

ORIGINAL RESEARCH ARTICLE

Adaptive Resources Allocation CUSUM for Binomial Count Data Monitoring with Application to COVID-19 Hotspot Detection

Jiuyun Hu^a, Yajun Mei^b, Sarah Holte^c, Hao Yan^a

^aSchool of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ, USA; ^cSchool of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA; ^cDivision of Public Health Sciences, Fred Hutchinson Cancer Research Center, Seattle, WA, USA

ARTICLE HISTORY

Compiled January 18, 2023

ABSTRACT

In this paper, we present an efficient statistical method (denoted as "Adaptive Resources Allocation CUSUM") to robustly and efficiently detect the hotspot with limited sampling resources. Our main idea is to combine the multi-arm bandit (MAB) and change-point detection methods to balance the exploration and exploitation of resource allocation for hotspot detection. Further, a Bayesian weighted update is used to update the posterior distribution of the infection rate. Then, the upper confidence bound (UCB) is used for resource allocation and planning. Finally, CUSUM monitoring statistics to detect the change point as well as the change location. For performance evaluation, we compare the performance of the proposed method with several benchmark methods in the literature and showed the proposed algorithm is able to achieve a lower detection delay and higher detection precision. Finally, this method is applied to hotspot detection in a real case study of county-level daily positive COVID-19 cases in Washington State WA) and demonstrates the effectiveness with very limited distributed samples.

KEYWORDS

Multi-arm bandit, change point detection, adaptive resources allocation, count data, CUSUM Statistics

1. Introduction

Nowadays, streaming data from multiple data sources or streams have become more and more common in public health surveillance. Rapid *hotspot* detection for the streaming data across different spatial regions over time is important and can provide valuable information to the decision-makers to mitigate the risk as soon as possible. In this paper, we define the *hotspot* as the structured outliers that are persistent after a certain time point. Besides, we also assume that full observation is not possible due to the limited sampling resources.

A motivating example for this research is to monitor the number of COVID-19 confirmed cases for different spatial regions in the United States. Monitoring the entire population is not possible due to the limited testing resources or labels. Here, we would like to focus on adaptively distributing the testing resources such that the hotspot can

be detected as soon as possible. Please see Section 2 for a more detailed description of the COVID-19 monitoring problem.

Generally speaking, hotspot detection in spatio-temporal data can be treated as a change-point detection problem. Take the COVID-19 monitoring as an example. There are three types of challenges for the hotspot detection problem: 1) High-dimensionality: the number of sites or locations to be monitored is often quite large. 2) Spatial sparsity of the hotspot: the affected counties are often quite sparse in the high-dimensional space. It implies that the counties undergoing significant changes in the number of confirmed cases should be small at certain time points. 3) Temporal consistency: the hotspots should last for a reasonably long period of time.

Besides the aforementioned challenges, there are also additional challenges in sequential change detection related to the limited sensing resource as follows: 1) Lack of access to the fully observational data: existing research for high dimensional sequential sparse change detection focuses on a fully observable process, i.e., at each sampling time point, all the variables can be observed for analysis. However, in reality, it is infeasible to acquire full measurements for the entire population. 2) Partial sampling with count data observation: existing research often focuses on the continuous observational data and often assumes that a location is either observed or not observed. However, in many examples, such as the COVID-19 monitoring case, the decision variables are not binary, which implies that you can distribute more test kits to a particular region to obtain a more accurate estimation of the percentage of the infected person. In such cases, the distribution of the sampling resources is much more complicated than simply deciding whether a region should be observed or not. 3) Adaptive decision-making to balance the exploration and exploitation: On one hand, we would distribute more tests in the particular region that has been detected with a high infection rate. On the other hand, other unobserved regions may be potentially at risk as well, and some testing resources should be distributed in these regions as well. Overall, the trade-off between exploration and exploitation is a fundamental challenge in multi-arm bandit (MAB) and reinforcement learning. Recently, Such a balance is also an important aspect to be considered in adaptive resource allocation problems for change point detection, e.g., in [35, 37, 14, 40]. Take Figure 1 as an example, infection rates in Florida and Louisiana were higher than in other states on Sep 13, 2020. However, the resource allocation algorithm should distribute enough samples randomly to all states to discover such a fact (i.e., exploration) before distributing more tests to these two states specifically (i.e., exploitation) for final confirmation and change point detection.

This paper aims to develop an adaptive resource allocation method to automatically distribute the number of testing resources into multiple regions for the quickest change point and hotspot detection. More specifically, we will focus on the case where the data follows a binomial distribution, where the total number of tests for each region i (i.e., n_i) depends on the number of testing resources distributed to region i . Overall, we propose to borrow the idea from the Upper Confidence Bound (UCB) method in solving the MAB problem to automatically distribute the testing resources to balance the exploration and exploitation. Bayesian statistics are used to update the estimation of the infection rate online with the consideration of uncertainty. A "reward" function is proposed based on the CUSUM statistics such that the change point can be identified as soon as possible.

Intuitively, the algorithm should work adaptively. Typically, the testing resources should be distributed equally to different regions in the early stages (i.e., exploration). If some regions with potentially high infection rates are observed, more testing resources should be distributed in these regions for confirmation (i.e., exploitation).

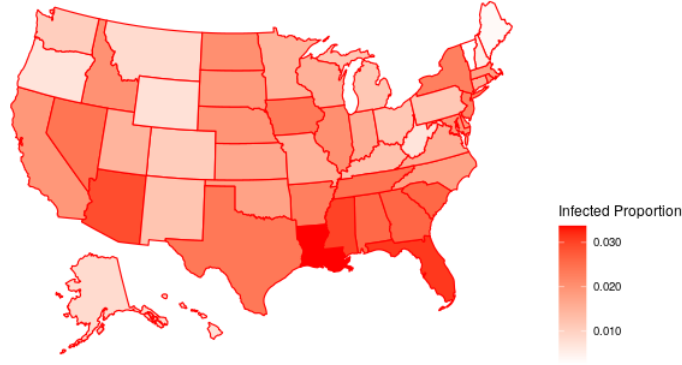


Figure 1.: Infection Rate on Sep 13, 2020 in the United States

Furthermore, for resource allocation, we will borrow the idea from MAB. MAB aims to sequentially allocate a limited set of resources between competing "arms" to maximize their expected gain, where each arm's reward function is unknown. MAB provides a principled way to balance exploration and exploitation. To apply MAB for change point detection under the sampling constraint, we propose to use MAB to combine the CUSUM statistics into the reward function in the MAB problem. However, different from [40, 14], the "arm" does not refer to whether a single sensor should be observed or not, but the combinations of the number of testing resources (e.g., testing kits) should be distributed to each particular region. Finally, an integer programming algorithm is proposed to optimize such reward function for resource allocation.

The rest of the paper is organized into the following parts. In Section 2, we describe the motivation example of the problem in detail. In Section 3, we review some related methods in change point detection. In Section 4, we introduce the formulation and algorithm of the proposed method in detail. In Section 5, we use the simulation to study the behavior and some properties of the algorithm. A threshold is also generated to be applied to the case study. In Section 6, we study the case of county-level daily positive cases in Washington State (WA). Section 7 summarises the result and contributions of this study.

2. Motivation Example

Since the initial outbreak of the novel coronavirus in early January 2020, the COVID-19 pandemic has rapidly spread across the world and brought enormous disruption to the economy and society. Various emergency measures, such as social distancing, school closures, and economic shutdowns, have been taken by different countries to control the first wave of the pandemic. To balance saving lives and the economy, a condition-based phased approach has since been adopted in the USA, where the strictness of public health measures is set to adapt according to the present epidemic condition. The success of such a phased approach critically depends on the accurate assessment of the current and near-future status of the pandemic.

Much recent research has been focusing on modeling and prediction of the spread of the COVID-19 case report data. For example, Jiang et al. [17] studied the pandemic from a statistical point of view. The piece-wise linear trend model has been applied to model the positive case growth and the self-normalization-based method was applied

to detect the change points and predict cases in the future. Tariq et al. [30] monitored pandemic data in Singapore and estimated the effective reproduction number R_t . They also took advantage of different local clusters and some international transmission. Despite various studies to forecast the positive cases and deaths, Luo [22] pointed out that it would be hard to predict the future of the pandemic. The direction of the mutant of a virus is random and, thus, hard to predict whether it will be more contagious or not. For example, the Delta and Omicron variant of COVID-19 caused a remarkable increase of positive cases in the US starting in June 2021 and Jan 2022.

Despite the prediction and modeling of the COVID-19 spread, many recent works focus on developing a monitoring framework to detect the pandemic outbreak while taking advantage of a more scientific resource allocation method to distribute the test kits and detect the outbreak as soon as possible. ?] Astley et al. [2] used the data from The University of Maryland Global COVID Trends and Impact Survey (UMD-CTIS) in cooperation with Facebook to monitor the pandemic. In general, this algorithm can monitor the COVID-19-related daily spatio-temporal cases with limited testing resources, and in this paper, we call this problem the adaptive partially observed spatio-temporal hotspot detection problem. However, most of these works assume that the data is passively collected but do not focus on how to adaptively distribute the testing resources. Chatzimanolakis et al. [5] discussed the optimal resources allocation problem in a different stage to achieve maximum information gain.

To better understand the COVID-19 status, different types of testing resource is typically distributed in different regions. Centers for Disease Control and Prevention (CDC) has classified the testing for COVID-19 into the following two categories: 1) **Diagnostic testing** is intended to identify current infection in individuals and is performed when a person has signs or symptoms consistent with COVID-19, or is asymptomatic, but has recently known or suspected exposure to COVID-19. 2) **Screening tests** are recommended for unvaccinated people to identify those who are asymptomatic and do not have known, suspected, or reported exposure to COVID-19. Screening helps to identify unknown cases so that measures can be taken to prevent further transmission.[1]

Overall, the screening test is very useful in testing employees in a workplace setting or universities, testing a person before or after travel, or randomly distributed test in some underdeveloped areas to identify unknown cases so that measures can be taken to prevent further transmission. In this paper, we will focus on hotspot detection for the case report data with limited screening test resources. For simplicity, we assume that the screening test is the only available source of information, and limited testing resource is available.

In this paper, we will use the real COVID-19 test report data from Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE) (Dong et al. [9]). The dataset is available on <https://github.com/CSSEGISandData/COVID-19>. More specifically, we will use the confirmed COVID-19 cases from all 39 counties in WA. The source of daily positive cases in WA is the Department of Health (<https://www.doh.wa.gov/Emergencies/COVID19>). The time-series data is updated daily around 23:59 (UTC). We will use the data from Jan 23, 2020, to Sep 13, 2020, a total of 235 days, as an illustration. On each day, the confirmed cases are recorded in all counties in the United States. We aim to identify hotspots in such a region and see if we can discover the hotspot with a much smaller number of screening test resources.

In this study, we will focus on the distribution of the number of screening testing resources for efficient hotspot detection. With this technique, we are able to conduct a real-time online monitoring process and actively emphasize the regions potentially

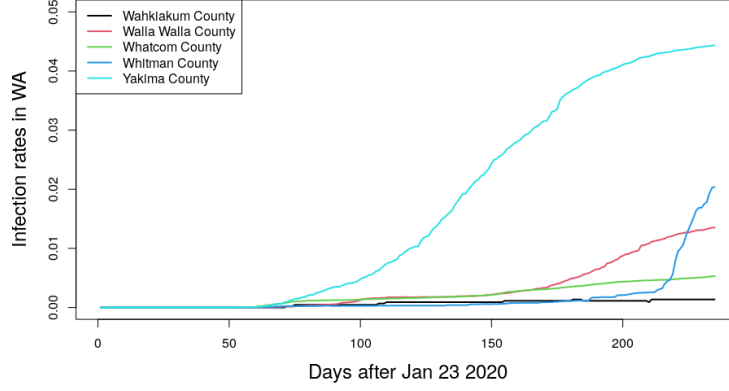
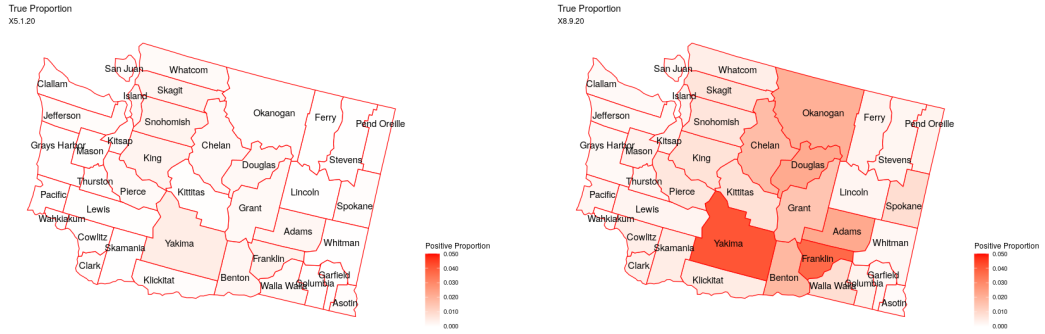


Figure 2.: Example of infection rates in WA

at risk. First, we showed the infection rates in some example counties in Figure 2. From this figure, we can see that the positive rates of Yakima County started to rise around day 70, and increased dramatically. Furthermore, we have added a figure of the proportion of infected cases in some of the time points during the spread of the COVID-19 data in Figure 3. We can see that the infection rates on day 100 are not very high for all counties. The infection rates in some of the counties started to rise.



(a) Infection proportion on day 100: May 1 2020 (b) Infection proportion on day 200: Aug 9 2020

Figure 3.: Comparison of county-wise infection rate in WA on different dates

3. Literature Review

In this Section, we will present a literature review on the change point detection methodology that can be potentially applied to the proposed spatio-temporal problem, including change point detection on the high dimensional data stream, change point detection with sampling control, and change-point detection on count data.

3.1. Change Point Detection for multiple data streams

Originally, the change point detection is first proposed to detect some sudden changes in the *univariate data stream* with the shortest detection delay. The CUSUM

statistics is proposed by Page [26, 27] to minimize the supremum average detection delay [25]. Then, the Shiryaev–Roberts procedure was introduced by Shiryaev [29] in a Bayesian decision framework. Please see [31] for a review paper of the existing change point detection methods. However, as mentioned, these methods cannot be applied to problems with multiple data streams.

To extend the change-point detection method for *multiple data streams*, Mei [23] extended the CUSUM statistics to multiple data streams by the summation of the CUSUM statistics for each data stream. Xie and Siegmund [32] compared the mixed procedure method they developed with some generalized log-likelihood ratio-based procedures to detect the change point in multiple but unknown data streams under the assumption that the data are normally distributed with known variance, and the change is a mean shift with unknown post-change mean. There are some works to extend these methods to multiple change-point detections with abrupt changes, such as [39, 4], which considered the functional subspace learning and exhaustively search of the change point. Enikeeva and Harchaoui [12] studied the change point detection when the number of dimensions and the length of each data stream goes to infinity based on the maximum of scan statistic and linear test statistic. However, these methods typically assume that the data follow a normal distribution or the data is fully observed.

3.2. Change Point Detection for Partially Observed Data with Sampling Control

Another line of research focuses on detecting anomalies for the *spatio-temporal data*. For example, Das et al. [8] applied some tests to study the change point of winter maximum, minimum, and average temperature from 6 meteorological stations in 102 year time period. Chen et al. [6] proposed a new S^3T statistic to detect change points in spatio-temporal data with mean shift or covariance structure change. Zhao et al. [41] developed a likelihood-based algorithm to solve the multiple change detection in non-stationary spatio-temporal processes by splitting the data into piecewise stationary processes. Knoblauch and Damoulas [19] developed a Bayesian online change-point detection algorithm to model the process between change points in spatio-temporal data. Zhang et al. [38] studied the change point detection problem when there are system-inherent variations and sensor feature differences and is able to identify the local change point in each cluster. Again, these methods focus on the fully observed data and do not provide a method to iteratively update the sampling resources.

To generalize the change point detection for *partially observed data*, some works focus on the change point detection by assuming that only partial data streams at each time point can be observed due to limited sampling resources. Such setting has been studied in social network monitoring [13], biological network [28], and manufacturing systems [20]. To deal with partially observed data, Xie et al. [33] developed a change point detection method by combining subspace manifold learning and multiscale analysis to solve the missing data problem. Dubey et al. [10] studied the online change point detection problem in the network where the pattern of the missing data of the network is heterogeneous. Corradin et al. [7] studied multiple change-point detections in multivariate time series with missing values. The conditional distribution of each data point given the rest data is derived to handle the missing value problem. However, these methods do not actively control which subsets of data to measure, which hinders the detection performance.

Recently, many existing works have focused on the change-point detection problem with sampling control, where the sensor(s) to be observed in each epoch are determined adaptively based on the current and historical observation. For example, Liu et al. [20] proposed to combine CUSUM statistics and a top- r strategy to select the data streams to be observed for online change detection. Xu and Mei [34] studied the problem when there is only one sensor to be observed at each time and an adaptive sampling strategy based on the CUSUM statistic and showed that it is asymptotically optimal in the sense of minimizing the detection delay. Zhang and Hoi [37] developed an online learning algorithm for anomaly detection for independent data streams using the combinatorial MAB approach. Zhang and Mei [40] studied the change point detection under sampling control by the combination of Thompson Sampling and a top- r local Shiryaev-Robert statistic to raise a global alarm. Guo et al. [14] proposed a Bayesian spike-and-slab prior distribution to decompose the data into the smooth background and sparse anomaly and focus on identifying the anomaly by selecting the sensors to observe using Thompson sampling. However, all aforementioned methods focus on change detection under the continuous observation data and assume each sensor is either fully observed or not, which cannot be used for more complex resource allocation problems for anomaly detection.

