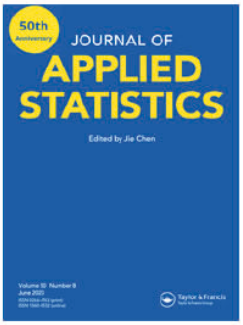

Journal of Applied Statistics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/cjas20

Hot-spots detection in count data by Poisson assisted smooth sparse tensor decomposition

Yujie Zhao, Xiaoming Huo & Yajun Mei

To cite this article: Yujie Zhao, Xiaoming Huo & Yajun Mei (2022): