# CHANGE-POINT DETECTION OF GAUSSIAN GRAPH SIGNALS WITH PARTIAL INFORMATION

*Yanxi Chen, Xianghui Mao, Dan Ling, and Yuantao Gu*

Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

**ABSTRACT**

In a change-point detection problem, a sequence of signals switches from one distribution to another at an unknown time step, and the goal is to quickly and reliably detect this change. By providing new insight into signal processing and data analysis, graph signal processing promises various applications including image processing and sensor network analysis, and becomes an emerging field of research. In this work, we formulate the problem of change-point detection on graph. Under the reasonable assumption of normality, we propose a CUSUM-based algorithm for change-point detection with an arbitrary, unknown and perhaps time-varying mean shift after the change-point. We further propose a decentralized, distributed algorithm, which requires no fusion center, to reduce computational complexity, as well as costs and delays of communication. Numerical results on both synthetic and real-world data demonstrate that our algorithms are efficient and accurate.

***Index Terms*—** Change-point detection, graph signal processing, CUSUM, distributed algorithm

## 1. INTRODUCTION

Detecting the abrupt change of distribution in a sequence of signals and data is a fundamental problem in various applications, such as quality control [1] and anomaly detection [2]. Numerous works have been devoted to the problem of change-point detection, among which methods based on cumulative sum (CUSUM) [3] are well-formulated and widely used. Moreover, CUSUM has a recursive form and therefore can proceed in an online manner, which enjoys memory and computation efficiency. While CUSUM was first proposed to tackle with a single data stream, CUSUM-based detection algorithms utilizing information from multiple sensors, i.e., in a high-dimensional situation, have been proposed [2, 4, 5, 6, 7] to handle the increasingly complicated modern sensing systems. [7] proposed a fully distributed algorithm that requires only communication between neighboring vertices and no fusion center, which reduces the computational complexity, as well as costs and delays of communication. While CUSUM requires knowledge of pre-change and post-change distributions, parameters of the post-change distribution are usually unknown in practice. One typical method to tackle this problem is generalized likelihood ratio (GLR) [8], which replaces unknown parameters with their maximum likelihood estimates using previous signal values. Algorithms that conduct joint estimation and detection based on either GLR [6, 9] or other methods [10] have been proposed, but they are generally less efficient than pure CUSUM.

The authors are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China. This work was partially supported by National Natural Science Foundation of China (NSFC 61571263, 61531166005), the National Key Research and Development Program of China (Project No. 2016YFE0201900, 2017YFC0403600), and Tsinghua University Initiative Scientific Research Program (Grant 2014Z01005). The corresponding author of this work is Y. Gu (E-mail: gyt@tsinghua.edu.cn).

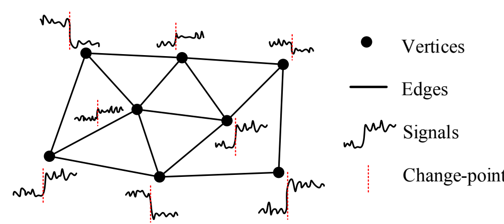



**Fig. 1**. Change-point detection on graph.

In many situations, a sensor network can be modeled as a graph, where sensors taking sequential measurements of different variables can be viewed as vertices of the graph. Naturally, the measurements residing on the vertices can be modeled as a graph signal [11]. Graph signal processing (GSP) [11, 12] is an emerging field of research in recent years. Classic concepts in signal processing, such as Fourier transform and filtering, can be extended to the framework of GSP.

In this paper, we formulate the problem of change-point detection of Gaussian graph signals. Fig. 1 is an illustrative example. We assume that the variance and pre-change mean of the graph signal are known, but the post-change mean is arbitrary, unknown and perhaps time-varying. In this situation, we first propose a centralized algorithm, which conducts detection *without* estimation. Then, in light of the distributed algorithm in a recent work [7], we propose a decentralized, distributed variant of our algorithm that fits in our problem formulation. Finally, we validate the effectiveness of our methods with both synthetic and real-world data.

This work is different from previous works on change-point detection in three ways. First, for the case of unknown parameters of post-change distribution, while previous works conducted joint estimation and detection, we obtain a qualified score for CUSUM by operations of maximization and correction, and conduct detection without estimation. Therefore, our algorithm is as efficient as pure CUSUM, and able to directly handle the case of time-varying post-change parameters. Second, compared with previous works in a high-dimensional setting, we adopt the framework of GSP and utilize the graph structure. *A priori* on the Fourier transform of graph signals can help to improve performance of our algorithm. Finally, while our distributed algorithm is a direct extension of the recent work [7], the problem settings are different, as [7] assumes that measurements are *i.i.d.* among both time steps and sensors, and that pre-change and post-change distributions are known, while based on an additional assumption of normality, we assume that post-change parameters are unknown and even time-varying.

## 2. PRELIMINARY

### 2.1. Change-point detection

For time $t = 1, 2, 3, \ldots$, we have a sequence of independent signals $\mathbf{x}^t \in \mathbb{R}^N$, where $N = 1$ for the scalar case and $N > 1$ for high-dimensional case. Given a change-point $T_c$, for $1 \leq t < T_c$, $\mathbf{x}^t \sim P_0$ under hypothesis $H_0$; for $t \geq T_c$, $\mathbf{x}^t \sim P_1$ under $H_1$. The goal is to detect the change at stopping time $T_s$, and achieve a small detection delay $T_s - T_c$ while keeping a low false alarm rate.

**Definition 1** (CUSUM [3]). Given measurements $\{\mathbf{x}^t\}$, we assign a score $L^t$ to each measurement, which is negative or around zero under $H_0$ and positive under $H_1$. Then $T_s = \inf\{t > 0 : \max_{1 \leq i \leq t} \sum_{k=i}^{t} L^k \geq b\}$ for some threshold $b$.

A commonly used score for CUSUM is log-likelihood ratio (LLR), $L^t = \log(f_1(\mathbf{x}^t)/f_0(\mathbf{x}^t))$, where $f_0$ and $f_1$ are probability density functions of distributions $P_0$ and $P_1$, respectively.

The recursive form of CUSUM [3] is as follows. We initialize the statistic $y^0 = 0$, then $y^t = \max\{y^{t-1} + L^t, 0\}$ for $t = 1, 2, \ldots$, and $T_s = \inf\{t > 0 : y^t \geq b\}$. The recursive form of CUSUM proceeds in an online manner and is very efficient in time and space.

**Definition 2** (ARL [3]). Average running length $\mathrm{ARL} = \mathbb{E}[T_s]$ is the expected number of time steps before the algorithm detects the change.

ARL is a popular measure to evaluate a change-point detection algorithm. One interesting case is when hypothesis $H_0$ holds all the time, and we denote it as $\mathrm{ARL}_0$. This can be viewed as the expected number of time steps before a false alarm occurs. Another interesting case is when $H_1$ holds all the time, and we denote it as $\mathrm{ARL}_1$. This can be viewed as the average detection delay. A good detection algorithm should achieve a large $\mathrm{ARL}_0$ to control false alarm rate, and a small $\mathrm{ARL}_1$ to quickly response to the change.

### 2.2. Graph signal processing

In GSP, a signal $\mathbf{x} \in \mathbb{R}^N$ is defined on a graph $G = (V, E)$ with $|V| = N$ vertices. The graph can be represented by the adjacent matrix $\mathbf{A} \in \{0, 1\}^{N \times N}$, where $A_{ij} = 1$ if vertex $i$ and $j$ are connected by an edge and $A_{ij} = 0$ otherwise. The graph Laplacian is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ is diagonal and $D_{ii} = \sum_{j=1}^{N} A_{ji}$. For an undirected graph, $\mathbf{L}$ is symmetric and semi-positive definite. We refer to the eigen-decomposition as $\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}}$, where eigenvalues in $\mathbf{\Lambda}$ is sorted in ascend.

**Definition 3** (Fourier transform [12]). The Fourier transform of a graph signal $\mathbf{x}$ is $\hat{\mathbf{x}} = \mathbf{V}^{\mathrm{T}}\mathbf{x}$, and the inverse transform is $\mathbf{x} = \mathbf{V}\hat{\mathbf{x}}$. $\mathbf{x}$ is $K$-bandlimited if $\hat{x}_i = 0, \forall i \in \{K+1, \ldots, N\}$.

In practice, the graph signal is usually smooth and (approximately) bandlimited with appropriately constructed graph structure.

## 3. ALGORITHMS FOR CHANGE-POINT DETECTION

A sequence of graph signals $\mathbf{x}^t \in \mathbb{R}^N$ *i.i.d.* follow distribution $P_0$ for $t < T_c$ and $P_1$ for $t \geq T_c$, where $P_0$ and $P_1$ are Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$ with $\boldsymbol{\mu} = \boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$, respectively. $\mathbf{\Sigma}$ is diagonal and $\Sigma_{ii} = \sigma_i^2, i = 1, 2, \ldots, N$. For simplicity, we assume that $\sigma_i^2 = \sigma^2, \forall i \in \{1, 2, \ldots, N\}$, although we will show that the analysis can be easily extended to the more general case of $\mathbf{\Sigma}$. We also assume that $\boldsymbol{\mu}_0$ and $\sigma^2$ are known, as they can be estimated from historical data. However, $\boldsymbol{\mu}_1$ is unknown to us, because it is the result of an unexpected and unpredictable event. Moreover, the post-change mean can be time-varying, and we will show that our analysis and algorithms still apply to this case.

**Algorithm 1** Centralized Change-point Detection

**Input:** Number of vertices $N$, $\boldsymbol{\mu}_0$ and its bandwidth $K$, Gaussian variance $\sigma^2$, threshold $b$.

```
Initialize: y^0 = 0.
for t = 1, 2, ... do
    Projection: r ← x^t − μ_0, r̂ ← V^T r, r̂_h ← r̂_{K+1:N}.
    LLR: L^t ← ||r̂_h||_2^2 / (2σ^2) .
    Correction: L^t ← L^t − (N − K)/2.
    CUSUM: y^t ← max(y^{t−1} + L^t, 0).
    Inference: if y^t ≥ b, then T_s ← t, detection is done.
end for
```

**Output:** Stopping time $T_s$.

### 3.1. Case 1: emerging high-frequency component

A special case of concern is that, in a normal state ($H_0$), the signal is smooth and bandlimited, and in the abnormal state ($H_1$), a high-frequency, non-smooth component (e.g., an anomaly) appears and results in a full-band signal. In this case, $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_0 + \mathbf{V}\hat{\boldsymbol{\mu}}_h$, where $\boldsymbol{\mu}_0$ is $K$-bandlimited and $\hat{\boldsymbol{\mu}}_h$ is the Fourier transform of the emerging high-frequency component, i.e., for any $i$ in $\{1, 2, \ldots, K\}$, we have that the $i$-th entry of $\hat{\boldsymbol{\mu}}_h$ is zero. Under $H_0$, there is $\mathbf{x}^t = \boldsymbol{\mu}_0 + \mathbf{e}^t$, where $\mathbf{e}^t \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, and LLR $L^t = \log(f_1/f_0) = (\|\mathbf{e}^t\|_2^2 - \|\mathbf{e}^t - \mathbf{V}\hat{\boldsymbol{\mu}}_h\|_2^2)/(2\sigma^2)$. However, $\hat{\boldsymbol{\mu}}_h$ is unknown. Our strategy is to take the *maximum*,

$$L^t = \max_{\hat{\boldsymbol{\mu}}_h} \frac{\|\mathbf{e}^t\|_2^2 - \|\mathbf{e}^t - \mathbf{V}\hat{\boldsymbol{\mu}}_h\|_2^2}{2\sigma^2} = \frac{\|\hat{\mathbf{e}}_h^t\|_2^2}{2\sigma^2}, \tag{1}$$

where $\hat{\mathbf{e}}_h^t$ is obtained by setting the first $K$ entries of $\hat{\mathbf{e}}^t = \mathbf{V}^{\mathrm{T}}\mathbf{e}^t$ to be zero. This can be viewed as *projecting* the noise onto the high-frequency subspace. To make $L^t$ a qualified score in CUSUM, we need to make the *correction*

$$L^t = \frac{\|\hat{\mathbf{e}}_h^t\|_2^2}{2\sigma^2} - \frac{N-K}{2}, \tag{2}$$

so that the expectation $\mathbb{E}[L^t|H_0] = 0$. Similarly, under $H_1$, $\mathbf{x}^t = \boldsymbol{\mu}_0 + \mathbf{V}\hat{\boldsymbol{\mu}}_h + \mathbf{e}^t$, $L^t = \|\hat{\mathbf{e}}_h^t + \hat{\boldsymbol{\mu}}_h\|_2^2/(2\sigma^2) - (N-K)/2$, and $\mathbb{E}[L^t|H_1] = \|\hat{\boldsymbol{\mu}}_h\|_2^2/(2\sigma^2) > 0$, which has the form of *signal-noise-ratio* (SNR). The complete CUSUM-based algorithm is demonstrated in Algorithm 1. It is a centralized algorithm, since the computation of $L^t$, update of $y^t$ and inference are conducted by a center collecting information from all vertices.

We provide some further explanations for the proposed algorithm. The operation of maximization in (1) actually says that we replace (not estimate) the unknown $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$ with the high-frequency component of the current measurement $\mathbf{x}^t$, including the noise. Compared with GLR, the proposed method is more flexible and efficient in computation and memory as no historical observations are cached for estimating $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$. In other words, the performance of Algorithm 1 do not rely on the quality of the estimation of unknown post-change parameters. As a result, Algorithm 1 can even deal with the case when the post-change mean $\boldsymbol{\mu}_1$ is varying over time, which is difficult for joint estimation and detection algorithms. For simplicity of analysis, by default we still assume that the post-change mean is constant over time in the remaining of this paper.

*Remark* 1. Case 1 essentially says that $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$ lies in a subspace spanned by a subset $\Omega \subseteq \{1, 2, \dots, N\}$ of $\mathbf{V}$'s columns. By default we assume that $\Omega = \{K+1, \dots, N\}$, which is unnecessary. For example, another important case is when all vertices share the same mean shift, in which case $\Omega = \{1\}$.

### 3.2. Case 2: arbitrary change

Now we consider a more general case where $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$ are both arbitrary and full-band. Without loss of generality, we let $\boldsymbol{\mu}_0 = \mathbf{0}$, because we can otherwise conduct detection from $\mathbf{x}^t - \boldsymbol{\mu}_0$. The trick is that, since $\boldsymbol{\mu}_0 = 0$ and $\boldsymbol{\mu}_1$ is full-band, we set bandwidth $K = 0$ in previous analysis, and simplify score $L^t$ in Algorithm 1 as

$$L^t = \frac{\|\hat{\mathbf{r}}_h\|_2^2}{2\sigma^2} - \frac{N-K}{2} = \frac{\|\mathbf{r}\|_2^2}{2\sigma^2} - \frac{N}{2}. \tag{3}$$

The Fourier transform and projection steps can be omitted. This further simplifies the computation, and in this case, Algorithm 1 is as efficient as the original recursive form of CUSUM.

*Remark* 2. The analysis can be easily extended to the more general case when $\sigma_i^2$ is not necessarily equal to $\sigma_j^2$, $i \neq j$. We modify (3) as $L^t = (\sum_{i=1}^N r_i^2/\sigma_i^2 - N)/2$. It is easy to check that, $\mathbb{E}[L^t|H_0] = 0$, and $\mathbb{E}[L^t|H_1] = \sum_{i=1}^N (\mu_{1i}^2/(2\sigma_i^2))$. In the remaining of this paper, by default we still assume that $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$ for simplicity.

### 3.3. A decentralized distributed algorithm

While the centralized algorithm is recursive and efficient already, in a large-scale network the complexity of computation and, more importantly, the costs and delays of communication, can be prohibitive for deployment of a sensor system in practice. We propose a decentralized distributed algorithm for detection in case 2, as demonstrated in Algorithm 2, which is a direct extension of [7]. The only difference is how the CUSUM score $L_v^t$ is obtained: in [7] it is assumed that LLR is readily available, while in Algorithm 2 we use the method of maximization and correction to handle unknown post-change parameters, provided the additional assumption of normality. Note that one result of the maximization in inference statistic $\max_v z_v^t$ is that, it will slowly increase under $H_0$ even when we make the correction on $L_v^t$ and have $\mathbb{E}[L^t|H_0] = 0$. Therefore, we need an over-correction of $L_v^t$ with a small positive parameter $\delta$ such that $\mathbb{E}[L^t|H_0] = -\delta < 0$ and thus the increase under $H_0$ is suppressed, while the detection delay is only mildly affected if SNR is relatively high [1]. Provided $L_v^t$, the remaining procedure is the same as in [7]. Each vertex conducts local CUSUM computation, and communicates the increase in statistic $y_v^t$ with vertices in its neighbor set $N(v)$ (including vertex $v$ itself). When the statistic $z_v^t$ of any one of the vertices exceeds the threshold, it alarms and the detection is done. Note that the non-negative weight matrix of communication $\mathbf{W}$ satisfies $\sum_{u \in N(v)} W_{vu} = 1, \forall v$. Under the framework of GSP, we have a natural link between $\mathbf{W}$ and the graph structure, i.e., $W_{vu}$ is non-zero if and only if $v = u$ or $v$ and $u$ are connected by an edge.

**Algorithm 2** Decentralized Distributed Change-point Detection

**Input:** Number of vertices $N$, Gaussian variance $\sigma^2$, threshold $b$, over-correction parameter $\delta$.

Initialize: $y_v^0 = 0, z_v^0 = 0, \forall v \in \{1, 2, \dots, N\}$.
**for** $t = 1, 2, \dots$ **do**
  **for** $v \in \{1, 2, \dots, N\}$ **do**
    LLR: $L_v^t \leftarrow |x_v^t|^2/(2\sigma^2)$.
    Correction: $L_v^t \leftarrow L_v^t - 1/2 - \delta$.
    Local CUSUM: $o_v^t \leftarrow y_v^{t-1}$, $y_v^t \leftarrow \max(y_v^{t-1} + L_v^t, 0)$.
    Communication: $z_v^t \leftarrow \sum_{u \in N(v)} W_{vu}(z_u^{t-1} + y_u^t - o_u^t)$.
    Inference: if $\max_v z_v^t \geq b$, then $T_s \leftarrow t$, detection is done.
  **end for**
**end for**

**Output:** Stopping time $T_s$.

### 3.4. Discussions

#### 3.4.1. *A first-step analysis on* $\mathrm{ARL}_0$ *and* $\mathrm{ARL}_1$

While rigorous analysis on the lower bound of $\mathrm{ARL}_0$ and upper bound of $\mathrm{ARL}_1$ is beyond the scope of this short paper, we briefly provide some hints. Here, we focus on the centralized algorithm in case 1. For $\mathrm{ARL}_0$, we are concerned with the distribution of $L^t$ under $H_0$. Since $\mathbf{r} = \mathbf{e}$ in this case, (3) becomes $L^t = \|\hat{\mathbf{e}}_h^t\|_2^2/(2\sigma^2) - (N-K)/2 = (\sum_{i=K+1}^N (\hat{e}_i^t/\sigma)^2 - (N-K))/2$, where $\hat{e}_i^t$ denotes the $i$-th entry of the vector $\hat{\mathbf{e}}^t$, and there is $\hat{e}_i^t/\sigma \sim \mathcal{N}(0,1)$. That is, $L^t$ follows Chi-square distribution with degree of freedom $N-K$, with simple scale and shift. Notice that the distribution is only determined by $N-K$, and has nothing to do with noise variance or other parameters. According to Lemma 1 in [13], we can bound the tails of the distribution of $L^t$ as follows.

**Lemma 1.** *Under* $H_0$, *we have* $P(L^t \geq \sqrt{(N-K)c} + c) \leq \mathrm{e}^{-c}$, *and* $P(L^t \leq -\sqrt{(N-K)c}) \leq \mathrm{e}^{-c}$.

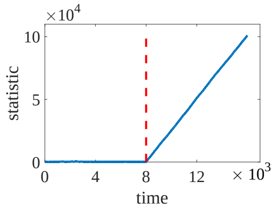
As for $\mathrm{ARL}_1$, given $\mathbb{E}[L^t|H_1] = \|\hat{\boldsymbol{\mu}}_h\|_2^2/(2\sigma^2)$, for sufficiently large threshold $b$, we can estimate the detection delay as $T_s - T_c \approx b/\mathbb{E}[L^t|H_1] = 2b\sigma^2/\|\hat{\boldsymbol{\mu}}_h\|_2^2$. We note that case 1 is still meaningful, even when case 2 is more general. In case 1, we utilize the graph structure, and project noise onto a subspace, which reduces the effect of noise compared with case 2. That is, the distribution of $L^t$ is more concentrated around its mean, while $\mathbb{E}[L^t|H_0]$ and $\mathbb{E}[L^t|H_1]$ remain unchanged. This leads to a lower false alarm rate, and more stable linear increase in $y^t$. Therefore, when we know *a priori* that $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$ lies in a subspace spanned by only a known subset of $\mathbf{V}$'s columns, then we still prefer to apply case 1 to improve performance in ARL.

As for distributed algorithm, we refer readers to [7] for a more complete analysis of its behaviors and performance. However, to be noted, in our problem, $\mu_{1v}$ varies for different vertex $v$, which results in various increaments of $y_v^t$ among vertices under $H_1$. When vertices are densely connected, there will be an effect of average in the communication step of Algorithm 2, and the expected increament of statistic $\max_v z_v^t$ at a time step can be estimated as $\|\boldsymbol{\mu}_1\|_2^2/(2N\sigma^2) - \delta$. On the other hand, when the graph is sparse, vertex $v$ with a large $\mu_{1v}$ will play the major role, leading to a faster increase of $\max_v z_v^t$ under $H_1$ and thus a smaller detection delay. The cost, however, is that the effect of noise under $H_0$ is also less averaged, and therefore false alarm rate may increase.

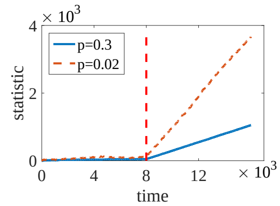
#### 3.4.2. *Scalability*

From section 3.4.1, as size of the graph $N$ increases, $L^t$ is less concentrated around its mean under $H_0$. This seems to result in a higher false alarm rate. However, in many cases it is reasonable to assume that $\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\|_2^2$ (and thus $\mathbb{E}[L^t|H_1]$) also increases with $N$. Therefore, for a large-scale network we just need to accordingly set a higher threshold $b$ to achieve similar performance in ARL.

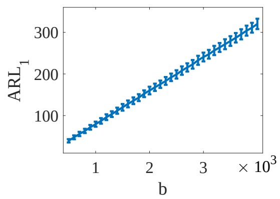
[1] It is also reasonable to make over-correction in the centralized algorithm to further lower false alarm rate, although it is not as necessary and essential as in the decentralized case.



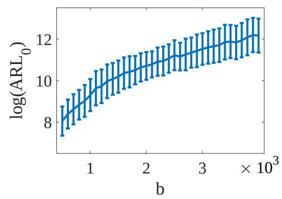

(a) $y^t$ for Algorithm 1.




(b) $\max_v z_v^t$ for Algorithm 2.




(c) $\mathrm{ARL}_1$-$b$ curve.




(d) $\log(\mathrm{ARL}_0)$-$b$ curve.