3.3. Change Point Detection on Counting Data

Much existing research has been focusing on extending the change point detection method for *count data or binary data*. For example, Hinkley and Hinkley [16] studied the change point detection problem for binary variables based on the asymptotic distribution of the likelihood ratio test statistics. Yu et al. [36] developed the CUSUM statistics to study the change point detection problem in the binomial thinning process and applied it to the 2001-2012 influenza data. Jiang et al. [18] proposed a weighted CUSUM control chart for monitoring Poisson processes with varying sample sizes. Gut and Steinebach [15] studied the change point detection problem in renewal counting data with the discrete-time frame, where the term ‘partially observed’ meant the discrete-time frame. Ellenberger et al. [11] developed a binary segmentation method to solve the change point detection problem when the sample size of the binomial distribution is small. However, these methods typically assume that the sample size of the binomial distribution is pre-determined, which cannot be used to solve the resource allocation problem.

4. Proposed Methodology

In this Section, we will explore the adaptive sampling method on discrete data with the binomial distribution. In Section 4.1, we will review the overall problem formulation. In Section 4.2, we will introduce the procedure of the proposed algorithm. In Section 4.3, we will introduce the tuning parameter selection procedure for the proposed algorithm.

4.1. Problem Formulation

We assume that the data is $X_{k,t}$, where $t = 1, 2, \dots$ is the time index and $k = 1, 2, \dots, K$ is the region index. In the COVID-19 monitoring example, $X_{k,t}$ can be the

number of newly observed confirmed cases in region k and time t .

We further assume that all $X_{k,t}$ prior-change follows a binomial distribution f_k^0 as $\text{Bin}(c_{k,t}, p)$, with the number of test $c_{k,t}$ and the probability p . Furthermore, we assume that after the change $t > t_0$, only one hotspot region (i.e., $k \in \mathcal{H}$) is impacted by the change. In contrast, we assume that the regions affected by the changes follow another post-change distribution f_k^1 where f_k^1 is another binomial distribution as $\text{Bin}(c_{k,t}, q)$, with the number of tests as $c_{k,t}$ with the probability q , as shown in Eq. (1). The superscript 0 means in control, and the superscript 1 means out of control.

$$X_{k,t} \sim \begin{cases} \text{Bin}(c_{k,t}, p), & t \leq t_0 \text{ or } t > t_0, k \notin \mathcal{H} \\ \text{Bin}(c_{k,t}, q), & t > t_0, k \in \mathcal{H} \end{cases} \quad (1)$$

Finally, we assume that there are limited sampling resources. The constraint we assume is that limited testing kits C are distributed to all the regions at each time t with the constraint $\sum_k c_{k,t} = C$. For simplicity, we assume that the actively distributed test kits are the only source of observation data. The objective of this paper is to sequentially distribute the testing resources $c_{k,t}$ for region k at time t for faster change point detection.

4.2. Proposed Adaptive Resources Allocation CUSUM for Binomial Count data (ARA-CUSUM) Method

In this Section, we first introduce the overall step of the algorithm and then discuss three major components of the proposed methods as follows: monitoring statistics updates, planning for resource allocation, and change-point detection decision. The overall framework is shown in Figure 4. The detailed steps are discussed as follows.

- (1) **Monitoring statistics update:** We propose to update the monitoring statistics at each time t for the quickest change-point detection. This procedure will be discussed in detail in Section 4.2.1.
- (2) **Planning for adaptive sampling:** We would like to distribute the testing resource for the next time point based on previous results by the UCB bandit algorithm. The detailed planning formulation and the optimization algorithm are discussed in Section 4.2.2 and Section 4.2.4.
- (3) **Change point detection decision:** Finally, the alarm is raised when the largest CUSUM statistics for all regions exceeds some threshold. Finally, the corresponding region that triggers such change can also be used for hotspot identification.

4.2.1. Monitoring Procedure

In this Section, we will first introduce the definition of CUSUM statistics and then derive the monitoring procedure for online change detection.

Definition 4.1. Suppose in a sequential data, x_n is the score of n th observation. The CUSUM statistics Page [26] is defined by

$$S_0 = 0, \quad S_n = \max\{0, S_{n-1} + x_n\}, (n \geq 1) \quad (2)$$

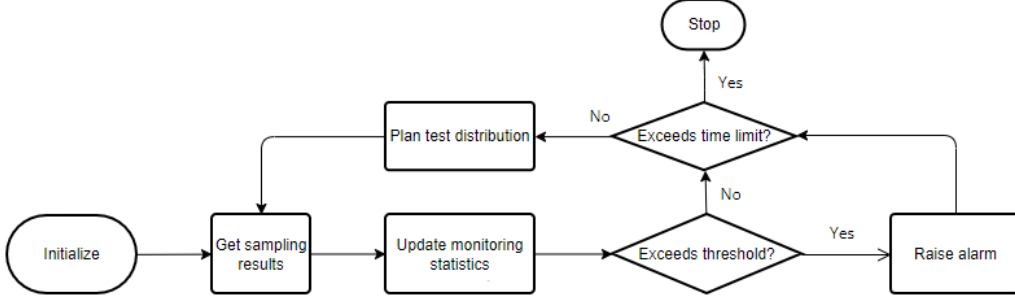


Figure 4.: Flow chart for the algorithm

Moustakides [24] suggested that the CUSUM statistic is mini-max optimal in the sense of detection delay [21].

$$D(T) = \sup_{\nu \geq 0} \text{ess sup } \mathbb{E}_{\nu}[(T - \nu)^+ | X_1, \dots, X_{\nu}] \quad (3)$$

with the constraint of in-control average run length $\mathbb{E}_{\infty}[T] \geq \gamma > 0$. In change point detection, the score x_n for each observation can be the log-likelihood ratio, which is $x_n = \log \frac{f^1(X_n)}{f^0(X_n)}$, where X_n is the n th observed data, f^0 is the pdf of X_n before change and f^1 is the pdf of X_n after change.

In the proposed formulation, $X_{k,t}$ is defined as the observed positive cases in time t and region k . Inspired by the CUSUM statistics definition, the CUSUM statistics W_t^k is formulated as:

$$W_{k,t} = \max\{W_{k,t-1}, 0\} + \log \frac{f_k^1(X_{k,t})}{f_k^0(X_{k,t})}, \quad (4)$$

where $f_k^0 \sim \text{Bin}(c_{k,t}, p)$ and $f_k^1 \sim \text{Bin}(c_{k,t}, q)$ represent the distribution of null and alternative hypothesis, respectively. The initial value $W_{k,0}$ is set to be 0 for all k . Following the CUSUM statistics definition, we define the CUSUM statistics $W_{k,t}$ in the proposed formulation for each county k and time t in Proposition 4.2.

Proposition 4.2. *Suppose that at time t , the number of tests distributed at region k is $c_{k,t}$, the number of positive cases collected is $X_{k,t}$. For each region k , for $k = 1, 2, \dots, K$, the CUSUM statistics $W_{k,t}$ can be computed for each time t and region k can be derived as*

$$W_{k,t} = \max\{W_{k,t-1}, 0\} + c_{k,t} \log \frac{1-q}{1-p} + X_{k,t} \log \frac{q(1-p)}{p(1-q)} \quad (5)$$

The proof is detailed in Appendix A.

Here, we will raise the alarm when the largest of the CUSUM statistics $\max_k W_{k,T}$ exceeds some threshold c at some time T . Finally, we will discuss the procedure of selecting the threshold c in Section 4.3.

4.2.2. Planning Objective Function for Resource Allocation

In this Section, we present an efficient algorithm for resource allocation at the next time point for hotspot detection. Overall, a good resource allocation algorithm for change detection should achieve a good balance between exploration and exploitation. Here, exploration in the problem formulation implies the test kits should be distributed to the regions that were not measured before, and exploitation means that when the change occurs, the planning will focus more on the region with the change for the quickest change detection. To better tackle the exploration-exploitation trade-off problem, we apply the UCB as the target function to be optimized. Auer [3] suggested that the UCB criteria for the MAB problem maximize the final reward. This method takes advantage not only of the mean reward μ but also the standard deviation of the reward σ . In this case, we should not only focus on the region leading to the larger reward but also the action with larger variability, which has the potential to lead to a larger reward. Specifically, in this hotspot detection application, for the true hotspot region with a larger infection rate, we may want to continue to distribute more test kits to the particular region the next day to encourage further exploitation. However, when a region has not been observed enough in previous days, it typically has a larger variance, and some testing resources will be distributed in the regions with large variability the next time to encourage further exploration.

Given that the log-likelihood ratio $\log \frac{f_k^1(X_{k,t})}{f_k^0(X_{k,t})}$ for change point detection is often a good quantification of how likely the change may happen in a particular region k , here we propose to use the sum of UCB of the likelihood ratio in the CUSUM statistics as the reward function. For the convenience of notation, we denote the log-likelihood ratio in the CUSUM statistics in each region k by

$$\Delta r_{k,t} = c_{k,t} \log \frac{1-q}{1-p} + X_{k,t} \log \frac{q(1-p)}{p(1-q)} \quad (6)$$

Therefore, the UCB function of the summation of the likelihood ratio statistics can be expressed as

$$f(\mathbf{c}_t) = \sum_{k=1}^K \left[E[\Delta r_{k,t}] + \sqrt{\text{Var}(\Delta r_{k,t})} \right] \quad (7)$$

Here, the goal is to find the best resource allocation planning method to optimize the resource distribution $\mathbf{c}_t = (c_{1,t}, \dots, c_{K,t})$ at time t . However, one of the challenges is that the $X_{k,t}$ is the number of positive testing results at time t , which is a random variable at time t . In Section 4.2.3, we will derive the Bayesian updating procedure to update the posterior distribution of $X_{k,t}$, which can be used estimate the $E[\Delta r_{k,t}]$ and $\text{Var}(\Delta r_{k,t})$.

4.2.3. Bayesian Updating of $X_{k,t}$

To estimate the distribution of $X_{k,t}$, we assume that it follows a binomial distribution of $\text{Bin}(c_{k,t}, p_k)$, where p_k is the true infection rate to be estimated. Therefore, an online estimation of the rate is needed for better planning and resource allocation. Here, we propose a weighted Bayesian update method to estimate the infection rate in Proposition 4.3. The reason to consider such a weighted updating procedure is that

the samples in recent time points are typically more likely to represent the current state of the system and should be used to better detect the change. Therefore, more weight should be put into the recent time points. For simplicity, we propose to use an exponential decayed weight $\lambda_t = w^{T-t}$ for each sample $X_{k,t}$.

Finally, the posterior distribution $p_{k,t}$ can be derived as

$$P(p_k|X_{k,1}, \dots, X_{k,t}) \propto P(p_k) \prod_{t=1}^T P(X_{k,t}|p_k)^{\lambda_t}, \quad (8)$$

where $P(p_k)$ is the prior distribution of the infection rate, which is assumed to follow a Beta distribution as $\text{Beta}(a, b)$. $P(p_k|X_{k,1}, \dots, X_{k,t})$ is the posterior distribution of p_k , which can be derived in Proposition 4.3.

Proposition 4.3. *Suppose that the data $(c_{k,t}, X_{k,t})$ of the k^{th} region until time T is collected, where $c_{k,t}$ is the number of total tests distributed to region k and time t and $X_{k,t}$ is the daily positive cases. We further assume the prior distribution of p_k is $\text{Beta}(a, b)$, the weighted posterior distribution of $P(p_k|X_{k,1}, \dots, X_{k,T})$ can be derived as*

$$\text{Beta}(a + \sum_{t=1}^T X_{k,t} w^{T-t}, b + \sum_{t=1}^T (c_{k,t} - X_{k,t}) w^{T-t}) \triangleq \text{Beta}(\alpha_{k,t}, \beta_{k,t}) \quad (9)$$

The proof is given in Appendix B. Here, the updated posterior distribution of $p_k|X_{k,1}, \dots, X_{k,T}$ can be used to calculate the uncertainty in $X_{k,t}$ to derive the UCB function in the planning method.

4.2.4. Optimization Algorithm for Planning

By plugging in the weighted posterior distribution of $P(p_k|X_{k,1}, \dots, X_{k,T})$ into the UCB reward function in (7), we will derive the objective function to be optimized and propose an integer programming algorithm to optimize the number of test kits $c_{k,T+1}$ in each region k at time $T + 1$.

Proposition 4.4. *Suppose $X_{k,t} \sim \text{Bin}(c_{k,t}, p_k)$ and the posterior distribution $p_k|X_{k,1}, \dots, X_{k,T} \sim \text{Beta}(\alpha_{k,t}, \beta_{k,t})$. The target function $f_t(\mathbf{c}_{T+1}) = \sum_{k=1}^K [E[\Delta r_{k,T+1}] + \sqrt{\text{Var}(\Delta r_{k,T+1})}]$ can be derived as*

$$\begin{aligned} f_t(\mathbf{c}_{T+1}) &= \sum_{k=1}^K f_{k,t}(c_{k,T+1}) \\ &= \sum_{k=1}^K \left[\frac{\alpha_{k,t}}{\alpha_{k,t} + \beta_{k,t}} c_{k,T+1} + \sqrt{c_{k,T+1} \frac{\alpha_{k,t} \beta_{k,t}}{(\alpha_{k,t} + \beta_{k,t})(\alpha_{k,t} + \beta_{k,t} + 1)} \left(\frac{c_{k,T+1}}{\alpha_{k,t} + \beta_{k,t}} + 1 \right)} \right] \end{aligned} \quad (10)$$

The proof is shown in Appendix C. Therefore, Algorithm 1 is proposed to determine the best resource allocation $(c_{1,T+1}, \dots, c_{K,T+1})$ at time $T + 1$ as follows:

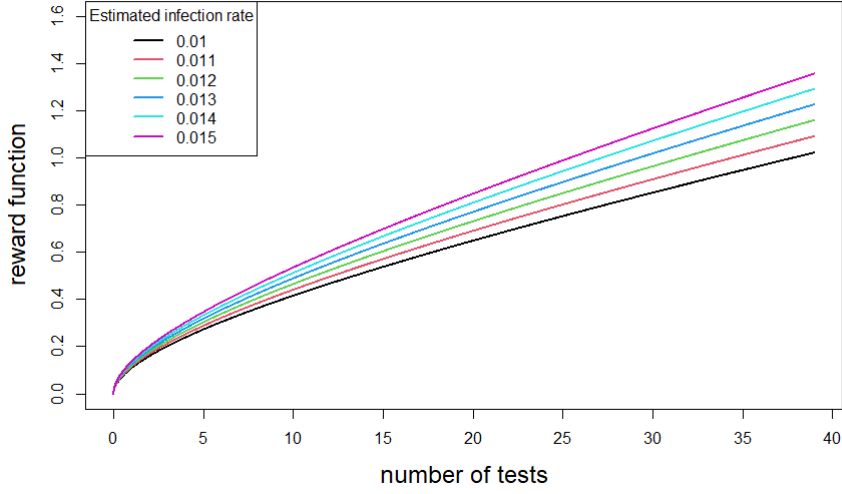


Figure 5.: Example of reward functions with different estimated infection rate

$$\begin{aligned} \hat{c}_{1,T+1}, \dots, \hat{c}_{K,T+1} = \arg \max f_t(\mathbf{c}_{T+1}) \\ s. t. \sum_{k=1}^K c_{k,T+1} \leq C, c_{k,T+1} \geq 0, k = 1, \dots, K \end{aligned} \quad (11)$$

Here, we would like to give a brief discussion on the reward function $f_{k,t}(c_k)$ defined in (10). An example of the function $f_{k,t}(c_k)$ with different infection rates are given in Figure 5. The expectation part and standard deviation part is larger for regions with a higher estimated infection rate. However, there is a diminishing return effect on this reward function, which implies that as more testing resources are distributed in the highly infected regions, the increase of the reward function will become smaller. At a certain point, it may be less rewarding to distribute additional tests to the highly infected region compared to the low infected region. To demonstrate, Figure 5 shows that $f_5(x) > f_k(x)$ for $k = 1, 2, 3, 4$ and $x \in [1, 50]$, but $f'_k(x) > f'_5(y)$ for some $k = 1, 2, 3, 4$ and $x, y \in [1, 50]$.

To optimize problem (11), a greedy algorithm is proposed. The idea of the greedy algorithm is to distribute the testing kits one by one to the region that can result in the largest target increment until all the tests are properly distributed. Given the concavity of $f_t(\mathbf{c})$, such a greedy approach can guarantee the global optimum. This property will be discussed in Section 4.6. The greedy algorithm is designed as Algorithm 1.

Finally, the entire process can be shown in Algorithm 2. Overall, the steps of the algorithm are described here: 1) On day one, we have an initial test distribution. If the prior distribution is set the same for all regions, uniform distribution can be used. The confirmed number of $X_{k,1}$ will be collected on Day 1. 2) The collected dataset $X_{k,t}$ can be used to update the posterior distribution of the rate p_k by (9). 3) Such distribution information can be plugged into (11) to optimize the test distribution of the next day. Such a procedure can be estimated recursively until the current time T .

Algorithm 1: Optimal Resources Planning by the Greedy Algorithm

Parameters: $\alpha_{k,t}, \beta_{k,t}, k = 1, \dots, K$

Result: $c_k, k = 1, \dots, K$

Step 1: Initialize $c_k = 0$ for $k = 1, \dots, K$

Step 2: for $i = 1$ to C do

Step 2.1: for $k = 1$ to K do

 | Estimate the increment in region k : $\Delta_k = f_k(c_k + 1) - f_k(c_k)$.

 end

Step 2.2: Let $k_m = \arg \max_k \Delta_k$. $c_{k_m} = c_{k_m} + 1$

end

Algorithm 2: Proposed ARA-CUSUM Algorithm

Parameters: a, b, w, c

Result: $W_{k,t}, k = 1, \dots, K, t = 1, 2, \dots$

Step 1: Initialize $c_{k,1}, X_{k,1}$ and $W_{k,1}$ for $k = 1, \dots, K$;

Step 2: for $t = 1, 2, \dots$ do

Step 2.1: Bayesian Update: Set $\alpha_{k,t} = a + \sum_{i=1}^t X_{k,i} w^{t-i}$;

$\beta_{k,t} = b + \sum_{i=1}^t (c_{k,i} - X_{k,i}) w^{t-i}$;

Step 2.2: Resource Allocation: Solve $c_{k,t+1}$ in the optimization problem (11) by Algorithm 1 ;

Step 2.3: Monitoring Statistics Update and Detection: Update $W_{k,t+1}$ by (5) ;

Step 2.4: if $W_{k,t+1} > c$ for some k then

 | Raise alarm at time $t + 1$ and county k ;

end

4.2.5. Property of the Optimization Algorithm

In this subsection, we would like to prove that the greedy approach proposed in Section 4.2.4 can obtain the global optimum.

