nodes or edges in the network. Third, in many dynamic network applications such as monitoring network connectedness or its centrality, it will be more practically applicable to detect the change in an *online* fashion than the offline version. The proposed detection method at each time point projects the entire network onto a low-dimensional vector by taking the sparsity into account, then sequentially detects the change by comparing consecutive estimates of the optimal projection direction. As long as the change is sizeable and persistent, the projected vectors will converge to the optimal one, leading to a jump in the sine angle distance between them. A change is therefore declared. Strong theoretical guarantees on both the false alarm rate and detection delays are derived in a sub-Gaussian setting, even under spatial and temporal dependence in the data stream. Numerical studies and an application to the social media messages network support the effectiveness of our method.

Keywords: Change point, network, CUSUM, sparsity, social media

[*]Adam Smith Business School, University of Glasgow, UK; IRTG 1792 High Dimensional Non Stationary Time Series, Humboldt-Universität zu Berlin (*e-mail: CathyYi-Hsuan.Chen@glasgow.ac.uk*)

[†]Department of Statistics, University of Augsburg, Germany (*e-mail: yarema.okhrin@wiwi.uni-augsburg.de*)

[‡]Department of Statistical Science, University College London, UK (*e-mail: tengyao.wang@ucl.ac.uk*)

1 Introduction

Modeling network and its dynamics has been an active area of modern statistics and econometrics research, partly due to the increasing popularity of network data generated e.g. from social media and communication networks. Literature in this area is abundant (Jochmans, 2018; Zhu et al., 2017; Han et al., 2021), with some involving the modelling and analysis of very large-scale networks (Chen et al., 2021). However, so far, relatively little emphasis has been placed on detecting changes in dynamic networks. An overlook of a potential change of networks may come at risk; the network has different structures if there is a change and the models based on pre-change network data are no longer applicable in prediction or risk management tasks. More importantly, any change of the network structure may signify a change of connectivity between nodes or a replacement of key players, also known as central nodes, within the network.

The problem of detecting changes in a dynamic network falls into the broad area of *change point analysis*, where the goal is to identify a change in the data generating mechanism in a data stream. Monitoring a univariate time series has been well studied under the banner of statistical process control (e.g. Duncan, 1952; Page, 1954; Barnard, 1959). Motivated by applications, there has been an increased interest in developing change point detection and estimation procedures in multivariate or even high-dimensional data sets. Examples include detection methods for changes in the mean structure (Zhang et al., 2010; Horváth and Hušková, 2012; Enikeeva and Harchaoui, 2019; Cho and Fryzlewicz, 2015; Jirak, 2015; Zou et al., 2015; Cho, 2016; Wang and Samworth, 2018; Chen et al., 2020) and for changes in the covariance structure Lavielle and Teyssiere (2006); Aue et al. (2009); Bücher et al. (2014); Preußet al. (2015); Wang et al. (2019); Dette et al. (2020) of the data. Detecting or monitoring a change in the network is even more methodologically challenging than the conventional detecting practice applied to the univariate or multivariate process. The connectivity structure of the network exhibits a time-varying feature. Network modeling strives to characterize the interactions within the network and their dynamics. The curse of dimensionality, which also happens to general change points detection tasks, exacerbates in a network given network's complexity and the featured interactions between nodes. Recently, Wang et al. (2021) studied the problem of change point detection in a

dynamic network with binary edge weights. Their method and analysis have focused on the so-called *offline* problem, requiring full knowledge of the entire history of the network dynamics before retrospectively identifying the change point locations. However, in many dynamic network applications such as monitoring network connectedness or its centrality, it will be more practically relevant to detect a change in an *online* fashion than the offline version as data are sequentially observed and most importantly, any change upon the new arrival observation needs to be scrutinized.

This paper presents a novel method for detecting changes in a dynamic network sequentially over time. The proposed method does not rely on direct estimation of the network at each time point but rather tracks the evolvement of networks over time. In other words, regardless of the complexity of the original network, the procedure works as long as the change is sufficiently structured. Depending on the nature of the network (e.g. directed or undirected) and the nature of the change (e.g. whether all edges incident on the same node change together), we propose a *degree-based* and also an *edge-based* scheme to tackle the variety of changes. Motivated by Wang and Samworth (2018), our key idea is to represent the dynamic network as a high-dimensional time series and estimate the optimal projection directions that can best aggregate the data into a one-dimensional series with the largest signal-to-noise ratio when a change has occurred. Given the sequential arrival of observation, we construct a sequence of the estimators of the projection directions. We declare a change when a sufficiently large number of consecutive projection direction estimators are well-aligned with each other.

More specifically, in line with Wang et al. (2021), a sparsity assumption underpins our framework. The changes are govern by a small subset of nodes or edges, which translates to a sparse change in the data vectors. The imposed sparsity allows us to consistently estimate an optimal projection direction from a high-dimensional network vector space. This machinery is iterated for every new arrival observation and delivers a sequence of projection vectors. The proposed sine angle distance is used to measure the alignment between the consecutive estimates. A change is declared when a sufficient number of such sine angle distances all fall below the specified threshold. The Network Sequential Monitoring (**NSM**) algorithm is proposed to delineate this procedure. The theoretical

justification and support for the invented method and algorithm are provided, including the choice of threshold and tail sequence in a consideration of false alarm rate. Theoretical guanrantees are also derived in the more general sub-Gaussian setting, even under spatial and temporal dependence in the data stream.

In the simulation section, we extensively simulate the network dynamics under different natures of changes and network structures. We evaluate the performance of the proposed methods and confirm their effectiveness in terms of false alarm rate and the length of the delay. We end up with the following tactical guidance for detection strategies. As long as the magnitude of the change is sizable, both the degree-based and the edge-based methods are comparable in terms of the average detection delay. On the other hand, for the changes that are edge-sparse (i.e. only a small number of edges change) or heterogeneous (i.e. changes in different edges can have different signs), a degree-based detection scheme suffers a longer detection delay, which is potentially caused by an offsetting effect between positive and negative shifts. In such a scenario, the edge-based detection method is recommended.

To further illustrate the utility of our procedure, we apply the proposed method and monitoring algorithm to social media networks. The social media networks are dynamic and sparse, reflecting a changing interaction and few opinion leaders in a social network. Particular attention is placed on the 'cryptocurrency bubble' during the end of 2017 and the beginning of 2018 (Hafner, 2020; Cavaliere et al., 2020; Chen and Hafner, 2019). We study the network comprising cryptocurrency-specific messages. The mainstream topics related to such a bubble are framing a bubble test and exploiting the proposed test on this unregulated market. To better understand the bubble, we take a different route to monitor changes in network dynamics, which supposes to capture a herding behavior in terms of the messages' similarity.

The paper is organized as follows. Section 2 describes a network, an algorithm for change detection, applies the algorithm to the models and discusses the theoretical properties. Section 3 conducts simulation exercises for the node-initiated and edge-initiated changes, respectively, and robustly checks when the temporal and spatial dependence is present. We present and discuss the application of our model to the social media network in Section 4. Section 5 concludes. We dedicate Section 6 to the proofs. The developed codes for this

study can be found at github.com/wangtengyao/NSM.

2 Online change detection via sequential estimation

2.1 Description of a network

Data on networks becomes increasingly popular in modern statistical applications. A convenient way to a formal definition and visualization of networks offers the graph theory. A graph G is defined to be a pair (V, E), where $\mathsf{V} = (\nu_1, \cdots, \nu_N)$ is a finite set of *vertices* or *nodes*, and $\mathsf{E} = (e_{ij})_{i,j\in\{1,\ldots,N\}}$ is a $N \times N$ *matrix of edge weights*. Within the context of networks the matrix E is frequently referred to as the *adjacency matrix*. In this paper we deal with time-varying networks characterized by *edge-dynamic* graphs, i.e. the graph has a fixed vertex set and its adjacency matrix E_t varies with time $t \in \mathbb{N}$. We write $\mathsf{G}_t = (\mathsf{V}, \mathsf{E}_t)$ with $\mathsf{E}_t = (e_{ij,t})$ for $i, j \in \{1, \ldots, N\}$ and $t \in \mathbb{N}$. If the adjacency matrix is symmetric, i.e. $e_{ij,t} = e_{ji,t}$ for all i, j, we refer to the underlying network as an *undirected network*. Alternatively it is featured as a *directed* one. Moreover, we assume that $e_{ii,t} = 0$ for all i, which implies that there are no loops on the vertices.

The network edges are assumed to be random variables reflecting the stochastic nature of real world networks. Let $e_{ij,t} \sim F$ and $\mathbb{E}(e_{ij,t}) = \mu_{ij,t}$ implying a potentially non-constant expected edge weights. The dynamics of the model is assumed to be affected by a location shift at time point τ, leading to a change-point model formalized as

$$\mu_{ij,t} = \begin{cases} \mu_{ij}^{(1)} & \text{for } t \le \tau \\ \mu_{ij}^{(2)} & \text{for } t \ge \tau + 1 \end{cases}.$$

The objective is to detect this shift in the network as soon as possible after its occurrence. To monitor purposes, we vectorize the adjacency matrices and consider the edge-based monitoring with

$$X_t = \begin{cases} \mathsf{vech}(\mathsf{E}_t) & \text{for undirected networks} \\ (\mathsf{vech}(\mathsf{E}_t), \mathsf{vech}(\mathsf{E}_t')) & \text{for directed networks} \end{cases},$$

where $\mathsf{vech}(\cdot)$ denotes the half-vectorization operator for the lower triangular part of a matrix excluding the main diagonal.

Note that monitoring large networks will suffer from the curse of dimensionality and make most algorithms computationally infeasible. In the economic context, the researchers are inclined to draw a conclusion as to which nodes are responsible for the change. The authorities may particularly keep eyeballing on these detected responsible nodes for a supervisory convenience and effectiveness. To overcome computational problem and benefit surveillance, we consider in the case of undirected graphs the degree of the ith vertex of a graph at t defined as $\mathsf{deg}(\nu_{i,t}) = \sum_{j=1}^{N} e_{ij,t}$, i.e. the sum of edge weights, $e_{ij,t} \in \mathbb{R}$, that are incident to that vertex i. In the case of directed graphs we distinguish between ingoing and outgoing degrees and define $\mathsf{deg}^{(in)}(\nu_{i,t}) = \sum_{j \neq i} e_{ji,t}$ and $\mathsf{deg}^{(out)}(\nu_{i,t}) = \sum_{j \neq i} e_{ij,t}$ respectively. Thus, the in-degree is computed as the number of incoming connections, whereas the out-degree is the number of connections starting at the vertex i. The use of degrees instead of the edges can be seen as a dimension reduction technique signifying the connectedness of nodes in the network. The degree-based monitoring controls the vector of degrees with

$$X_t = \begin{cases} (\mathsf{deg}(\nu_{1,t}), \ldots, \mathsf{deg}(\nu_{N,t}))' & \text{for undirected networks} \\ (\mathsf{deg}^{(in)}(\nu_{1,t}), \ldots, \mathsf{deg}^{(out)}(\nu_{N,t}))' & \text{for directed networks} \end{cases},$$

that quantifies the connectedness of nodes in G_t. We note that if edge weights are independently generated with a light-tail distribution, the vector of degrees only has a weak correlation across coordinates.

2.2 Algorithm for change detection

Using either the edge-based or the degree-based monitoring as described in the previous subsection, we have summarized the dynamic network at each time point as a data vector. This leads us to consider the following mathematical model for change detection, which motivates our NSM algorithm (see Algorithm 1). Let $X_1, X_2, \ldots$ be p-dimensional data vectors sequentially observed with means $\mu_t := \mathbb{E}X_t$, where $\mu_t = \mu^{(1)}$ for $t \leq \tau$ and $\mu_t = \mu^{(2)}$ for $t \geq \tau + 1$. We note that the dimension p of the observations is of order N in the degree-based monitoring and is of order N^2 in the edge-based monitoring. For sparse changes where $\|\mu^{(1)} - \mu^{(2)}\|_0 \ll p$ (we write $\|\mathbf{x}\|_0 := \sum_i \mathbf{1}(x_i \neq 0)$ for the number of nonzero entries in a vector $\mathbf{x}$), Wang and Samworth (2018) proposed a projection-based approach to

retrospectively identify the location of the change point after observing the entire history of data series. A key component of their approach is to estimate the optimal direction of projection, which is parallel to the sparse vector of change. Building upon this idea, we propose in Algorithm 1 to sequentially detect the change by comparing the consecutive estimates of the optimal projection direction. In Step 4 of the algorithm, we compute the CUSUM matrix using all available data sequentially, which is then used in Step 5 to estimate a projection direction $\hat{v}^{(t)}$. Ideally, after a sufficient number of post-change data points are observed, the estimated projection directions at different time points will all be close to the optimal one, and hence are close to each other. On the other hand, prior to the change, there is no signal in the data to favour any particular direction, and thus consecutive estimates are unlikely to align with each other.