**Fig. 2**. Results of the detection process on synthetic data.

### 3.4.3. *Extension to general distributions*

Although we assume normality in this paper, we briefly show how to extend the method to the case of more general distributions. Here we focus on Algorithm 1 for case 2. For general $P_0$, $P_1$, $\mathrm{LLR} = \mathrm{LLR}(\mathbf{x}^t, \theta_1) = \log(f_1(\mathbf{x}^t|\theta_1)/f_0(\mathbf{x}^t))$, where $\theta_1$ denotes unknown post-change parameters in $f_1$. By maximization and correction, we set the score for CUSUM as $L^t = \max_{\theta_1} \mathrm{LLR}(\mathbf{x}^t, \theta_1) - C$, for some constant $C$, such that $\mathbb{E}[L^t|H_0] \leq 0$ and $\mathbb{E}[L^t|H_1] > 0$. The remaining procedure in Algorithm 1 remains the same.

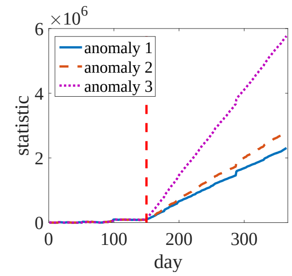
## 4. NUMERICAL EXPERIMENTS

### 4.1. Synthetic data

We test our methods on an undirected graph with $N = 100$ vertices, where each pair of vertices are connected by an edge with probability $p = 0.3$. We focus on case 2, i.e., we set $\boldsymbol{\mu}_0 = \mathbf{0}$, and $\boldsymbol{\mu}_1$ is randomly generated and scaled such that $\|\boldsymbol{\mu}_1\|_2 = 1$. White Gaussian noise level $\sigma = 0.2$. In this setting, $\boldsymbol{\mu}_1$ is immersed in noise. We first set $T_c = 8000$ and see how Algorithm 1 and 2 work out. Then we set $T_c = 1$ and $T_c = \infty$, respectively, to examine $\mathrm{ARL}_1$ and $\mathrm{ARL}_0$ of the centralized algorithm. The optimal design of $\mathbf{W}$ in the distributed algorithm is beyond the scope of this paper, and we simply set $W_{vu} = 1/|N(v)|$.

Numerical results are illustrated in Fig. 2. (a) shows that statistic $y^t$ in Algorithm 1 stays low before $T_c$ (indicated by the vertical dashed line), and linearly increases after $T_c$, with a slope of $12.5 = \|\boldsymbol{\mu}_1\|_2^2/(2\sigma^2)$, as expected. (b) shows similar results for Algorithm 2. Here we set $\delta = 0$, i.e., no over-correction, to see how the statistic slowly increases even under $H_0$. The slope of $\max_v z_v^t$ after $T_c$ is approximately $0.125 = \|\boldsymbol{\mu}_1\|_2^2/(2N\sigma^2)$. In comparison, for a sparse graph with $p = 0.02$, both $\mathrm{ARL}_0$ and $\mathrm{ARL}_1$ decrease with the same $b$, and the result is less stable, which validates our analysis in section 3.4.1. (c) and (d) are the curves of $\mathrm{ARL}_1$ and $\mathrm{ARL}_0$ (in natural logarithm), with threshold $b$ varying from 500 to 4000. The emperical means and standard errors are obtained from 400 independent trials. (c) shows that $\mathrm{ARL}_1$ is linear in $b$, as expected. More interestingly, with sufficiently large $b$, $\mathrm{ARL}_0$ seems to be exponential in $b$, though we have not rigorously proven this.

(a) Manhattan road map.




(b) Statistic $y^t$.