Lemma 4.5. *The target function (10) is the sum of K concave functions and is therefore concave.*

The target function in each region can be expressed as $f_{k,t}(x) = ax + \sqrt{bx + cx^2}$. One can easily verify that the second order derivative of $f_{k,t}(\cdot)$ is negative when $b, c > 0$.

Proposition 4.6. *The algorithm 1 can obtain the global optimum for the optimization problem (11).*

Proof. From Lemma 4.5, $f_{k,t}(\cdot)$ is an increasing and strictly concave function. For any fixed time t , we denote $g_k(x) = f_{k,t}(x) - f_{k,t}(x - 1)$ be the increment of the reward function to distribute one more tests on region k . $g_k(\cdot)$ will be a positive decreasing function for any k . Then the total reward function given the test distribution $c_{1,t}, \dots, c_{K,t}$ is

$$f_t(c_{1,t}, \dots, c_{K,t}) = \sum_{k=1}^K \left(\sum_{x_k=1}^{c_{k,t}} g_k(x_k) \right)$$

Now consider the increment parameter matrix

$$\begin{pmatrix} g_1(1) & g_1(2) & \cdots & g_1(C) \\ g_2(1) & g_2(2) & \cdots & g_2(C) \\ \vdots & \vdots & \ddots & \vdots \\ g_K(1) & g_K(2) & \cdots & g_K(C) \end{pmatrix}$$

In Algorithm 1, the first iteration will choose the largest value in the first column. Since each row in the matrix is decreasing, the first iteration will also choose the largest value in the matrix. Each of the remaining iterations will choose the largest value in the matrix that has not been chosen. Therefore, Algorithm 1 chooses the top C largest values in the matrix. And because the target function can be written by the sum of C values in the matrix, Algorithm 1 gives the optimal solution to the optimization problem (11)

□

4.3. Tuning Parameter Selection

There are several tuning parameters that need to be determined for the algorithm. The key parameters are prior distribution parameter a, b , weight w and threshold c . In this Section, we will discuss the effect of these tuning parameters and how to decide them.

- **Prior Distribution:** There are two parameters a, b in the prior distribution. The prior distribution reflects the prior knowledge about the infection rate. Given that the infection rate for the null hypothesis is p , we can also set the ratio to be p , i.e., $\frac{a}{a+b} = p$, assuming no change happens. Since p is a preset constant, we only need to determine a . When a is large, the influence of prior distribution is large, and the difference in the infection rate estimation from the posterior distribution will be smaller. Thus, the difference in test distribution will not be very large. In the simulation, we will choose $a + b = 0.5 \times C$.
- **Weight w :** The weight is used in Bayesian update to get the posterior distribution of $X_{k,t}$. Weight allows the algorithm to take not only the information from one day before but also the previous days. A large weight will make the test distribution in each region smooth across different days. Therefore, when occasionally the observed infection rate is high, the algorithm will only distribute a reasonably larger amount of testing kits on the next day rather than some extreme testing kit distribution. In the simulation, we choose $w = 0.3$.
- **Total Resources C :** This parameter determines the total number of tests to be distributed in all regions. Typically, C is pre-determined by the resource constraint.
- **Threshold c :** This parameter determines whether an alarm should be raised in each region every day. Higher threshold c leads to high in-control average run length (ARL_0) and higher out-of-control ARL_1 . Typically, we would like to maintain a preset ARL_0 , which determines the threshold c . Overall, for a preset C, p, q , a simulation study can be used to determine the best C with certain p and q . We will determine this parameter by the simulation work and use a preset average run length.

5. Simulation

In this Section, we will start with the simulation setup in Section 5.1 and then discuss the result evaluation in Section 5.2. The balance of exploration and exploitation is discussed in Section 5.3.

5.1. Simulation Setup

In this Section, we will evaluate the proposed methodology using a simulation study. To generate the simulation dataset, we will simulate a true infection rate for each county in the WA as an example.

There are 39 counties in WA, i.e., $K = 39$. Before the distribution change, the true infection proportion is set to be $p = 0.01$ for all counties. We will compare the detection delay for different infection proportions where the out-of-control infection rates are $q = 0.025, 0.03, 0.04, 0.05$ in the first county after the distribution change. We further assume that the total number of tests is 3900, where each county takes an average of 100 tests per day. The weighted update parameter w is set to be 0.3 and the prior distribution parameter a, b are set as the 50% of the total tests to make the prior distribution as $a = 3900 \times 50\% \times 0.01 = 19.5$, $b = 3900 \times 50\% \times 0.99 = 1930.5$.

The process of the simulation is described as follows:

- (1) The test distribution $c_{k,t}$ in each county determined from the previous iteration can be used.
- (2) Before the change point $t < t_0$, we will sample the observed positive cases by $X_{k,t} \sim \text{Bin}(c_{k,t}, p)$.
- (3) After the change point $t > t_0$, we will sample the observed positive cases in the first county by the out-of-control infection rate q and $X_{1,t} \sim \text{Bin}(c_{1,t}, q)$. We will sample the rest counties still by $X_{k,t} \sim \text{Bin}(c_{k,t}, p)$

Finally, we will follow Algorithm 1 for the proposed algorithm. The CUSUM statistics are updated, and the test distribution of the next day is determined by the proposed method or benchmark methods. When studying the average run length and detection delay, we will stop the process when the alarm is raised.

In this simulation study, we will compare the proposed algorithm with the following two benchmark methods.

- (1) **Even Distribution:** The first benchmark (denoted as Even) is a simple approach, which always evenly distributes the tests to all the counties. This even distribution will focus on exploration of the testing resources but do not have the power to exploit for a specific region.
- (2) **Top-R Distribution:** The second benchmark (denoted as top-R') is inspired by the Top-R strategy proposed in [20]. We first evenly distribute all tests evenly into 20 batches. For in total 3900 testing kits, it averages out 195 testing kits per batch. We then select the top 20 regions with the largest CUSUM statistics in the previous day. Here, we select the top 20 regions, given it is a little more than half of the whole regions.

5.2. Evaluation of the Detection Delay and Precision

To compare this algorithm with the benchmarks, we will first need to determine the thresholds for the proposed method and benchmark methods. Here, binary search

Table 1.: The result of detection delay and detection precision when $ARL_0 \approx 200$

Metric	Methods	$q = 0.025$	$q = 0.03$	$q = 0.04$	$q = 0.05$
ARL ₁	Proposed	7.893	4.958	3.388	2.863
	Even	14.885	7.93	4.408	3.299
	Top-R	10.385	5.688	3.443	2.891
DP	Proposed	0.918	0.932	0.938	0.939
	Even	0.892	0.926	0.939	0.943
	Top-R	0.904	0.918	0.936	0.937
SDRL	Proposed	4.67	2.47	1.31	1.01
	Even	12.81	5.48	2.28	1.45
	Top-R	10.556	3.72	1.63	1.12

can be used to decide the optimal threshold for each benchmark method to maintain the in-control average run length (ARL) as 200 for all benchmark methods with 1000 replications.

To evaluate the proposed methods, we use the following three metrics: 1) the out-of-control average run length (ARL₁): ARL₁ is defined as the average detection delay after the change has happened. Overall, we aim to reduce the ARL when keeping the same $ARL_0 = 200$; 2) Detection precision (DP): DP is evaluated as the proportion in the 1000 iterations that the alarm is raised in the correct county. Overall, we would aim to increase the DP so that we can identify the county that is accountable for the change more accurately. 3) the standard deviation of run-length (SDRL): SDRL is evaluated by computing the standard deviation with 1000 iterations. Overall, a smaller SDRL would imply a more robust detection with smaller variability for detection delay. These three metrics for the proposed method and the benchmark methods (namely the evenly distributed and the Top-R distributed) are evaluated in Table 1.

From Table 1, we can conclude that ARL is the smallest for the different q values. Especially, when q is small (i.e., 0.025 and 0.03), the detection delay can even be greatly reduced (i.e., 7.893 for "proposed", 14.885 for "Even", and 10.385 for "Top-R" for $q = 0.025$). For larger q , the differences between the proposed methods and the second best method, such as "Top-R" are similar, given it becomes easy to detect all the methods. However, the proposed method still outperforms others in this case. Finally, we find that the SDRL of the proposed method is also the smallest, especially when q is small. This implies that the proposed methods can also make the detection much more robust by reducing the standard deviation of the detection delay.

5.3. Exploration and Exploitation

As formerly mentioned, a good change point detection with a sampling control algorithm should balance both exploration and exploitation behavior. In this simulation, exploration means that we will need to explore all counties, given that any county can undergo large changes at certain times. Exploitation means that we will need to distribute more resources to the counties that have been detected or have the potential to be the hotspot. Figure 6 and Figure 7 give a more intuitive understanding of the exploration and exploitation behavior of this proposed algorithm from the following simulation study setup. In the first 500 days, all the counties are in control with the same infection rate $p = 0.01$. From day 501 to day 1000, only the first county, Adams

County is out of control with the infection rate $q = 0.05$, while the rest counties stay the same with $p = 0.01$. Figure 6 shows the density and the boxplot of the number of tests in control. Figure 7 shows the median number of test kits in each county under different conditions, in control and out of control.