In Step 5 of Algorithm 1, the soft-thresholding operator is defined as $\mathbf{soft}(T_{j,t}^{(n)}, \lambda_n) := \mathrm{sgn}(T_{j,t}^{(n)}) \max\{|T_{j,t}^{(n)}| - \lambda_n, 0\}$. This step constructs a projection direction estimator equivalent to the result of the `sparse.svd` function in the `inspect` package (Wang and Samworth, 2016) using argument `schatten=2`. This choice simplifies both the presentation and implementation of the algorithm since the estimator has a closed form solution, and in our experience, it does not significantly impact the empirical performance of the procedure. However, if desired, Step 5 of Algorithm 1 can be replaced with calling `sparse.svd` using argument `schatten=1`. The latter corresponds to a slightly more accurate projection direction estimator based on a nuclear-norm-based (instead of Frobenius norm) convex relaxation, which comes at the expense of a much higher computational cost. We also remark that if $\|T^{(t)}\|_\infty := \max_{j\in[p],s\in[t-1]} |T_{j,s}^{(t)}|$ falls below the soft-threshold level λ_t, Step 7 of Algorithm 1 will generate a projection direction estimator $\hat{v}^{(t)}$ uniformly at random on the unit sphere to reflect our lack of knowledge of the change direction when all CUSUM statistics are too small in absolute values.

After obtaining projection direction estimators at different time points, Step 8 of Algorithm 1 proceeds with computing the sine of the acute angle bounded between the successive estimates $\hat{v}^{(t)}$ and subsequently declaring a change when a sufficient number (provided by the input sequence $(b_t)_{t\in\mathbb{N}}$) of such sine angle distances all fall below the threshold $1/2$. Note that the exact choice of $1/2$ is not an issue. In fact, any number in $(0, 1)$ will have the

same effect theoretically. In the finite sample numerical experiments, the threshold, 1/2, is chosen for the purpose of monitoring convenience and efficiency.

Algorithm 1: **N**etwork **S**equential **M**onitoring (**NSM**) sparse mean change

Input: $X_1, X_2, \ldots \in \mathbb{R}^p$ observed sequentially, threshold sequence $(\lambda_t)_{t\in\mathbb{N}}$, tail length sequence $(b_t)_{t\in\mathbb{N}}$.

Set $t \leftarrow 1$ and draw $\hat{v}^{(1)}$ uniformly from the unit sphere $\mathbb{S}^{p-1}$ (w.r.t. the Haar measure).

repeat

  $t \leftarrow t+1$

  Compute the CUSUM matrix $T^{(t)} = (T^{(t)}_{j,s})_{j\in[p],s\in[t-1]}$ as

$$T^{(t)}_{j,s} \leftarrow \sqrt{\frac{s(t-s)}{t}}\left(\frac{1}{s}\sum_{r=1}^{s} X_{r,j} - \frac{1}{t-s}\sum_{r=s+1}^{t} X_{r,j}\right).$$

  if $\|T^{(t)}\|_\infty > \lambda_n$ **then**

    Define $\hat{v}^{(t)}$ to be the leading left singular vector of **soft**$(T^{(n)}, \lambda_n)$, where **soft** is the entrywise soft-thresholding operator.

  else

    Sample $\hat{v}^{(t)}$ uniformly on the unit sphere $\mathbb{S}^{p-1}$.

  Compute $A^{(t)} \leftarrow \sin\angle(\hat{v}^{(t)}, \hat{v}^{(t-1)})$.

until $t \geq b_t$ *and* $\max_{t-b_t+1\leq i\leq t} A^{(i)} < 1/2$;

Output: $\hat{\tau} = t$

2.3 Theoretical guarantee

The theoretical justification for the NSM algorithm is formalized and articulated in this section. We start by considering a simplified Gaussian model where there is no temporal dependence in the observed data sequence, and later we extend to more general data generating mechanisms in Section 2.4. Specifically, we assume in our theoretical analysis that $X_t = \mu_t + W_t$ for $t = 1, 2, \ldots,$

$$\mu_t = \begin{cases} \mu^{(1)} & \text{if } 1 \leq t \leq \tau \\ \mu^{(2)} & \text{if } t \geq \tau + 1, \end{cases} \tag{1}$$

and that $W_t \sim N_p(0, \Sigma)$ for some covariance matrix

$$\Sigma \in \mathcal{C} := \{A : A \text{ is positive semidefinite with all diagonal entries equal to } 1\}.$$

Moreover, we write $s := \|\mu^{(1)} - \mu^{(2)}\|_0$ for the sparsity of the change and $\eta := \|\mu^{(1)} - \mu^{(2)}\|_2$ for the Euclidean norm of change.

Our first result below shows that with a suitable sequence of soft-thresholding parameters $(\lambda_t)_{t\in\mathbb{N}}$, the NSM algorithm can achieve theoretical control of false alarm rate.

Theorem 1. *Fix some $\alpha \in (0, 1/2]$. Let $\hat{\tau}$ be the output of Algorithm 1 with input $X_1, X_2, \ldots \overset{\text{iid}}{\sim} N_p(\mu, \Sigma)$ for $\Sigma \in \mathcal{C}$, $\lambda_t = 2\sqrt{\log(t^2 p/\alpha)}$ and tail sequence $b_t = \lceil \frac{2\log t + \log(2/\alpha)}{p/8} \rceil$. We have*

$$\mathbb{P}(\hat{\tau} < \infty) \leq \alpha.$$

Here, the choice of the specific form of b_t, which increases logarithmically with respect to t, is to ensure that when no change point is present, the procedure will not produce a false alarm, even for an arbitrarily long sequence of observations. We remark that for most practical applications of moderately large dimension and a not too long monitoring time span, the tail sequence b_t defined in Theorem 1 is often very small. For example, for $p \geq 80$ and $\alpha \geq 0.01$, we have $b_t \leq 2$ for all $t \leq 1500$ and $b_t \leq 3$ for all $t \leq 200,000$, which means that it suffices compare the most recent three or four estimates of projection direction $\hat{v}^{(t)}$ to detect a change. However, practitioners may opt to use a slightly larger value of b_t to be sure that changes have indeed occurred. Of course, this comes at the price of a slightly longer detection delay.

Theorem 1 guarantees that regardless of the location of change, the proposed algorithm is unlikely to declare a change before it happens. The following theorem complements it by showing that our procedure will declare a change not long after the actual change point.

Theorem 2. *Fix $\alpha \in (0, 1/2]$ and $\Sigma \in \mathcal{C}$. Let $z \in \mathbb{N} \cup \{\infty\}$ and let $\hat{\tau}$ be the output of Algorithm 1 with input $X_1, X_2, \ldots X_\tau \overset{\text{iid}}{\sim} N_p(\mu^{(1)}, \Sigma)$, $X_{\tau+1}, X_{\tau+2}, \ldots \overset{\text{iid}}{\sim} N_p(\mu^{(2)}, \Sigma)$, $\lambda_t = 2\sqrt{\log(t^2 p/\alpha)}$ and tail sequence $b_t = \lceil \frac{2\log t + \log(2/\alpha)}{p/8} \rceil$. Assume that*

$$\tau \geq \frac{2^{15/2}\lambda_{2\tau}\sqrt{s\tau}}{\eta} + b_{2\tau}. \tag{2}$$

We have

$$\mathbb{P}\left(\hat{\tau} - \tau \leq \frac{2^{15/2}\lambda_{2\tau}\sqrt{s\tau}}{\eta} + b_{2\tau}\right) \geq 1 - \alpha.$$

From the discussion after Theorem 1, one may regard $b_{2\tau}$ essentially as a constant. Hence, the condition in (2) amounts to $\tau \gtrsim s\eta^{-2}\log(\tau^2\eta/\alpha)$. On the other hand, the response delay is shown to be inversely proportional to the signal strength η. We justify the theoretical choice of threshold in the simulation section below.

2.4 Extensions to more general data generating mechanisms

For the ease of exposition, in Section 2.3 we have focused our theoretical analysis on the setting where $W_t \overset{\text{iid}}{\sim} N_p(0,\Sigma)$ for $\Sigma \in \mathcal{C}$. While Theorems 1 and 2 already allow for arbitrary spatial covariance structure within the class $\mathcal{C}$, these results can be further extended to accommodate temporal correlations and non-Gaussian observations, at the price of a slight modification on the thresholding parameter λ_t. The following result considers temporally dependent component series with sub-Gaussian entries.

Recall that a centered random variable Y is said to be sub-Gaussian if $\|Y\|_{\psi_2} := \inf\{C > 0 : \mathbb{E}(Y^2/C^2) \le 2\}$ is finite. Here, $\|Y\|_{\psi_2}$ is called the sub-Gaussian norm (or the Orlicz 2-norm) of Y.

Theorem 3. *Fix some $\alpha \in (0, 1/2]$. Let $(\xi_t)_{t\in\mathbb{Z}}$ be a sequence of independent random vectors such that $\mathbb{E}(\xi_t) = 0$, $\mathrm{Var}(\xi_t) = \Sigma \in \mathcal{C}$ and $\|\xi_{t,j}\|_{\psi_2} \le C$ for some universal constant $C > 0$ and every $j \in [p]$. For $t = 1, 2, \ldots$, suppose $X_t = \mu_t + W_t$ for $\mu_t = \mu$ and $W_t = \sum_{\ell=0}^{\infty} \beta_\ell \xi_{t-\ell}$, where $\xi_t = (\xi_{t,1}, \ldots, \xi_{t,p})^\top$. If $\left|\sum_{\ell=0}^{\infty} \beta_\ell\right| \le B$, then there exists a universal constant $K > 0$ such that if $\hat{\tau}$ is the output of Algorithm 1 with input $X_1, X_2, \ldots$, $\lambda_t = KBC\sqrt{\log(t^3 p/\alpha)}$ and tail sequence $b_t = \lceil \frac{2\log t + \log(2/\alpha)}{p/8} \rceil$, we have*

$$\mathbb{P}(\hat{\tau} < \infty) \le \alpha.$$

If instead, μ_t satisfies (1) for $\|\mu^{(1)} - \mu^{(2)}\|_2 \ge \eta$ and $\|\mu^{(1)} - \mu^{(2)}\|_0 \le s$, and that the condition specified in (2) holds, then for the same choice of λ_t and b_t, we have

$$\mathbb{P}\left(\hat{\tau} - \tau \le \frac{2^{15/2}\lambda_{2\tau}\sqrt{s\tau}}{\eta} + b_{2\tau}\right) \ge 1 - \alpha.$$

We remark that the temporal and spatial correlation structures of the noise vectors $(W_t)_{t\in\mathbb{N}}$ studied in the above theorem satisfies $\mathrm{Cov}(W_t, W_{t+r}) = \sum_{\ell=0}^{\infty} \beta_\ell \beta_{\ell+r} \Sigma$.

3 Simulation exercise

In this section, we study the numerical performance of our proposed procedure in simulated settings. In all our numerical examples below, we work with the following data generating mechanism. Let $\mathsf{G}_t = (\mathsf{V}, \mathsf{E}_t)$ be an edge-dynamic graph with a deterministic vertex set $\mathsf{V} = \{\nu_1, \ldots, \nu_N\}$ and independent edge weight matrices E_t over time t. For simplicity, we assume all diagonal entries of E_t are equal to 0. We consider both directed and undirected networks. In the former case, $\mathsf{E}_t = (e_{ij,t})_{i,j\in\{1,\ldots,N\}}$ has independent entries and in the latter case, E_t has independent upper-triangular entries. Let $\mu_t = (\mu_{ij,t}) \in [0,1]^{N\times N}$ be some deterministic matrix, we assume that entries of E_t are marginally distributed as

$$e_{ij,t} \sim \text{Beta}\big(10\mu_{ij,t}, 10(1-\mu_{ij,t})\big).$$

Specifically, we have $\mathbb{E}(\mathsf{E}_t) = \mu_t$. In our change point setting, we assume $\mu_t = \mu^{(1)}$ for all $t < \tau$ and $\mu_t = \mu^{(2)}$ for all $t \geq \tau$, where $\tau \in \mathbb{N}$ is the change point of interest.

We will mainly focus on vertex-initiated changes in our simulations. Specifically, let $\mathsf{V}_0 \subseteq \mathsf{V}$ be a subset of vertices of cardinality k, we assume that

$$\mu^{(2)}_{ij} - \mu^{(1)}_{ij} = 0 \quad \forall \nu_i \notin \mathsf{V}_0,\, \nu_j \notin \mathsf{V}_0.$$

In other words, changes only occur in the edges incident on a vertex belonging to the set V_0.

We consider the following three different scenarios of how changes are developed.

(S1) Undirected graph with aligned changes: $\mu^{(2)}_{ij} - \mu^{(1)}_{ij} = c$ for all $\nu_i, \nu_j \in \mathsf{V}_0$ and $i \neq j$.

(S2) Undirected graph with random-signed changes: $\mu^{(2)}_{ij} - \mu^{(1)}_{ij} = \mu^{(2)}_{ji} - \mu^{(1)}_{ji} \sim \text{Unif}\{c, -c\}$ with equal probability independently for all $\nu_i, \nu_j \in \mathsf{V}_0$ and $i < j$.

(S3) Directed graph with random-signed changes: $\mu^{(2)}_{ij} - \mu^{(1)}_{ij} \sim \text{Unif}\{c, -c\}$ with equal probability independently for all $\nu_i, \nu_j \in \mathsf{V}_0$ and $i \neq j$.

As described in Section 2.1, we will apply the NSM algorithm to the simulated datasets generated from three different scenarios (S1), (S2) and (S3) via either an edge-based monitoring scheme (where the data vectors X_t are constructed by vectorising the edge matrix

E_t) or a degree-based monitoring scheme (where the data vectors X_t are constructed as the vector of degrees). We recall that the dimension p of the constructed data vectors is equal to $N(N-1)$ (resp. $N(N-1)/2$) for an edge-based monitering scheme in a directed (resp. undirected) network, and is equal to N for a degree-based monitering scheme. Before discussing our simulation results in detail, we first describe here our tuning parameter choices. Algorithm 1 depends on two sets of tuning parameters $(\lambda_t)_{t\in\mathbb{N}}$ and $(b_t)_{t\in\mathbb{N}}$. As mentioned after Theorem 1, it is typically sufficient in practice to choose the tail length parameter b_t to be a small constant. For simplicity, we will set $b_t = 2$ for all t throughout this section. The choice of the soft-thresholding parameter λ_t is rather subtle. To study the dependence of the algorithmic performance on λ_t, we compared the false alarm rate and average response delays of the procedure over a wide range of settings. Figure 1 shows the simulation results with respect to two performance indicators; results in other models are qualitatively similar. We see that the theoretical choice of $\lambda_t = 2\sqrt{\log(t^2p/\alpha)}$ (corresponding to $r = 2$ in the plot) is rather conservative in terms of the average response delays. Even with a choice of $\lambda_t = \sqrt{2\log(t^2p/\alpha)}$ (corresponding to $r = \sqrt{2}$ in the plot) the false alarm rate is well under control and small response delays can be achieved. Hence, we suggest using $\lambda_t = \sqrt{2\log(t^2p/\alpha)}$ in practice and, unless otherwise specified, we will stick to this choice throughout our simulations.

3.1 Node-initiated changes

The performance of the proposed network sequential detection method hinges on the size of the signal as well as the sparsity of the network. Theorem 1 and 2 characterize how signal strength and soft-thresholding jointly determine the average response delay and the false alarm rate. To gain better insights, we vary both sparsity and signal strength over a grid of possible values. Specifically, we apply the NSM algorithm with a degree-based detection scheme to each of the three scenarios (S1), (S2) and (S3) under $p = 30$, $\tau = 200$, $\alpha = 0.01$, $k \in \{3, 10, 30\}$ and $c \in \{0.05, 0.1, 0.15, 0.2, 0.25\}$. k controls for the sparsity of the network. The parameter c captures the magnitude of the signal and is chosen to range from where the change is just detectable to where the change can be declared almost immediately after it occurs if signals are considerably large. For each setting, we perform 100 Monte

	FA rate	$c = 0.05$	$c = 0.1$	$c = 0.15$	$c = 0.2$	$c = 0.25$
			S1			
$k = 3$	0	10.4	4.84	3.81	3.17	3.02
$k = 10$	0	9.37	4.98	3.91	3.13	3.00
$k = 30$	0	8.70	5.13	3.98	3.13	3.00
$k = 3$	0	189	34.4	17.1	12.5	9.68
$k = 10$	0	162	28.2	15.6	11.3	10.0
$k = 30$	0	151	26.6	14.4	11.4	10.2
			S2			
$k = 3$	0	224	110	32.9	18.8	12.2
$k = 10$	0	150	24.0	12.4	8.01	6.20
$k = 30$	0	69.6	16.9	8.92	6.07	4.94
$k = 3$	0	190	38.2	19.2	12.5	10.0
$k = 10$	0	176	30.5	16.3	12.1	9.93
$k = 30$	0	154	25.9	15.4	11.6	9.68
			S3			
$k = 3$	0	238	150	35.2	18.5	12.6
$k = 10$	0	191	29.3	13.6	8.86	6.83
$k = 30$	0	99.1	19.3	10.2	7.17	5.56
$k = 3$	0	204	41.2	19.9	12.8	10.3
$k = 10$	0	173	32.3	16.9	12.2	10.1
$k = 30$	0	174	28.3	15.6	12.0	10.1

Table 1: Performance of NSM algorithm under sparsity and signal-to-noise ratio
(S1): fixed-sign change in an undirected graph; (S2): random-sign change in an undirected graph; (S3): random-sign change in a directed graph. In each model, the upper part is the results of degree-based detection and the lower part is those from edge-based detection. The first column shows the false alarm (FA) rate, followed by the results of detection delay.

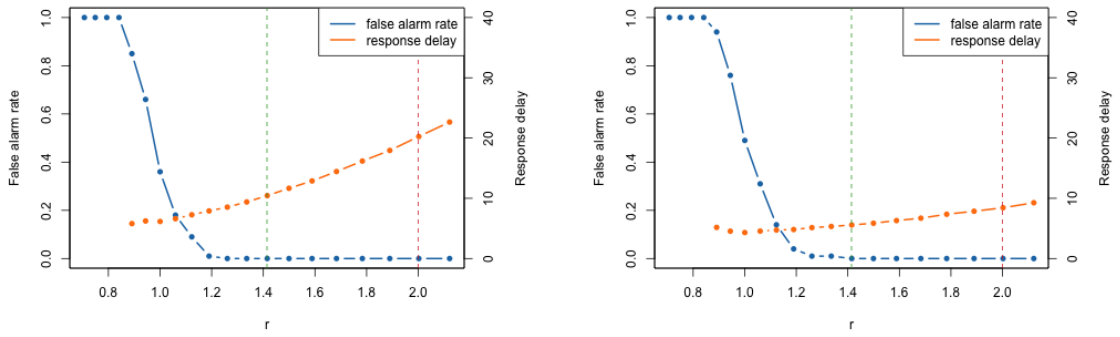


Figure 1: Dependence of false alarm rate and response delay on tuning parameter sequence $(\lambda_t)_{t\in\mathbb{N}}$. Horizontal axis show the scale r for the tuning parameter $\lambda_t = r\sqrt{\log(t^2p/\alpha)}$ and $b_t = 2$. In the left panel, data are generated under (S1) with $N = 30$, $\tau = 200$, $k = 3$, $c = 0.05$ and $\alpha = 0.01$. In the right panel, data are generated under S3 with $N = 30$, $\tau = 200$, $k = 30$, $c = 0.25$ and $\alpha = 0.01$. Both false alarm and response delays are averaged over 100 Monte Carlo simulations. The theoretical and empirical suggestions for the tuning parameter λ are indicated with red ($r = 2$) and green ($r = \sqrt{2}$) vertical dashed lines respectively.

Carlo repetitions and estimate the false alarm rate $\mathbb{P}(\hat{\tau} \leq \tau)$ via the empirical proportion of declared change points that is smaller than or equal to τ. Similarly, we estimate the response delay $\mathbb{E}(\hat{\tau} - \tau \mid \hat{\tau} > \tau)$ by the empirical average of $\hat{\tau} - \tau$ over all repetitions where no false alarm occurs.

In Table 1, we summarize the false alarm rate and the response delay under various sparsity and signal-size settings. The three panels correspond to the three scenarios (S1), (S2) and (S3). For each scenario, in addition to the degree-based detection scheme, we also deploy an edge-based detection scheme as described in Section 2.1 for a comparison purpose.[1] We see that in all simulation settings, no false alarms were observed. In other words, the false alarm rate is well-controlled at the nominal level of $\alpha = 0.01$. The response delay shrinks as either the signal strength c or the sparsity k increases. Note that when we set $b_t = 2$, even if every $\hat{v}^{(t)}$ is estimated to the oracle projection direction, it takes at least three observations after the change point to activate a declaration of a change. Hence, we see that in the setting with the strongest signals, the response delay levels cut off at three.

[1]In Table 1, on a quad-core 4.0 GHz machine, one repetition of each simulation setting in Table 1 takes less than 1 second for the degree-based detection method and less than 30 seconds for the edge-based detection method.

We notice also that the signal size c significantly impacts on the response delay, whereas the sparsity level k is less critical.

Figure 2 illustrates how the NSM algorithm works in our simulated data. It depicts the evolution over time of the sine angle between two consecutive non-zero projection vectors $\hat{v}^{(t)}, \hat{v}^{(t-1)} \in \mathbb{R}^p$, and $\sin \angle(\hat{v}^{(t)}, \hat{v}^{(t-1)})$. In panels (c) and (d), where shift is sizable ($c = 0.2$), the detection works well in the sense that the value of $\sin \angle(\hat{v}^{(t)}, \hat{v}^{(t-1)})$ drops immediately and sharply at the predetermined change point $\tau = 200$ (green line). In the case of (a), the response delay is visible. We conclude that applying the proposed method is sensible as long as signal is considerably substantial.

Figure 3 conveys the similar messages as Figure 2, except for panel (a). Consistent with (S3) in Table 1, in the case, the network is featured with direction and random-sign, under a weak signal and a sparsity the detection blows up as shown in panel (a).

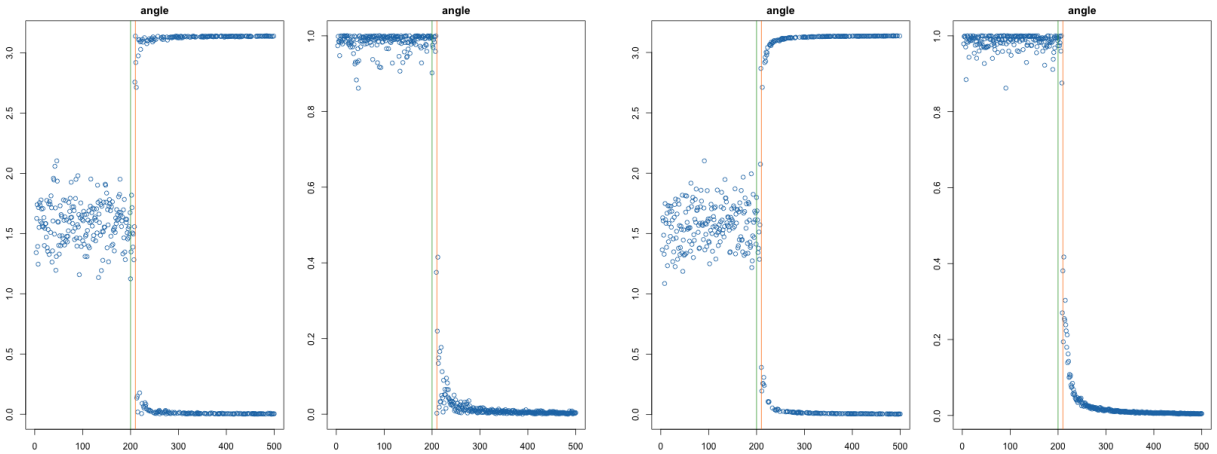
3.2 Edge-initiated changes

The sparsity considered above is restricted to shifts in all edges incident on a subset of vertices of cardinality k. Such shifts induce a well detectable change in the degrees due to aggregation and lead to a superior performance of the degree-based detection method. This is particularly evident in the case of fixed-sign changes (see Panel (S1) of Table 1). Here we relax the above restriction and allow only a few edges of each vertex to shift. Let Ω_0 be the set of vertices in V, such that there is a shift in $\mathbb{E}(e_{ij,t})$ for $\nu_i \in \mathsf{V}_0$ and $\nu_j \in \Omega_0$ at time τ. To formalize this we specify

$$\mu_{ij}^{(2)} - \mu_{ij}^{(1)} = 0 \quad (\forall \nu_i \notin \mathsf{V}_0,\ \forall \nu_j \notin \mathsf{V}_0\ ) \vee (\forall \nu_i \in \mathsf{V}_0,\ \forall \nu_j \notin \Omega_0\ ).$$

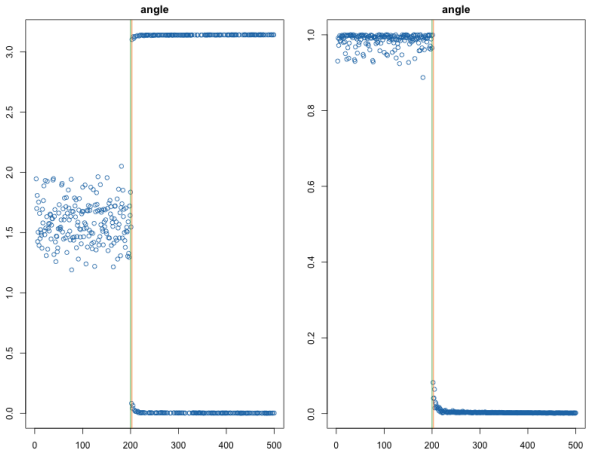
We consider here only undirected graphs with random-signed changes. This implies that the shifts are highly sparse and heterogeneous. The node degrees are expected to react more reluctantly to this type of changes due to offsetting of positive and negative shifts and due to a minor impact of only a few edges on the node degree. The resulting model that modifies (S2) is summarized as follows:

(S4) Undirected graph with random-signed changes and shifts in the first k_e edges: $\mu_{ij}^{(2)} - \mu_{ij}^{(1)} = \mu_{ji}^{(2)} - \mu_{ji}^{(1)} \sim \text{Unif}\{c, -c\}$ with equal probability independently for all $\nu_i \in \mathsf{V}_0$

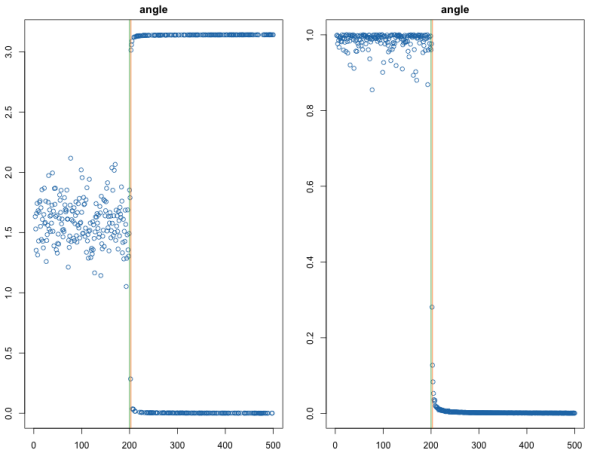


(a) sparse and small change

(b) dense but small change

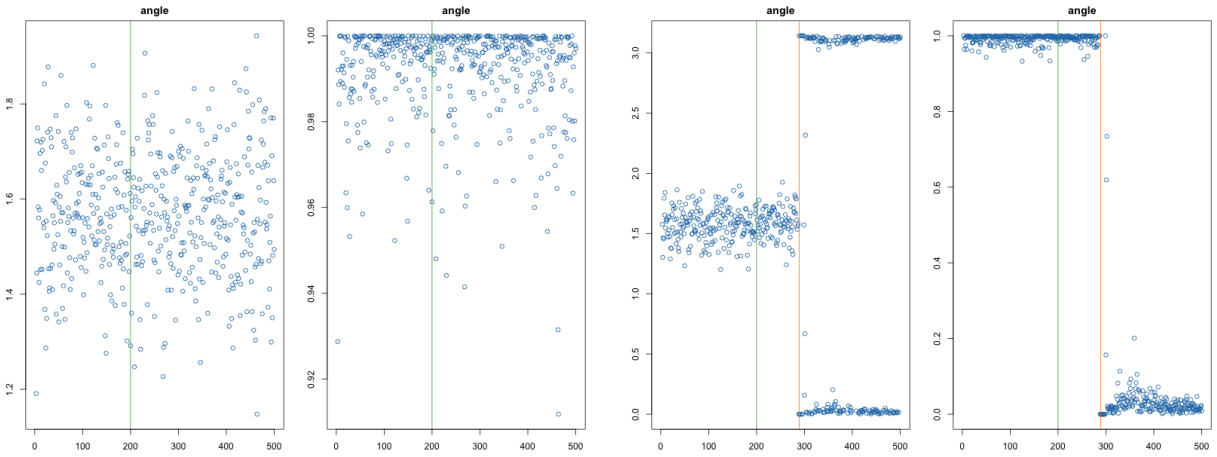


(c) sparse and remarkable change



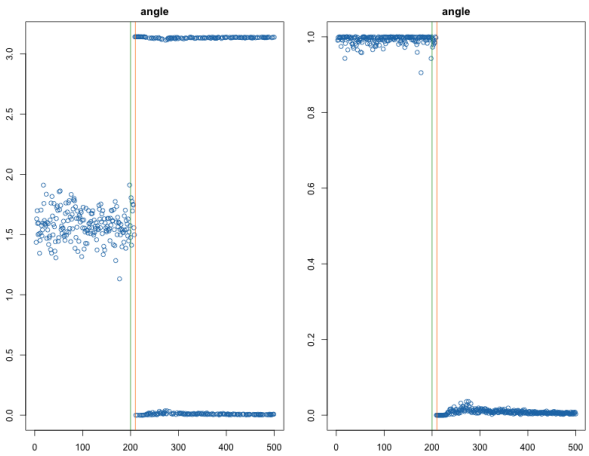
(d) dense and remarkable change

Figure 2: Simulate under scenario (S1) with different sparsity levels and magnitudes of change. Parameter values: (a) $k = 3$, $c = 0.05$; (b) $k = 30$, $c = 0.05$; (c) $k = 3$, $c = 0.25$; (d) $k = 30$, $c = 0.25$.

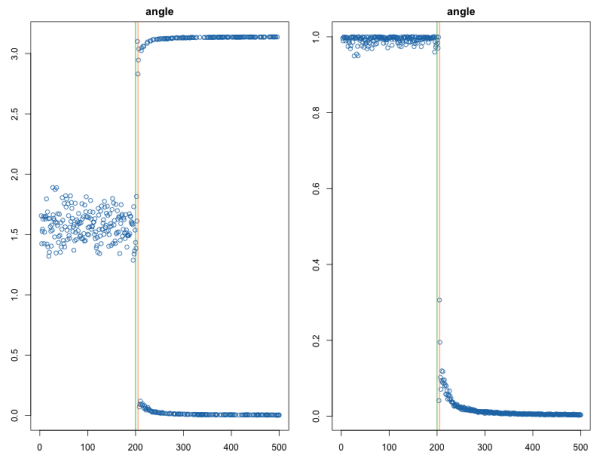


(a) sparse and small change

(b) dense but small change



(c) sparse and remarkable change



(d) dense and remarkable change

Figure 3: Simulate under scenario (S3) with different sparsity levels and magnitudes of change. Parameter values: (a) $k = 3$, $c = 0.05$; (b) $k = 30$, $c = 0.05$; (c) $k = 3$, $c = 0.25$; (d) $k = 30$, $c = 0.25$.

and $\nu_j \in \Omega_0$.

Specifically, we assume that for every vertex in V_0 only the first k_e edges shift. For simplicity, we set $\mathsf{V} = \{1, \ldots, N\}$, $\Omega_0 = \{1, ..., k_e\}$ and apply the NSM algorithm with $k \in \{2, 3, 5\}$ and $k_e \in \{10, 15, 20, 25\}$. The remaining parameters are set as in the case of node-initiated changes. Figure 4 shows the average delay for edge-initiated shifts as described in Scenario (S4). If the shift is driven only by 2 or 3 nodes, the edge-based method enjoys a shorter delay. However, such advantage is gradually dampening as more nodes contribute to the change. We will discuss the comparison between the degree-based and the edge-based detection in more details in Section 3.4.

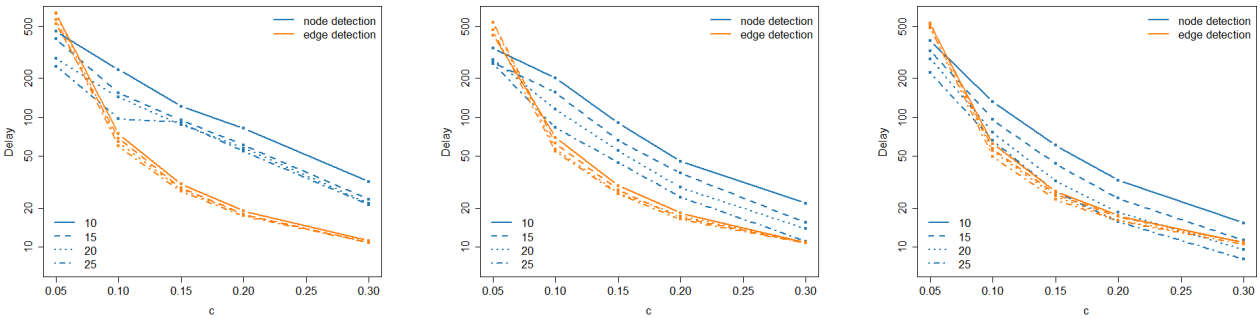


Figure 4: Detection delay initiated by random direction edge-initiated shifts of size c in $k_e = 10$, 15, 20, 25 (out of 29) edges of the first $k = 2$, 3 or 5 nodes (left to right) as in (S4) with node and edge monitoring.

3.3 Spatial and temporal dependent analysis

The above simulation study assumes independent edge weights and degrees both temporally and spatially. The theoretical justifications for the suggested monitoring technique allow us to extend the NSM algorithm to time-dependent and cross-correlated data. This section assesses the performance of the algorithm in this more general setting. We restrict the discussion to node-initiated shifts and degree-detection for undirected networks only. Very similar simulation results were also obtained for directed networks and edge-initiated changes. As above we monitor the vector of degrees

$$X_t = (\mathsf{deg}(\nu_{1,t}), \ldots, \mathsf{deg}(\nu_{p,t}))'.$$

For the ease of constructing spatially and temporally correlated data series, we will change to a Gaussian design in this subsection. More specifically, for a cross-sectional dependence we impose a spatial correlation structure via

$$X_t \sim N_p(\mu_t, \Sigma),$$

where $\Sigma = (\sigma_{ij})_{i,j=1,\dots,p}$ with $\sigma_{ij} = \rho^{|i-j|}$ and $\sigma_{ii} = 1$ for all i. Since the node degrees are sums of the incident edge weights, we justify the use of normal approximation by the CLT. For consistency with the above results we assume that only three vertices are affected by shifts, i.e. $k = 3$. The results of the simulation study are visualized in the left panel Figure 5. Since the spatial correlation is typically characterized by positive dependence, we restrict ρ to the unit interval. The figure reveals that the delay only slowly increases as a function of ρ, whereas this increase becomes more evident in absolute terms for small shift sizes and extreme correlations.

To render temporal dependence in the node degrees we impose an autoregressive dynamics on X_t in the form

$$X_t = \mu_t + \varepsilon_t,$$

with $\varepsilon_t = (\varepsilon_{1t}, ..., \varepsilon_{pt})'$ such that

$$\varepsilon_{it} = \gamma\varepsilon_{i,t-1} + \sqrt{1-\gamma^2}u_{it}, \qquad \text{and} \quad u_{it} \sim N(0,1). \tag{3}$$

Scaling the residual u_{it} guarantees constant variance of the node degrees and comparability of the results. Motivated by our theoretical result in Theorem 3, we suggest that the tuning parameter λ_t should be scaled up to $\lambda_t = \frac{1}{1-\gamma}\sqrt{2\log(t^2p/\alpha)}$ to account for the temporal correlation, since otherwise, the procedure will raise false alarms too often. The right panel of Figure 5 shows a moderate increase in detection delay as the autoregressive parameter α increases from -0.3 to 0.3. Such an increase can be intuitively explained by the decrease in the effective sample size as the successive observations become increasingly positively correlated.

3.4 Comparison between degree-based and edge-based detection

There exists a striking difference between the degree-based and edge-based detection scheme in terms of performance. By comparison, the degree-based method performs much better

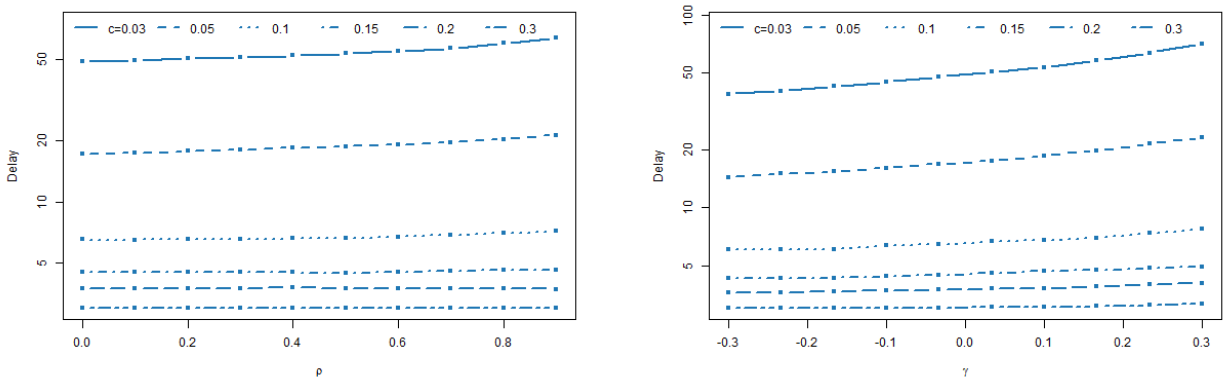


Figure 5: Detection delay of shifts in the first three nodes assuming spatial (left) and temporal (right) dependence of the node degrees. The false alarm rates in both settings were below 0.5%.

in (S1) and (S2), which corresponds to the scenarios where all edges weights change in the same direction in mean. The observed superiority of the degree-based scheme is attributed to the fact that the degree-based aggregation exploits the 'grouping structure' of $\mu^{(2)} - \mu^{(1)}$ in the node-initiated change settings, in the sense that the edges incident on the same vertex are likely to change coincidentally in the same direction. On the other hand, the comparison between the degree-based and the edge-based detection schemes yields a mixed result in (S3) of Table 1. Since the change is featured with random signs, degree aggregation may stumble caused by an offset effect. While aggregating signs, the positive sign cancels out the negative one, and vice versa. This offset effect exacerbates in the sparse change case when $k = 3$. For denser changes ($k = 10$ or $k = 30$), however, the advantage, e.g. the hidden grouping structure outweighs the cost imposed by an offset effect, and the degree-based method becomes viable again.

The results of simulation studies that are summarized in Figures 4 and 6 confirm our intuition. As has been observed in Section 3.2, Figure 4 shows that the edge-based detection method enjoys an advantage when the change is driven by a small number of nodes. Larger sizes of the shift reveal the superiority of the edge-detection technique. This becomes particularly evident if the number of affected nodes is small and the aggregated shift in the node degree is difficult to identify. Moreover, the edge-detection is extremely robust to the variation in the number of affected edges k_e. On the contrary, the variation in the delays

for degree detection as a function of k_e is very substantial, mirroring the aggregation effect in the degrees. The dominance of the edge detection diminishes, however, if we weaken the sparsity of the changes by increasing the number of vertices with shifts. For example, if most of the edges of the five nodes are subject to a change, then both approaches show almost equivalent performance. The extreme case of node-initiated shifts with all edges of the selected vertices being shifted is shown in Figure 6. We confirm the finding that if many vertices are affected by the change, then the node-detection shows superior performance, whereas the edge detection dominates for a few shifted nodes. The size of the shift has a similar impact as in the case of the edge-initiated changes.

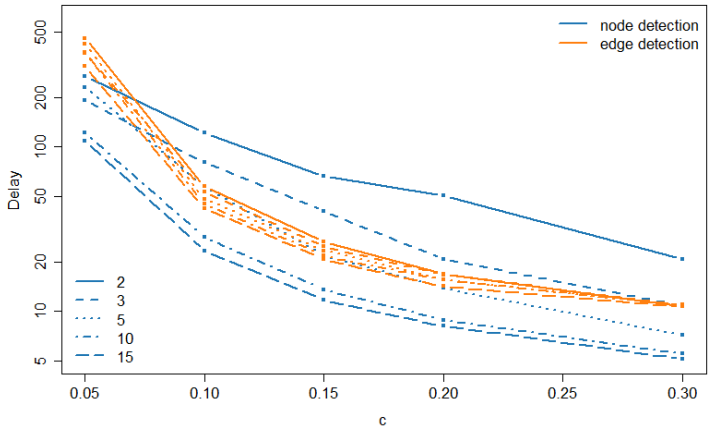


Figure 6: Detection delay initiated by random direction node-initiated shifts of size c in all edges of the first 2, 3, 5, 10 and 15 vertices with node and edge monitoring.

4 An application to social media networks

4.1 Social media data

The social media text are retrieved from StockTwits. It is similar to Twitter but dedicates to the financial discussion. Individuals, investors, market professionals can publish 140-character messages to 'Tap into the Pulse of the Market and financial assets'. With an emphasis on financial discussions, it gains popularity from investors and traders. We limit our attention to cryptocurrency for the presence of 'bubble' during the end of 2017 and the

beginning of 2018, known as a speculative bubble. The formation and burst of a bubble have been studied in Hafner (2020); Cavaliere et al. (2020); Chen and Hafner (2019), and more are in production.

Since 2014 StockTwits adds streams and symbology for cryptocurrencies and tokens, expanding from 100 cryptos to more than 400 cryptos recently. New cryptocurrencies are regularly added to the list of cashtags supported by StockTwits.[2] A cashtag refers to a cryptocurrency if and only if it ends with ".X" (e.g. $BTC.X for Bitcoin, $LTC.X for Litecoin). We use this convention and StockTwits Application Programming Interface (API) to download all messages containing a cashtag referring to a cryptocurrency. StockTwits API also provides for each message its user's unique identifier, the time it was posted at with a one-second precision, and the sentiment associated by the user ("Bullish", "Bearish" or unclassified).

Investors' interest in cryptos, revealed by message volumes, is clearly skewed toward to top 30 cryptos as shown in Figure 7. These 30 leading cryptos have attracted 1,575,205 messages in total, amount to 48% of total retrieved messages across 425 cryptos in a time span 23.07.2017–31.12.2018. Among these 30 coins, Bitcoin stands out the most popular coin from other alternative coins and has attracted around 1,141 messages per day. We decide to emphasize on these 30 coins in a consideration of message volumes. The retrieved messages are unstructured and need to be pre-processed by using the Python NLTK toolkit that offers the equipped modules for the text normalization, including word-level tokenization, Lemmatization, part-of-speech tagging and stop words removal.

Social media is an ideal venue to build up social networks comprising of users or symbols as nodes. Once the vertex is chosen by the research of interest as "symbol" rather than "user", a direct network is no longer observable. To cope with it, one may try to infer the hidden edges in the network by measuring the similarity between two symbols, in terms of user characteristics and their mentioning information. In other words, the network to be formed makes use of social media message information as to the extent to which the expressed message tone/sentiment of symbol i resembles that of symbol j. We then infer nodes i, j are connected in terms of a shared opinion between them.

[2]This list can be found at https://api.stocktwits.com/symbol-sync/symbols.csv

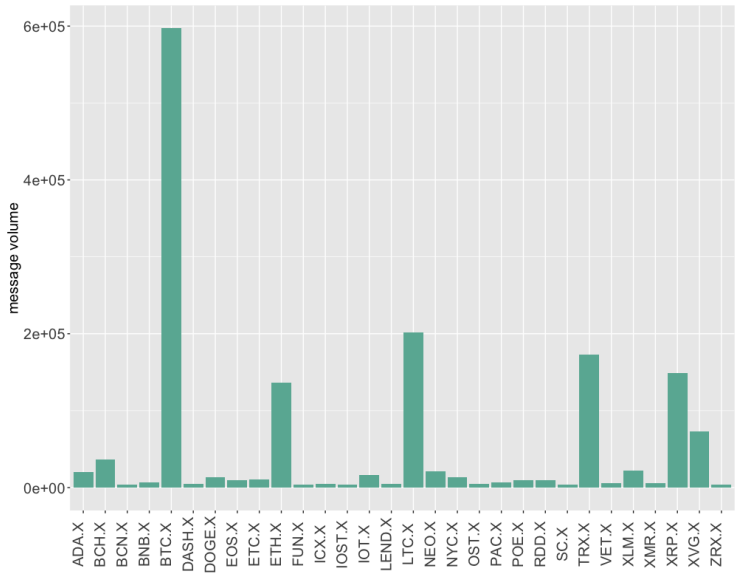


Figure 7: Message volume across top 30 cryptos

The network made by top 30 cryptos at each time point is constructed by the messages' similarity in an aggregation level at t. A collection of messages for coin i at any time point is converted to a vector of word counts $\mathbf{x}_i = (x_{i,1}, \cdots, x_{i,V})$, where $x_{i,w}$ is the number of appearance of the $w-$th word in i, and V is the vocabulary size in the corpus. The text similarity between coin i and j is measured by `cosine similarity` as follows:

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \frac{\phi(\mathbf{x}_i)'\phi(\mathbf{x}_j)}{\|\phi(\mathbf{x}_i)\|_2 \|\phi(\mathbf{x}_j)\|_2} \tag{4}$$

where $\phi(\mathbf{x}) = \text{tf-idf}(\mathbf{x}_i) \equiv \left[\text{tf}(x_{i,w}) \times \text{idf}(w)\right]_{w=1}^{V}$, $\text{tf}(x_{i,w}) \equiv \log(1 + x_{i,w})$, $\text{idf}(w) \equiv \log \frac{M}{1+\sum_{i=1}^{M} \mathbf{1}(x_{i,w}>0)}$ with M, the total number of messages at the particular time point. The value of $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ is in the range of zero (documents are orthogonal) and one (documents are highly similar).

4.2 Monitoring changes in cryptocurrency network dynamics

We apply the sequential network change detection method along with the NSM algorithm to the prescribed social media network that evolves over time. The network of which the edge between a pair of nodes is their messages' similarity features a undirected and weighted type of network. Messages arrive sequentially and text of messages vary across cryptocurrencies and time, constituting time-varying weighted edges and eventually a dynamic network.

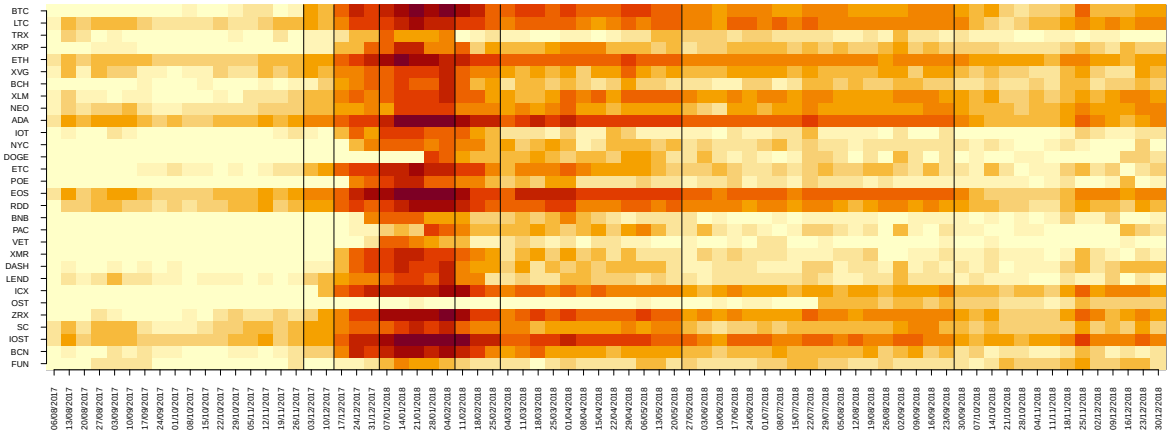


Figure 8: Heatmap of the degree of vertex in the cryptocurrency message network

The coins are in order (top-down) by its message volume in Figure 7. The solid lines are the estimated change point locations through a data-driven offline change point test, see Wang and Samworth (2018).

Starting from 23.07.2017, we sequentially, upon newly arrived messages, monitor a change of network dynamics until 31.12.2018. As such, given a weekly frequency, observation n can grow up to 78 during an investigation period. Figure 8 displays the heatmap of the nodal degree across the selected 30 coins. Clearly, by the color distribution of the heatmap we observe a comovemnt of the dynamics of the nodal degree, especially during the bubble. Some leading coins e.g. BTC or LTC persistently exhibit a high value of nodal degree starting from end of 2017, showing a strong connection with other nodes.

Both the degree-based and the edge-based detection are implemented. To manifest the online monitoring feature of detection, the NSM algorithm is reset whenever the change is detected and iterates until the next detected change. To account for the temporal correlation, we apply the refined threshold $\lambda_t = \frac{1}{1-\gamma}\sqrt{2\log(t^2 p/\alpha)}$ to the NSM algorithm. γ, specified in (3), is estimated for each individual coin, and further are averaged out across coins to get a cross-section average. In Figure 9 we depict the resulting CUSUM and the sine angle between two non-zero projection vectors to address the detected change points. For the degree-based one (Figure 9a), we conclude the two change points; one is in 2018-01-07 and the other one is declared in 2018-06-03. The first change point coincides with the detected bubble periods in Chen and Hafner (2019). We associate the change points with

the price dynamics of Bitcoin with a price soar since mid of December 2017 (see Figure 9c). Thereby, the first change point captures this soar with a delay in a range of 2-3 week, which is not a surprise due to a choice of $b_t = 2$. As noted, we set $b_t = 2$ to imply at least three observations after the change point to activate a declaration of a change. In this regard, the declared change is a compromise between the false alarm and the efficiency of detection. The second change point, however, captures a shrink of the bubble. The Bitcoin price at that time point was reverted back to the pre-bubble level.

We turn to the edge-based detection in Figure 9b. It appears that instead of two changes detected in the degree-based method, there are five changes being declared, and only one change point – 2018-01-07 – coincides. The coincidence happened to 2018-01-07 may imply that such change is widespread throughout the network, thereby both methods flag a change. The first change point in 2017-10-29 is detected to signal an incoming price soar, whereas the third point in 2018-02-11 corresponds to a big price swing. The price of Bitcoin fell by about 65 percent during the month from 6 January to 6 February 2018. The trajectory of the CUSUM during 2018-01-07 and 2018-02-11 exhibits a sharp exponential growth along with a drop in the value of sine angle, compared to other segmented periods divided by the change points. The last two change points in 2018-09-09 and in 2018-11-18 signal a vanishing bubble in two phases. Figure 10 displays the snapshots of the network corresponding to the declared change points. From the first change, 2017-10-29, one observes that a set of 22 coins form a connected network, whereas the other 8 coins are disconnected. As to 2018-01-07, except for coin 13, DOGE (Dogecoin), [3] all coins manifest a strong similarity of messages to imply an adjacency. During the burst of the bubble, from 2018-02-11 to 2018-09-09, the connectivity of networks remains, although one finds the former one looks denser than the latter one. The snapshot of 2018-06-03, albeit not a change point, it helps visualize a transition between two networks. As of 2018-11-18, the 12th coin (NYC, an open-source P2P cryptocurrency) is completely disconnected. The 13th coin (Dogecoin) and 19th coin (PAC, a coin for retail adoption between consumers and merchants) turn less connected. These small and also low-value coins have prices below $ 0.001. It appears, during the post-bubble period, these small coins start to disconnect

[3]The coin that is created to for a purpose of making fun of the wild speculation in cryptocurrencies at the time. As shown in Figure 8, it starts connecting with other coins from 2018-01-28

from the bigger ones.

We discuss the extracted insights after evaluating these two approaches. First, the degree-based detection is likely to respond more slowly than the edge-based one as it requires to average out all edges in the network. In the case whereby there is an abrupt change followed by another change, the subsequent change may incur a longer delay, which is attributed to the required number of observations after the preceding change was detected. To leverage a minimum required observation used to declare a change, researchers often impose a restriction on the minimum size of signal in the online version of change tests. By comparison, the edge-based detection is less restricted by a minimum requirement of signal, which is attributed to the fact that the edge-based method benefits from more observations. Second, the effectiveness of the degree-based method might be subject to the structure of a change. It may work effectively if the changes across the edges are rather homogeneous (same direction and magnitude). In the case where the changes are pervasive in the network to signal an outbreak of crisis or an emerging bubble, the degree-based method features an efficient and effective technique. However, in reality, the changes can be very heterogeneous and might be smoothed out by aggregating these divergent changes.

As to the edge-based method, it benefits from less restriction on the size of signal and also observation, experiences a shorter delay, and is effective under a variety of change structures. A possible downside of employing an edge-based method is that nuance might be overstated as a change, causing a "false alarm". An offline investigation from Figure 8 can be seen as a validation reference. There are six breakpoints, of which "2018-01-07" coincides with the online detection, supporting the proposed method. The detected "2018-02-11" further supports the online fashion of the edge-based method. From the detection results, we conclude that the edge-based one responds more promptly than the offline method, whereas the degree-based method reacts slowly until the changes are widespread throughout. Likewise, using a piece of offline information, we judge a possibility of a false alarm. In a nutshell, to which degree the change should be paid attention is subject to the researcher's discretion and their research questions.

An inspection of the leading left singular vectors via Figure 11a brings additional insights as to the extent to which nodes are responsible for a change. Before the change

being declared, the coordinates (nodes) evenly appear in the projection vectors, implying their homogeneous representativity. Whenever there is a change, the projection vectors are led by a few nodes. We observe that some nodes dominate over other nodes in terms of governing the projection direction. For instance, at the first change point, BTC (Bitcoin, the most bottom one), LTC (Litecoin), ETH (Ethereum) are the main supports. These cryptocurrencies are characterized by a long lifetime, higher adoption, high market capitalization, and high market attention proxied by message volume in Figure 7. These identified nodes reconcile with the leading cryptos classified by market participants. By comparison, the constellation of projection vectors in the case of an edge-initiated change in Figure 11b looks similar. After scrutiny, those edges driving the change are nearly centered around the identified leading nodes, although an additional clustering technique can be applied in the future study. In terms of sparsity, only 14% of edges drive the change, which seems quantitatively comparable to the node-initiated one, that is, 16% nodes are in charge of the change.

We show that, for a purpose of surveillance in the decentralized markets, these cryptos driving a change in the network require close monitoring. Another implication is that the opinions on the digital currency extracted from the crowd can be approximately represented by these leading coins to cope with a challenge of the opinion extraction in a high dimension network.

5 Conclusion

In this paper, we consider the problem of surveillance and detection of structural breaks in dynamic networks. In contrary to the modeling-related issues and ex-post change-point tests, which are widely discussed in the literature, we focus on sequential monitoring and aim to detect the change as soon as possible after its occurrence. Due to a specific structure of a network, we cannot apply the techniques commonly used for monitoring multivariate data. To overcome this problem we deploy the idea of optimal projection direction, which assumes a sparse change in a small subset of nodes or edges which is consistent with the empirical evidence from economic and financial networks. In practice, we can frequently hypothesize that a change is driven either by a few nodes of the network or by specific edges

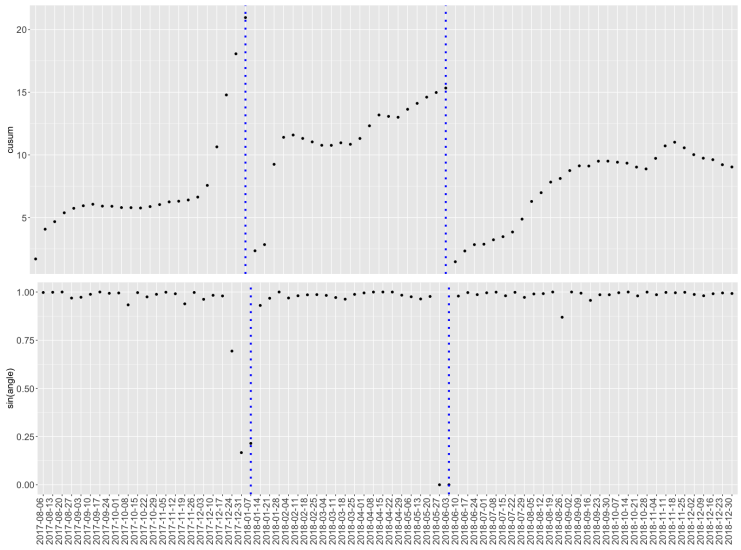
leading to a node- or edge-initiated shift. To reflect this insight we suggest degree- and edge-based monitoring schemes, where the objects of interest are the vectorized adjacency matrix or the vector of node degrees respectively. We recommend using the degree-based detection if we believe that all edges incident to a few nodes are unanimously affected by a shift or if the direction of the shift is not random, implying links getting either only weaker or only stronger. It is important to stress, that the NSM algorithm is capable of detecting both abrupt and smooth shifts in the network characteristics and thus can be used under very heterogeneous economic conditions. Additionally to the practical relevance of the problem, the suggested approach relies on a solid theoretical background. We show that with a suitable choice of parameters we attain the prespecified false alarm rate and declare a signal with high probability within a short time after an actual shift.

In an extensive simulation study, we confirm several important features of the algorithm. First, the detection delays are particularly short for fixed-sign changes and even for small sizes of the shifts. Second, the algorithm is computationally efficient and requires a few minutes even in the case of edge-based detection for networks of moderate size. Third, the potential spatial dependence between the nodes and/or edges has only a very minor impact on the detection delay and can be mostly neglected in applications. The impact of time-dependent network characteristics is slightly stronger and might lead to more frequent false alarms. However, this can be alleviated by appropriately setting the detection threshold based on an estimated strength of the temporal dependence. Monitoring the social media data in the empirical illustration reveals the relevance of the developed methodology for practical applications. Particularly, the core idea of optimal projection direction allows for deep insights into the structure of shifts and for a more precise economic interpretation of the results.

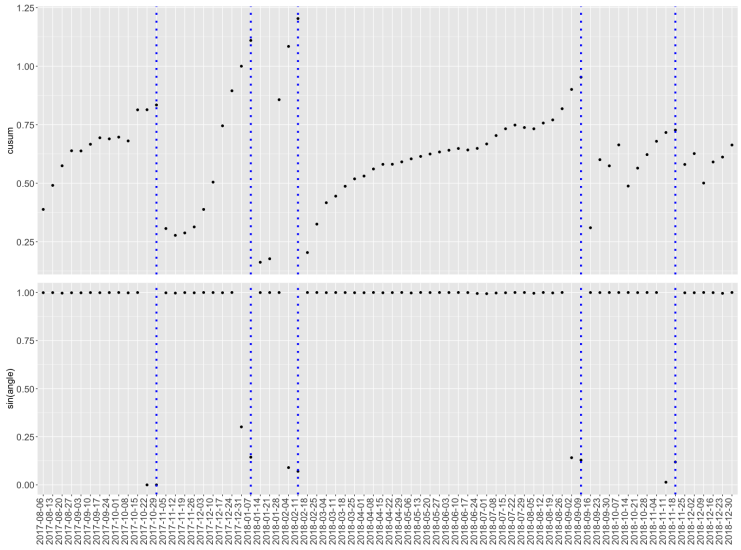
6 Proof of theoretical results

Proof of Theorem 1. For every $t \geq 1$, define the noise CUSUM matrix $E^{(t)} = (E^{(t)}_{j,s})_{j\in[p],s\in[t-1]}$ as

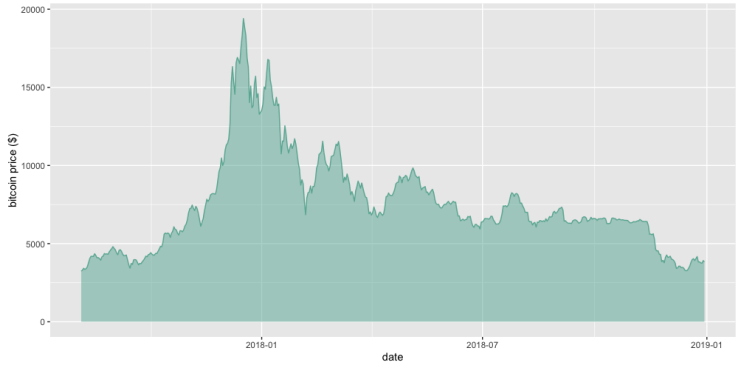
$$E^{(t)}_{j,s} := \sqrt{\frac{s(t-s)}{t}}\left(\frac{1}{s}\sum_{r=1}^{s} X_{r,j} - \frac{1}{t-s}\sum_{r=s+1}^{t} X_{r,j}\right).$$



(a) Degree-based detection



(b) Edge-based detection



(c) Bitcoin price dynamics

Figure 9: Monitoring changes in the dynamics of cryptocurrency networks

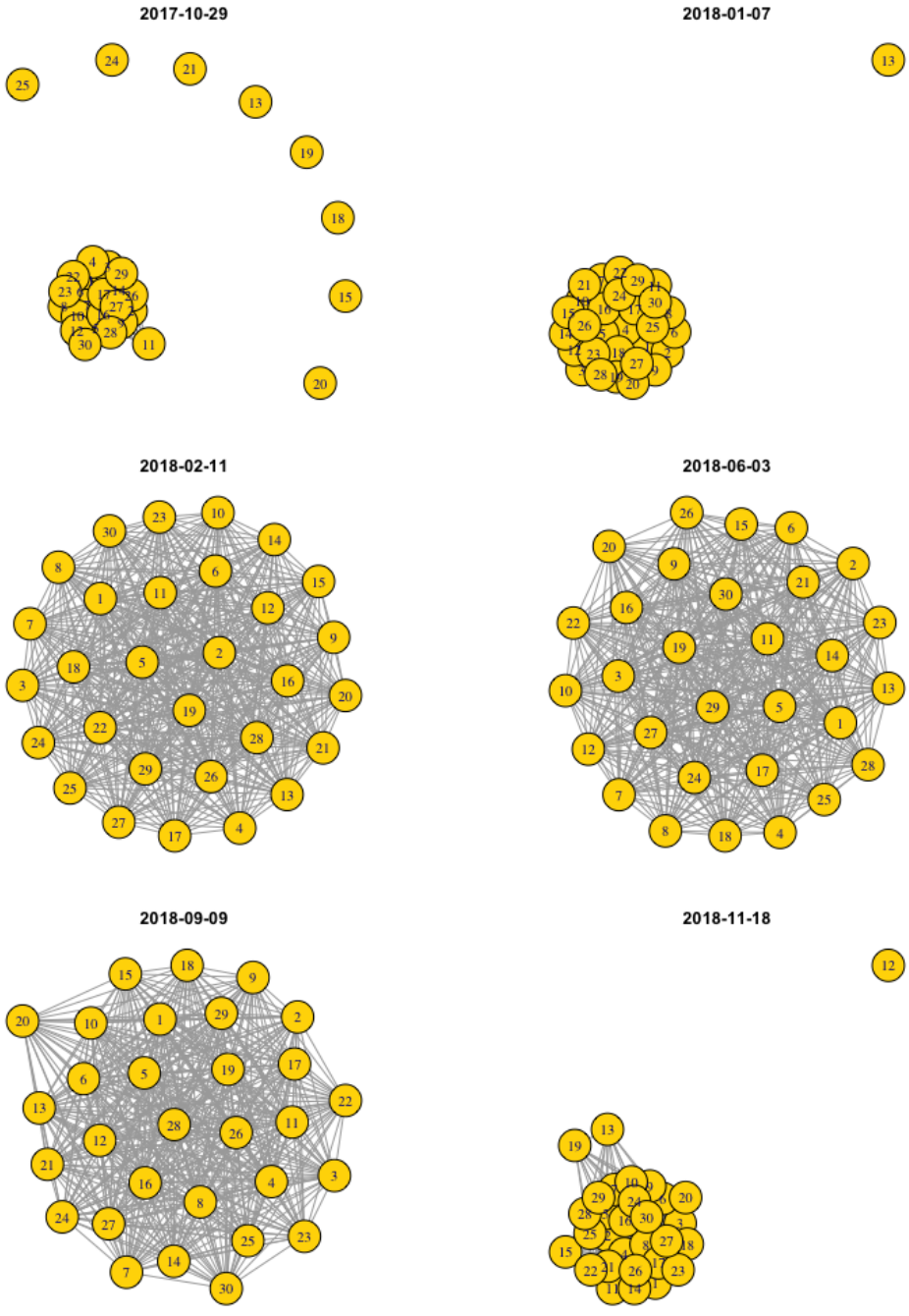
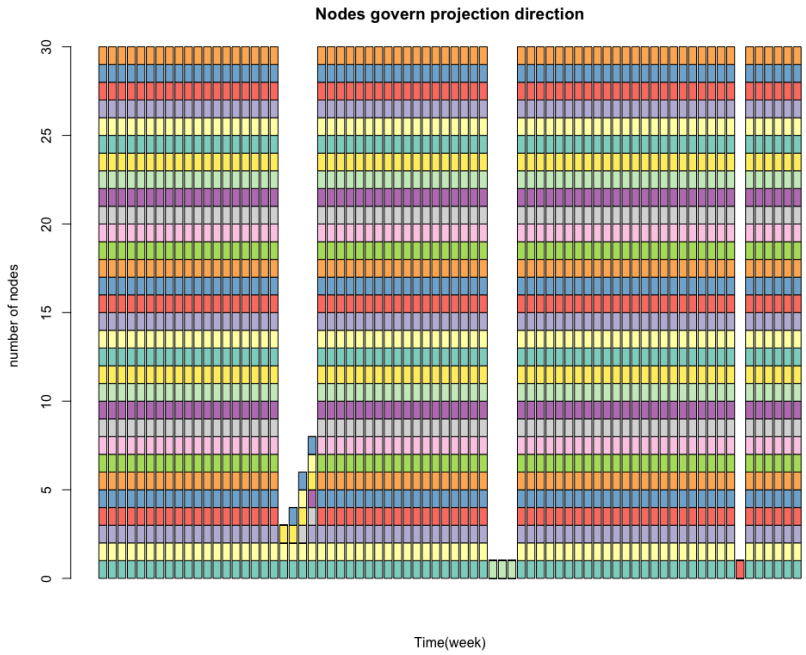
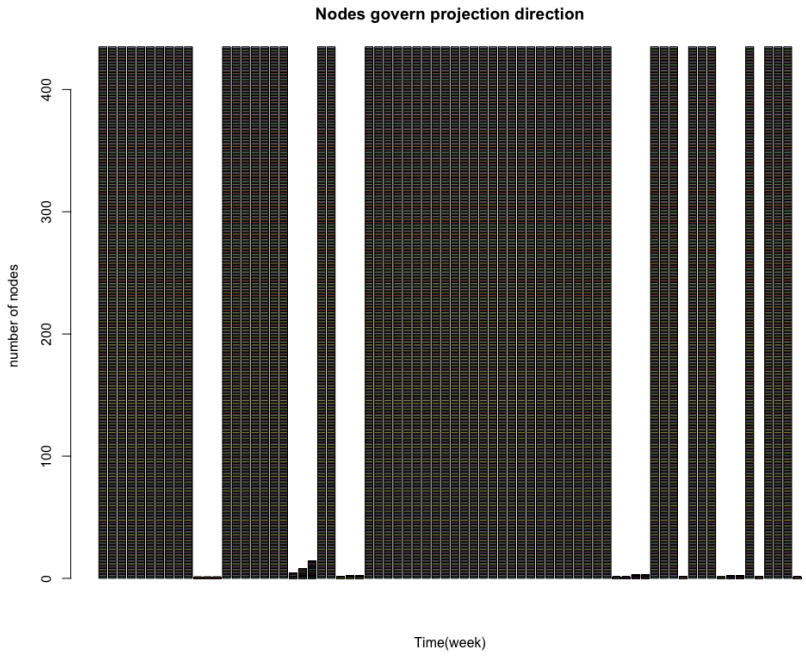


Figure 10: Snapshots of network dynamics



(a) Degree-based detection



(b) Edge-based detection

Figure 11: Dynamics of projection vectors
The bars represent the coordinates which are non-zero valued in the projection vectors. The emergence of coordinates change over time. The green line indicates the location of change.

We consider the following event

$$\Omega_1 := \bigcap_{t=2}^{\infty} \{\|E^{(t)}\|_\infty \le \lambda_t\}. \tag{5}$$

Note that under the assumption of this theorem, we have $T^{(t)} = E^{(t)}$. By applying Lemma 4 of Wang and Samworth (2018) to each component series of $E^{(t)}$ and using a union bound, we have for each $t \ge 2$ and $\lambda > 0$ that

$$\mathbb{P}(\|E^{(t)}\|_\infty > \lambda) \le \sqrt{\frac{2}{\pi}} p \lceil \log t \rceil (\lambda + 2/\lambda) e^{-\lambda^2/2}.$$

Since $\lambda \mapsto (\lambda + 2/\lambda)e^{-\lambda^2/4}$ is decreasing for $\lambda \in (0, \infty)$ and $\lambda_t = 2\sqrt{\log(t^2 p/\alpha)} \ge 2\sqrt{\log 8}$ for $t \ge 2$ and $\alpha \le 1/2$, we have $(\lambda_t + 2/\lambda_t)e^{-\lambda_t^2/4} \le 0.45$. Therefore,

$$\mathbb{P}(\|E^{(t)}\|_\infty > \lambda_t) \le \sqrt{\frac{2}{\pi}} p \lceil \log t \rceil 0.45 e^{-\lambda_t^2/4} \le \frac{0.36\alpha \lceil \log t \rceil}{t^2}.$$

Taking a union bound of the above inequality for all $t \ge 2$, we obtain that

$$\mathbb{P}(\Omega_1^{\mathrm{c}}) \le \sum_{t=2}^{\infty} \frac{0.36\alpha \lceil \log t \rceil}{t^2} \le 0.451\alpha.$$

Working on the event Ω_1, we have $\hat{v}^{(t)} \overset{\text{iid}}{\sim} \mathrm{Unif}(\mathbb{S}^{p-1})$ for all $1 \le t \le \tau$. By rotational symmetry, we have $A^{(t)} \overset{\mathrm{d}}{=} \sin \angle(\hat{v}^{(t)}, e_1)$ for $1 \le t \le \tau$ where e_1 is the first standard basis vector in $\mathbb{R}^p$. In particular, $A^{(t)}$ is independent of $v^{(t-1)}$ and consequently $A^{(1)}, \ldots, A^{(\tau)}$ are independent and identically distributed. Let $Z \sim (0, I_p)$. We see that

$$\langle v^{(t)}, e_1 \rangle^2 \overset{\mathrm{d}}{=} \frac{Z_1^2}{Z_1^2 + \cdots + Z_p^2} \sim \mathrm{Beta}\left(\frac{1}{2}, \frac{p-1}{2}\right).$$

Using a Beta distribution tail bound (see, e.g. Marchal et al., 2017, Theorem 2.1), we have

$$\mathbb{P}(A^{(t)} < 1/2 \mid \Omega_1) \le \mathbb{P}\left\{\mathrm{Beta}\left(\frac{1}{2}, \frac{p-1}{2}\right) \ge \frac{3}{4}\right\} \le e^{-2(p/2+1)(3/4-1/p)^2} \le e^{-p/8}.$$

Consequently, we have

$$\begin{aligned}
\mathbb{P}(\hat{\tau} \le \tau) &\le \mathbb{P}(\Omega_1^{\mathrm{c}}) + \sum_{t=\max\{2, b_t\}}^{\infty} \prod_{i=t-b_t+1}^{t} \mathbb{P}(A^{(i)} < 1/2 \mid \Omega_1) \\
&\le 0.451\alpha + \sum_{t=2}^{\infty} e^{-pb_t/8} \le 0.451\alpha + \sum_{t=2}^{\infty} \frac{\alpha}{2t^2} \le \alpha,
\end{aligned} \tag{6}$$

as desired. □

Proof of Theorem 2. We work on the event Ω_1 defined in (5), which satisfies $\mathbb{P}(\Omega_1^c) \leq \alpha$ as shown in the proof of Theorem 1. We will show that the desired response delay bound holds on the event Ω_1.