**Fig. 3**. Change-point detection on Manhattan taxi pickup data.

### 4.2. Real-world data

We test the centralized algorithm (general $\boldsymbol{\Sigma}$, case 2) in anomaly detection of daily taxi pickups in New York City. We view the road map of Manhattan, as illustrated in Fig. 3 (a), as a graph, with $N = 13679$ intersections being its vertices. Each taxi pickup is assigned to the closest intersection, and signal $\mathbf{x}^t \in \mathbb{N}^N$ consists of taxi pickup numbers at each intersection in day $t$. Given $\{\mathbf{x}^t\}_{t=1}^{365}$ in year 2014, we estimate $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}$ by standard statistical routines. For vertices with constant zero taxi pickup, we set $\sigma_i^2 = 0.01$. Then, we simulate three additive anomalies in the data of 2015 after $T_c = 150$: (1) add a constant 5 to the daily taxi pickup numbers of 112 vertices, indicated by green color in Fig. 3 (a); (2) add an increament to green vertices, uniformly drawn from $\{1, 2, \ldots, 9\}$ and independent among time steps and vertices; (3) add an increament to 112 randomly chosen vertices, uniformly and independently drawn from $\{1, 2, 3, 4\}$. Parameters of these three anomalies are unknown to the algorithm. Also, they are of small sizes, compared with the original daily taxi pickups.

Numerical results are demonstrated in Fig. 3 (b). (Note that result for the third anomaly varies depending on the set of vertices chosen.) It shows that statistic $y^t$ linearly increases after $T_c$ for all three cases, which indicates our algorithm's ability to detect various types of anomalies with unknown parameters. Note that the performance of our algorithm in anomaly detection does not rely on the size of anomaly alone, but instead on SNR, which ensures its success with an anomaly of relatively small size.

## 5. CONCLUSION

In this work, we formulate the problem of change-point detection for Gaussian graph signals. In the case of unknown and perhaps time-varying post-change mean, we use a method of maximization and correction to obtain a qualified score for CUSUM, and conduct detection without estimation. We further propose a decentralized distributed algorithm, in light of [7], to reduce the costs and delays of computation and communication. Future work includes rigorous analysis on $\mathrm{ARL}_0$ and $\mathrm{ARL}_1$ of both centralized and decentralized algorithms, the optimal choice of communication weight matrix $\mathbf{W}$ in the distributed algorithm, and change-point detection of graph signals with general pre-change and post-change distributions.

## 6. ACKNOWLEDGEMENTS

The authors would like to thank Siheng Chen for providing preprocessed data of Manhattan taxi pickups in 2014 and 2015.

## 7. REFERENCES


[1] Tze Leung Lai, "Sequential changepoint detection in quality control and dynamical systems," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 57, no. 4, pp. 613–658, 1995.

[2] A. G. Tartakovsky, B. L. Rozovskii, R. B. Blazek, and Hongjoong Kim, "A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods," *IEEE Transactions on Signal Processing*, vol. 54, no. 9, pp. 3372–3382, Sept 2006.

[3] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954.

[4] Alexander G Tartakovsky and Venugopal V Veeravalli, "Quickest change detection in distributed sensor systems," in *Proceedings of the 6th International Conference on Information Fusion*, 2003, pp. 756–763.

[5] Yajun Mei, "Efficient scalable schemes for monitoring a large number of data streams," *Biometrika*, vol. 97, no. 2, pp. 419–433, 2010.

[6] Yao Xie, Jiaji Huang, and Rebecca Willett, "Change-point detection for high-dimensional time series with missing data," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 1, pp. 12–27, 2013.

[7] Qinghua Liu and Yao Xie, "Fully distributed multi-sensor change-point detection," *arXiv preprint arXiv:1710.10378*, 2017.

[8] G. Lorden, "Procedures for reacting to a change in distribution," *The Annals of Mathematical Statistics*, vol. 42, no. 6, pp. 1897–1908, 1971.

[9] Tze Siong Lau, Wee Peng Tay, and Venugopal V Veeravalli, "Quickest change detection with unknown post-change distribution," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3924–3928.

[10] Yang Cao, Liyan Xie, Yao Xie, and Huan Xu, "On near optimality of one-sample update for joint detection and estimation," *arXiv preprint arXiv:1705.06995*, 2017.

[11] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[12] Aliaksei Sandryhaila and José MF Moura, "Discrete signal processing on graphs," *IEEE transactions on signal processing*, vol. 61, no. 7, pp. 1644–1656, 2013.

[13] B. Laurent and P. Massart, "Adaptive estimation of a quadratic functional by model selection," *The Annals of Statistics*, vol. 28, no. 5, pp. 1302–1338, 2000.