Here are some behaviors of the algorithm. When there is no distribution change, the distributions of the number of tests in each county are similar, see Figure 6 and Figure 7a. This shows that the proposed method is able to explore all the counties undergoing potential changes. However, if a county has a higher infection rate on the current day, it is likely to have higher tests the next day. If the sampled positive proportion is small, the increment of CUSUM statistics will be likely to be 0. If the distribution change happens in that county, the CUSUM test statistic will be likely to grow dramatically and we will continue to focus on that county further. See Figure 7b, when Adams County is out of control, the median number of the distributed tests of Adams County is significantly larger than the number of other counties. In this case, the CUSUM test statistic of that Adams County is more likely to increase dramatically, which is a good behavior of exploitation.

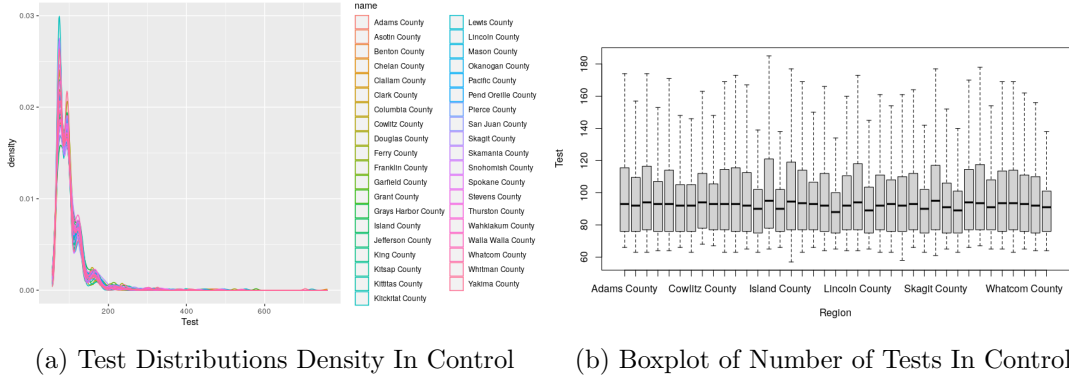


Figure 6.: Visualization of Test Distributions In Control

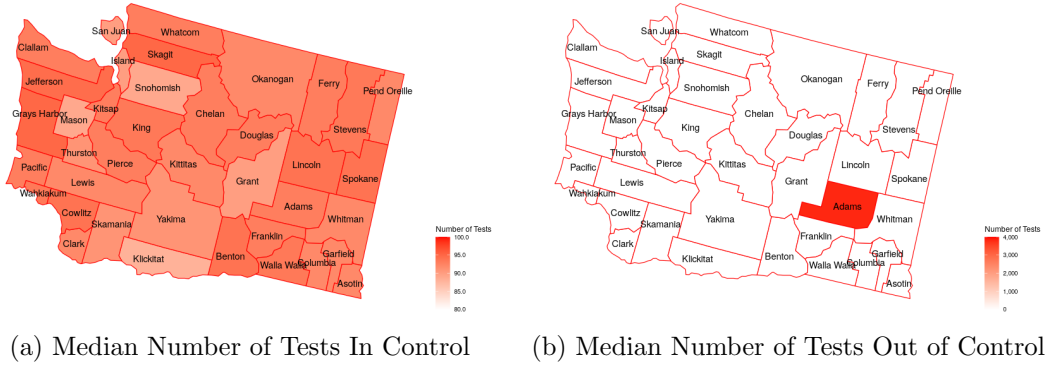


Figure 7.: Median Number of Tests

6. Case Study

In this case study, we would like to evaluate the proposed methods to the real case study. The data we are working on is the county-level daily positive cases, and the parameters are $K = 39$, $p = 0.01$, $q = 0.05$, $a = 19.5$, $b = 1930.5$ and $w = 0.3$. The average number of tests in each county is 100.

6.1. Detection of the Out-of-control County

The alarms were raised in Yakima county on Jun 19, 2020. The comparison figure of the test statistics is in Figure 8. Starting from day 135, the infection rate of Yakima county started to show a dominating trend over the other counties. Therefore, the algorithm emphasizes Yakima county to seek potential risk. But the true infection rate is not large enough to produce a positive increment in the CUSUM statistics. Thus, we will see this oscillating trend. When the infection rate of Yakima county increases consistently, the CUSUM statistics will increase dramatically. Finally, on June 21, 2021, the CUSUM statistics in Yakima County exceeded the threshold of 6.5, and the alarm was raised.

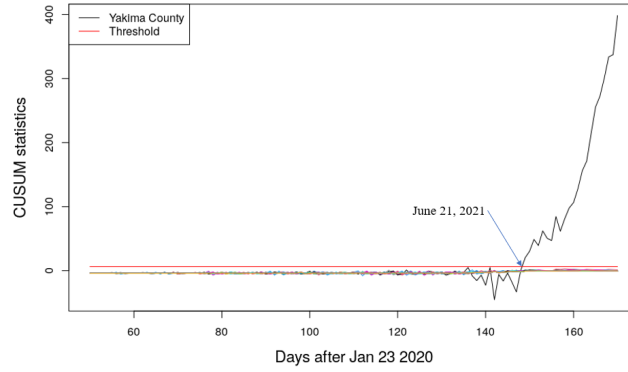
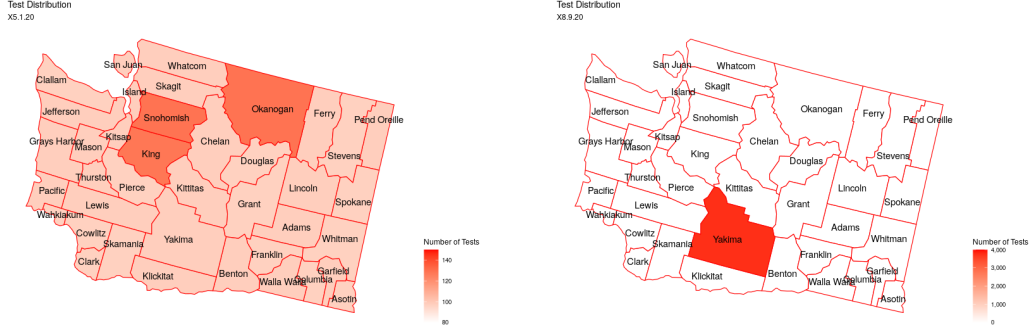


Figure 8.: Comparison of the Test Statistics

6.2. Behavior of Sampling Resource Distribution

The plots in Figure 9 are examples of the test distribution of this case study before and after the alarm is raised. This shows similar behavior as the simulation study. The test distribution before the alarm is close to an even distribution to explore all the counties for the risks. The test distribution after the alarm focuses on Yakima county and consistently monitors this county to show good behavior of exploitation.



(a) Example of test distribution in control (b) Example of test distribution out of control

Figure 9.: Example of test distribution in WA

7. Conclusion

In conclusion, the proposed method is able to handle the challenge of exploration and exploitation trade-offs and limited resources. In the simulation work, we show that the adaptive sampling method can beat the two benchmarks either in the sense of detection delay or detection precision. In the case study in WA, the alarm was raised in Yakima county. This algorithm works well when there is only one out-of-control region. When there are multiple out-of-control regions, this algorithm will focus on the first region with a higher out-of-control risk. If there are multiple potential regions at risk, one can start the algorithm from the beginning again when an alarm is raised and remove the alerted region. In the future, we can address this concern and detect multiple out-of-control regions in a time series manner. Some other future work includes taking geometric distance and graphing into consideration and solving the detection problem when there are multiple out-of-control regions.

8. Funding Statement

H. Yan was supported in part by NSF grant DMS-1830363. Y. Mei was supported in part by NSF grant DMS-2015405. S. E. Holte’s research was supported in part by NSF grant DMS [grant number DMS-1830372] and an NIH grant [AI027757]. In addition, this research was supported in part by an NIH grant [R21 AI157618]. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.”

9. Acknowledgments

The authors are grateful to the Associate Editor and the reviewers for their constructive comments that greatly improved the quality and presentation of this article.

10. Disclosure statement

No potential conflict of interest was reported by the author(s).