Writing $v := \theta/\|\theta\|_2$, by the proof of Proposition 1 of Wang and Samworth (2018), we have on Ω_1 and for all $t > \tau$ that

$$\sin \angle(\hat{v}^{(t)}, v) \leq \frac{32\lambda_t\sqrt{st}}{\min\{\tau, t-\tau\}\eta}.$$

In particular, for all t satisfying

$$\tau + \frac{2^{15/2}\lambda_t\sqrt{s\tau}}{\eta} \leq t \leq 2\tau, \tag{7}$$

we have

$$\sin \angle(\hat{v}^{(t)}, v) \leq \frac{32\lambda_t\sqrt{2s\tau}}{(t-\tau)\eta} \leq \frac{1}{4}.$$

By triangle inequality, we then have

$$A^{(t)} = \sin \angle(\hat{v}^{(t)}, \hat{v}^{(t-1)}) \leq \sin \angle(\hat{v}^{(t)}, v) + \sin \angle(v, \hat{v}^{(t-1)}) \leq 1/2,$$

for all but the first t satisfying (7).The condition in (2) ensures that the set of t satisfying (7) has cardinality at least $b_{2\tau} + 1$. Consequently, we have on event Ω_1 that

$$\hat{\tau} \leq \tau + \frac{2^{15/2}\lambda_{2\tau}\sqrt{s\tau}}{\eta} + b_{2\tau},$$

as desired. □

The proof of Theorem 3 requires the following lemma. We first establish the following lemma.

Lemma 4. *Let $(\xi_t)_{t\in\mathbb{Z}}$ be independent random variables such that $\mathbb{E}(\xi_t) = 0$, $\mathrm{Var}(\xi_t) = 1$ and $\|\xi_t\|_{\psi_2} \leq C$ for some constant $C > 0$. Suppose there are constants $\beta_0, \beta_1, \beta_2, \ldots$ such that $\left|\sum_{\ell=0}^{\infty} \beta_\ell\right| < B$. Define $W_t := \sum_{\ell=0}^{\infty} \beta_\ell \xi_{t-\ell}$ for $t \in \{1, \ldots, n\}$ and let $E = (E_1, \ldots, E_{n-1})$ be defined such that*

$$E_t = \sqrt{\frac{t(n-t)}{n}}\left\{\frac{1}{t}\sum_{i=1}^{t} W_i - \frac{1}{n-t}\sum_{i=t+1}^{n} W_i\right\}.$$

Then there exists a universal constant $K > 0$, such that for any $u > 0$, we have

$$\mathbb{P}(\|E\|_\infty \geq u) \leq 2ne^{-u^2/(K^2B^2C^2)}.$$

Proof. Let $\kappa = (\kappa_1, \ldots, \kappa_n)$ be defined such that

$$\kappa_r = \begin{cases} \sqrt{\frac{n-t}{nt}} & \text{if } 1 \le r \le t \\ \sqrt{\frac{t}{n(n-t)}} & \text{if } t+1 \le r \le n. \end{cases}$$

Then

$$E_t = \sum_{r=1}^{n} \kappa_r W_r = \sum_{r=1}^{n} \kappa_r \sum_{\ell=-\infty}^{\infty} \mathbb{E}(W_r \xi_\ell)\xi_\ell = \sum_{\ell=-\infty}^{\infty} \sum_{r=1}^{n} \kappa_r \mathbb{E}(W_r \xi_\ell)\xi_\ell.$$

Since $(\xi_\ell)_{\ell \in \mathbb{Z}}$ are independent and sub-Gaussian, by the rotational invariance of the sub-Gaussian norm given in Vershynin (2012, Lemma 5.9), we have for some universal constant K that

$$\|E_t\|_{\psi_2}^2 \le K^2 C^2 \sum_{\ell=-\infty}^{\infty} \left\{ \sum_{r=1}^{n} \kappa_r \mathbb{E}(W_r \xi_\ell) \right\}^2 = K^2 C^2 \mathrm{var}(E_t). \tag{8}$$

By construction, for any $1 \le r_1 \le r_2 \le n$, we have

$$\mathrm{cov}(W_{r_1}, W_{r_2}) = \sum_{\ell=0}^{\infty} \beta_\ell \beta_{\ell + r_2 - r_1} =: g(r_2 - r_1).$$

Hence,

$$\begin{aligned} \mathrm{var}(E_t) &= \sum_{r_1=1}^{n} \sum_{r_2=1}^{n} \kappa_{r_1} \kappa_{r_2} g(|r_1 - r_2|) = 2 \sum_{u=1}^{n-1} g(u) \sum_{r=1}^{n-u} \kappa_r \kappa_{r+u} + g(0) \sum_{r=1}^{n} \kappa_r^2 \\ &\le 2 \sum_{u=1}^{n-1} g(u) \left\{ \min\{t-u, 0\} \frac{n-t}{nt} + \min\{n-t-u, 0\} \frac{t}{n(n-t)} \right\} + K(0) \\ &\le 2 \sum_{u=1}^{\infty} g(u) + g(0) = \left(\sum_{\ell=0}^{\infty} \beta_\ell \right)^2 \le B^2. \end{aligned}$$

Substituting the above bound into (8), we obtain that $\|E_t\|_{\psi_2} \le KBC$. By the Markov inequality, we obtain that

$$\mathbb{P}(|E_t| \ge u) = \mathbb{P}\big(e^{E_t^2/(KBC)^2} \ge e^{u^2/(KBC)^2}\big) \le e^{-u^2/(KBC)^2} \mathbb{E} e^{E_t^2/(KBC)^2} \le 2e^{-u^2/(KBC)^2}.$$

The desired result follows by taking a union bound. □

Equipped with the above lemma, we are now in a position to prove Theorems 3.

Proof of Theorem 3. We will choose K to be the same universal constant as in Lemma 4. Consider the event Ω_1 as defined in (5). By Lemma 4 and a union bound, we have for each $t \geq 2$ and $\lambda > 0$ that

$$\mathbb{P}(\Omega_1^c) \leq \sum_{t=2}^{\infty} \mathbb{P}(\|T^{(t)}\|_\infty > \lambda_t) \leq \sum_{t=2}^{\infty} 2te^{-\lambda_t^2/(K^2B^2C^2)} = \sum_{t=2}^{\infty} \frac{2\alpha}{t^2} < \frac{2\alpha}{3}.$$

Then, arguing similarly as in the proof of Theorem 1 up to (6), we have

$$\begin{aligned}\mathbb{P}(\hat{\tau} \leq \tau) &\leq \mathbb{P}(\Omega_1^c) + \sum_{t=\max\{2,b_t\}}^{\infty} \prod_{i=t-b_t+1}^{t} \mathbb{P}(A^{(i)} < 1/2 \mid \Omega_1) \\ &\leq 0.451\alpha + \sum_{t=2}^{\infty} e^{-pb_t/8} \leq \frac{2\alpha}{3} + \sum_{t=2}^{\infty} \frac{\alpha}{2t^2} \leq \alpha,\end{aligned}$$

which establishes the first claim. On Ω_1, arguing in the same way as in Theorem 2, we obtain that $\hat{\tau} \leq \tau + 2^{15/2}\lambda_{2\tau}\sqrt{s\tau}/\eta + b_{2\tau}$ and so the second claim follows. □

References

Aue, A., S. Hörmann, L. Horváth, and M. Reimherr (2009). Break detection in the covariance structure of multivariate time series models. *Ann. Statist.*, 4046–4087.

Barnard, G. A. (1959). Control charts and stochastic processes. *J. Roy. Statist. Soc., Ser. B 21*, 239–271.

Bücher, A., I. Kojadinovic, T. Rohmer, and J. Seger (2014). Detecting changes in cross-sectional dependence in multivariate time series. *J. Mult. Anal. 132*, 111–128.

Cavaliere, G., H. B. Nielsen, and A. Rahbek (2020). Bootstrapping noncausal autoregressions: with applications to explosive bubble modeling. *J. Bus. Econom. Statist. 38*(1), 55–67.

Chen, C. Y.-H. and C. M. Hafner (2019). Sentiment-induced bubbles in the cryptocurrency market. *J. Risk Financ. Manag. 12*(2), 53.

Chen, C. Y.-H., W. K. Härdle, and Y. Klochkov (2021). Sonic: Social network analysis with influencers and communities. *Journal of Econometrics*, to appear.

Chen, Y., T. Wang, and R. J. Samworth (2020). High-dimensional, multiscale online changepoint detection. *arXiv preprint*, arxiv:2003.03668.

Cho, H. (2016). Change-point detection in panel data via double cusum statistic. *Electron. J. Statist. 10*, 2000–2038.

Cho, H. and P. Fryzlewicz (2015). Multiple-change-point detection for high dimensional time series via sparsified binary segmentation. *J. Roy. Statist. Soc., Ser. B 77*, 475–507.

Dette, H., G. Pan, and Q. Yang (2020). Estimating a change point in a sequence of very high-dimensional covariance matrices. *J. Amer. Statist. Assoc.*, to appear.

Duncan, A. J. (1952). *Quality Control and Industrial Statistics*. Chicago: Richard D. Irwin Inc.

Enikeeva, F. and Z. Harchaoui (2019). High-dimensional change-point detection under sparse alternatives. *Ann. Statist. 47*(4), 2051–2079.

Hafner, C. M. (2020). Testing for bubbles in cryptocurrencies with time-varying volatility. *J. Financ. Econom. 18*(2), 233–249.

Han, X., C.-S. Hsieh, and S. I. Ko (2021). Spatial modeling approach for dynamic network formation and interactions. *J. Bus. Econom. Statist. 39*, 120–135.

Horváth, L. and M. Hušková (2012). Change-point detection in panel data. *J. Time Series Anal. 33*, 631–648.

Jirak, M. (2015). Uniform change point tests in high dimension. *Ann. Statist. 43*, 2451–2483.

Jochmans, K. (2018). Semiparametric analysis of network formation. *J. Bus. Econom. Statist. 36*(4), 705–713.

Lavielle, M. and G. Teyssiere (2006). Detection of multiple change-points in multivariate time series. *Lith. Math. J. 46*, 287–306.

Marchal, O., J. Arbel, et al. (2017). On the sub-gaussianity of the beta and dirichlet distributions. *Electron. Commun. Probab. 22*.

Page, E. S. (1954). Continuous inspection schemes. *Biometrika 41*, 100–115.

Preuß, P., R. Puchstein, and H. Dette (2015). Detection of multiple structural breaks in multivariate time series. *J. Amer. Statist. Assoc. 110*, 654–668.

Vershynin, R. (2012). Introduction to the non-asymptotic analysis of random matrices. In Y. Eldar and G. Kutyniok (eds.). *Compressed Sensing, Theory and Applications*, 210–268.

Wang, D., Y. Yu, and A. Rinaldo (2021). Optimal change point detection and localization in sparse dynamic networks. *Ann. Statist. 49*, 203 – 232.

Wang, D., Y. Yu, A. Rinaldo, and R. Willett (2019). Localizing changes in high-dimensional vector autoregressive processes. *arXiv preprint*, arxiv:1909.06359.

Wang, T. and R. Samworth (2016). *InspectChangepoint: high-dimensional changepoint estimation via sparse projection.* R package version 1.1.

Wang, T. and R. J. Samworth (2018). High dimensional change point estimation via sparse projection. *J. Roy. Statist. Soc., Ser. B 80*(1), 57–83.

Zhang, N. R., D. O. Siegmund, H. Ji, and J. Z. Li (2010). Detecting simultaneous changepoints in multiple sequences. *Biometrika 97*, 631–645.

Zhu, X., R. Pan, G. Li, Y. Liu, H. Wang, et al. (2017). Network vector autoregression. *Ann. Statist. 45*(3), 1096–1123.

Zou, C., Z. Wang, X. Zi, and W. Jiang (2015). An efficient online monitoring method for high-dimensional data streams. *Technometrics 57*, 374–387.