References

- [1] Overview of testing for sars-cov-2 (covid-19), 2022. URL <https://www.cdc.gov/coronavirus/2019-ncov/hcp/testing-overview.html>.
- [2] Christina M Astley, Gaurav Tuli, Kimberly A Mc Cord, Emily L Cohn, Benjamin Rader, Tanner J Varrelman, Samantha L Chiu, Xiaoyi Deng, Kathleen Stewart, Tamer H Farag, et al. Global monitoring of the impact of the covid-19 pandemic through online surveys sampled from the facebook user base. *Proceedings of the National Academy of Sciences*, 118(51), 2021.
- [3] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.
- [4] Valeriy Avanesov and Nazar Buzun. Change-point detection in high-dimensional covariance structure. *Electronic Journal of Statistics*, 12(2):3254–3294, 2018.
- [5] M Chatzimanolakis, P Weber, G Arampatzis, D Wälchli, I Kičić, P Karnakov, C Papadimitriou, and P Koumoutsakos. Optimal allocation of limited test resources for the quantification of covid-19 infections. *Swiss Medical Weekly*, 150:W20445, 2020. .
- [6] Junzhuo Chen, Seong-Hee Kim, and Yao Xie. S 3 t: A score statistic for spatiotemporal change point detection. *Sequential Analysis*, 39(4):563–592, 2020.
- [7] Riccardo Corradin, Luca Danese, and Andrea Ongaro. Bayesian nonparametric change point detection for multivariate time series with missing observations. *International Journal of Approximate Reasoning*, 2022.
- [8] Jayanta Das, Tapash Mandal, and Piu Saha. Spatio-temporal trend and change point detection of winter temperature of north bengal, india. *Spatial Information Research*, 27(4):411–424, 2019.
- [9] Ensheng Dong, Hongru Du, and Lauren Gardner. An interactive web-based dashboard to track covid-19 in real time. *The Lancet infectious diseases*, 20(5):533–534, 2020.
- [10] Paromita Dubey, Haotian Xu, and Yi Yu. Online network change point detection with missing values. *arXiv preprint arXiv:2110.06450*, 2021.
- [11] David Ellenberger, Berthold Lausen, and Tim Friede. Exact change point detection with improved power in small-sample binomial sequences. *Biometrical Journal*, 63(3):558–574, 2021.
- [12] Farida Enikeeva and Zaid Harchaoui. High-dimensional change-point detection under sparse alternatives. *The Annals of Statistics*, 47(4):2051–2079, 2019.
- [13] Mehrdad Farajtabar, Manuel Gomez Rodriguez, Mohammad Zamani, Nan Du, Hongyuan Zha, and Le Song. Back to the past: Source identification in diffusion networks from partially observed cascades. In *Artificial Intelligence and Statistics*, pages 232–240. PMLR, 2015.
- [14] Jie Guo, Hao Yan, Chen Zhang, and Steven Hoi. Partially observable online change detection via smooth-sparse decomposition. *arXiv preprint arXiv:2009.10645*, 2020.
- [15] Allan Gut and Josef Steinebach. Truncated sequential change-point detection based on renewal counting processes. *Scandinavian journal of statistics*, 29(4):693–719, 2002.
- [16] David V Hinkley and Elizabeth A Hinkley. Inference about the change-point in a sequence of binomial variables. *Biometrika*, 57(3):477–488, 1970.
- [17] Feiyu Jiang, Zifeng Zhao, and Xiaofeng Shao. Time series analysis of covid-19 infection curve: A change-point perspective. *Journal of Econometrics*, 2020. ISSN 0304-4076. .
- [18] Wei Jiang, Lianjie Shu, and Kwok-Leung Tsui. Weighted cusum control charts for monitoring poisson processes with varying sample sizes. *Journal of quality technology*, 43(4):346–362, 2011.
- [19] Jeremias Knoblauch and Theodoros Damoulas. Spatio-temporal bayesian on-line change-point detection with model selection. In *International Conference on Machine Learning*, pages 2718–2727. PMLR, 2018.
- [20] Kaibo Liu, Yajun Mei, and Jianjun Shi. An adaptive sampling strategy for online high-dimensional process monitoring. *Technometrics*, 57(3):305–319, 2015.
- [21] Gary Lorden. Procedures for reacting to a change in distribution. *The Annals of Mathe-*

- matical Statistics*, pages 1897–1908, 1971.
- [22] Jianxi Luo. Predictive monitoring of covid-19. *SUTD Data-Driven Innovation Lab*, 446, 2020.
 - [23] Yajun Mei. Efficient scalable schemes for monitoring a large number of data streams. *Biometrika*, 97(2):419–433, 2010.
 - [24] George V Moustakides. Optimal stopping times for detecting changes in distributions. *the Annals of Statistics*, 14(4):1379–1387, 1986.
 - [25] George V Moustakides, Aleksey S Polunchenko, and Alexander G Tartakovsky. Numerical comparison of cusum and shiryaev–roberts procedures for detecting changes in distributions. *Communications in Statistics—Theory and Methods*, 38(16-17):3225–3239, 2009.
 - [26] E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.
 - [27] E. S. Page. A test for a change in a parameter occurring at an unknown point. *Biometrika*, 42(3/4):523–527, 1955.
 - [28] Andreas Raue, Clemens Kreutz, Thomas Maiwald, Julie Bachmann, Marcel Schilling, Ursula Klingmüller, and Jens Timmer. Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics*, 25(15):1923–1929, 2009.
 - [29] Albert N Shiryaev. Problem of most rapid detection of a disturbance in stationary processes. *DOKLADY AKADEMII NAUK SSSR*, 138(5):1039, 1961.
 - [30] Amna Tariq, Yiseul Lee, Kimberlyn Roosa, Seth Blumberg, Ping Yan, Stefan Ma, and Gerardo Chowell. Real-time monitoring the transmission potential of covid-19 in singapore, march 2020. *BMC medicine*, 18:1–14, 2020.
 - [31] Charles Truong, Laurent Oudre, and Nicolas Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167:107299, 2020.
 - [32] Yao Xie and David Siegmund. Sequential multi-sensor change-point detection. In *2013 Information Theory and Applications Workshop (ITA)*, pages 1–20. IEEE, 2013.
 - [33] 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*, 7(1):12–27, 2012.
 - [34] Qunzhi Xu and Yajun Mei. Multi-stream quickest detection with unknown post-change parameters under sampling control. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 112–117. IEEE, 2021.
 - [35] Qunzhi Xu, Yajun Mei, and George V Moustakides. Optimum multi-stream sequential change-point detection with sampling control. *IEEE Transactions on Information Theory*, 2021.
 - [36] Xian Yu, Michael Baron, and Pankaj K Choudhary. Change-point detection in binomial thinning processes, with applications in epidemiology. *Sequential Analysis*, 32(3):350–367, 2013.
 - [37] Chen Zhang and Steven CH Hoi. Partially observable multi-sensor sequential change detection: A combinatorial multi-armed bandit approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5733–5740, 2019.
 - [38] Chen Zhang, Hao Yan, Seungho Lee, and Jianjun Shi. Multiple profiles sensor-based monitoring and anomaly detection. *Journal of Quality Technology*, 50(4):344–362, 2018.
 - [39] Chen Zhang, Hao Yan, Seungho Lee, and Jianjun Shi. Dynamic multivariate functional data modeling via sparse subspace learning. *Technometrics*, 63(3):370–383, 2021.
 - [40] Wanrong Zhang and Yajun Mei. Bandit change-point detection for real-time monitoring high-dimensional data under sampling control. *arXiv preprint arXiv:2009.11891*, 2020.
 - [41] Zifeng Zhao, Ting Fung Ma, Wai Leong Ng, and Chun Yip Yau. A composite likelihood-based approach for change-point detection in spatio-temporal process. *arXiv preprint arXiv:1904.06340*, 2019.

Appendix A. Proof of Proposition 4.2

Proof. The distribution of $X_{k,t}$ is a binomial distribution with total counts $c_{k,t}$. In the null hypothesis and alternative hypothesis, the probability is p and q , respectively. Thus

$$f_0(X_{k,t}) = \binom{c_{k,t}}{X_{k,t}} p^{X_{k,t}} (1-p)^{c_{k,t}-X_{k,t}}$$

$$f_k(X_{k,t}) = \binom{c_{k,t}}{X_{k,t}} q^{X_{k,t}} (1-q)^{c_{k,t}-X_{k,t}}$$

Plugging in the formula above to equation (4), we have

$$\begin{aligned} \log \frac{f_k(X_{k,t})}{f_0(X_{k,t})} &= X_{k,t} \log \frac{q}{p} + (c_{k,t} - X_{k,t}) \log \frac{1-q}{1-p} \\ &= c_{k,t} \log \frac{1-q}{1-p} + X_{k,t} \log \frac{q(1-p)}{p(1-q)} \end{aligned}$$

And we have the result of equation (5). □

Appendix B. Proof of Proposition 4.3

Proof. For binomial distribution, a reasonable conjugate prior distribution is the beta distribution. The posterior distribution from a weighted update is

$$f(p_{k,t}|X_{k,1}, \dots, X_{k,T}) \propto f_0(p_{k,t}) \prod_{t=1}^T f(X_{k,t}|p_{k,t})^{w_t^T}$$

where w_t^T is a series of time decayed weights with increasing order. Here we set it to be $w_t^T = w^{T-t}$. Recall that $X_{k,t} \sim \text{binomial}(c_{k,t}, p_{k,t})$, we have

$$\begin{aligned} f(p_{k,t}|X_{k,1}, \dots, X_{k,T}) &\propto f_0(p_{k,t}) \prod_{t=1}^T f(X_{k,t}|p_{k,t})^{w_t^T} \\ &\propto p_{k,t}^{a-1} (1-p_{k,t})^{b-1} \prod_{t=1}^T \left(p_{k,t}^{X_{k,t}} (1-p_{k,t})^{c_{k,t}-X_{k,t}} \right)^{w_t^T} \\ &= p_{k,t}^{a+\sum_{t=1}^T w_t^T X_{k,t}-1} (1-p_{k,t})^{b+\sum_{t=1}^T w_t^T (c_{k,t}-X_{k,t})-1} \end{aligned}$$

This is the kernel of the distribution $\text{Beta}(a + \sum_{t=1}^T w^{T-t} X_{k,t}, b + \sum_{t=1}^T w^{T-t} (c_{k,t} - X_{k,t}))$ □

Appendix C. Proof of Proposition 4.4

Proof. We refer to equation (6) to get $\mu + \sigma$ of the increment. Recall that $X_{k,t} \sim \text{Binomial}(c_{k,t}, p_{k,t})$, $p_{k,t} \sim \text{Beta}(\alpha_{k,t}, \beta_{k,t})$ and $c_{k,t}$ are the decision variables, which can be regarded as constant. We have

$$\begin{aligned} E\Delta r_{k,t} &= c_{k,t} \log \frac{1-q}{1-p} + \log \frac{q(1-p)}{p(1-q)} E[E(X_{k,t}|p_{k,t})] \\ &= c_{k,t} \log \frac{1-q}{1-p} + \log \frac{q(1-p)}{p(1-q)} E[c_{k,t}p_{k,t}] \\ &= c_{k,t} \log \frac{1-q}{1-p} + \log \frac{q(1-p)}{p(1-q)} \frac{c_{k,t}\alpha_{k,t}}{\alpha_{k,t} + \beta_{k,t}} \end{aligned}$$

$$\begin{aligned} \text{Var}(X_{k,t}) &= \text{Var}(E[X_{k,t}|p_{k,t}]) + E[\text{Var}(X_{k,t}|p_{k,t})] \\ &= \text{Var}(c_{k,t}p_{k,t}) + E(c_{k,t}p_{k,t}(1-p_{k,t})) \end{aligned}$$

The first term is

$$\text{Var}(c_{k,t}p_{k,t}) = c_{k,t}^2 \frac{\alpha_{k,t}\beta_{k,t}}{(\alpha_{k,t} + \beta_{k,t})^2(\alpha_{k,t} + \beta_{k,t} + 1)}$$

The second term is

$$\begin{aligned} E[c_{k,t}p_{k,t}(1-p_{k,t})] &= c_{k,t}E[p_{k,t}] - c_{k,t}E[p_{k,t}^2] \\ &= c_{k,t} \frac{\alpha_{k,t}}{\alpha_{k,t} + \beta_{k,t}} - c_{k,t} \left[\frac{\alpha_{k,t}\beta_{k,t}}{(\alpha_{k,t} + \beta_{k,t})^2(\alpha_{k,t} + \beta_{k,t} + 1)} + \frac{\alpha_{k,t}^2}{(\alpha_{k,t} + \beta_{k,t})^2} \right] \\ &= c_{k,t} \frac{\alpha_{k,t}\beta_{k,t}}{(\alpha_{k,t} + \beta_{k,t})(\alpha_{k,t} + \beta_{k,t} + 1)} \end{aligned}$$

Then, we can calculate the variance of $\Delta r_{k,t}$ as

$$\begin{aligned} \sigma^2 &= \left(\log \frac{q(1-p)}{p(1-q)} \right)^2 \text{Var}(X_{k,t}) \\ &= \left(\log \frac{q(1-p)}{p(1-q)} \right)^2 c_{k,t} \frac{\alpha_{k,t}\beta_{k,t}}{(\alpha_{k,t} + \beta_{k,t})(\alpha_{k,t} + \beta_{k,t} + 1)} \left(\frac{c_{k,t}}{\alpha_{k,t} + \beta_{k,t}} + 1 \right) \end{aligned}$$

Recall that the target function is formulated to be the sum of the mean and standard deviation. We also have a constrain that $\sum_{k=1}^K c_{k,t} = C$. Then when we sum up $E\Delta r_{k,t}$, the first term of the expectation part will be a constant and can be left out from the target function. We also notice that when taking the squared root of σ^2 , the coefficient will be the same as the coefficient of the second term in $E\Delta r_{k,t}$. Thus, this coefficient can also be removed. The simplified reward function can now be formulated as a formula (10) and the tests distribution of next day $t + 1$ is to solve the optimization problem (11) $\square$

Appendix D. Sensitivity Analysis

In this section, we will study the effect of hyperparameters w and a, b on the average run length by performing an additional simulation study. Here, we will start with the effect of w on the average run length. When $a = 39/2$, $b = 39/2 * 99$, the average run length is around 200, when $w = 0.01, 0.3, 0.6$. The detection delay and detection precision are in Table D1.

Table D1.: Comparison of detection delay (ARL_1) and Detection precision (DP) for different weight w

	w	$q = 0.025$	$q = 0.03$	$q = 0.04$	$q = 0.05$
ARL_1	0.01	9.362	5.23	3.391	2.843
	0.3	7.893	4.958	3.388	2.863
	0.6	7.293	4.793	3.438	2.88
DP	0.01	0.907	0.919	0.928	0.931
	0.3	0.918	0.932	0.938	0.939
	0.6	0.919	0.929	0.937	0.939

From Table D1, we can conclude that when q is small, a smaller w results a larger ARL_1 or higher detection delay. However, $w = 0.3$ and $w = 0.6$ yield similar ARL_1 . When studying one specific iteration, we found that the hyperparameter w affects the distribution of the testing kits. Figure D1 is the box-plot of a region for different weights as $w = 0.01, 0.3, 0.6$. We can see that the median of test distributions is similar for different weights. However, when w is large, it results in more days with extremely large numbers of testing kits. This is because when w is large, it puts more weight on the previous days to update the posterior distribution of the infection rate. Therefore, it is harder to reduce the number of testing kits the next day compared to when w is small.

We also studied the impact of different a in the prior distribution, given that the ratio of a and b are kept the same in the simulation study. The testing kit distribution of one county in the in-control state is shown in Figure D2. Intuitively, a higher a makes it harder to distribute more tests in a region. For example, if a is 19.5, the algorithm will allocate fewer tests in the region with 10 tests and 2 positive cases than when a is 1. The simulated result of detection delay and detection precision when the average run length in control is around 200 is as Table D2. We can see that when a increases from 0.1 to 19.5, the detection delay decreases, and the detection precision increases. Therefore, the choice of a and b depends on how much weight we would like to rely on prior knowledge.

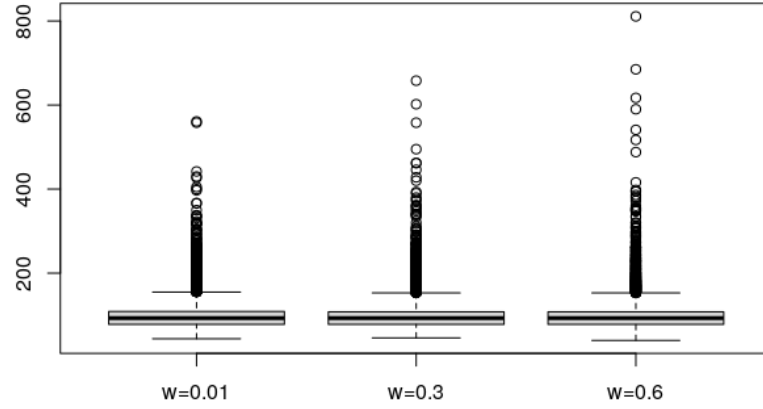


Figure D1.: Boxplot of the testing kit distribution in one county before the change for different w

Table D2.: Comparison of detection delay (ARL_1) and Detection precision (DP) for different a

	a	$q = 0.025$	$q = 0.03$	$q = 0.04$	$q = 0.05$
ARL_1	0.1	17.463	10.229	8.097	7.247
	1	12.453	8.064	6.214	5.373
	19.5	7.893	4.958	3.358	2.863
DP	0.1	0.88	0.91	0.925	0.93
	1	0.901	0.923	0.93	0.939
	19.5	0.918	0.932	0.938	0.939

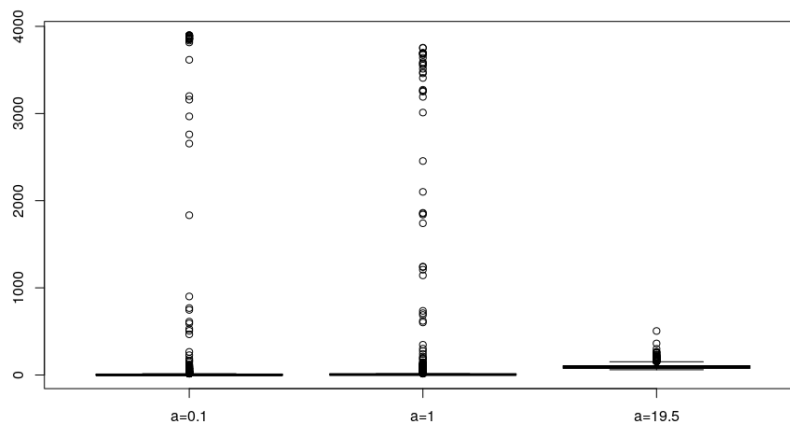


Figure D2.: Boxplot of the testing kit distribution in one county before the change for different a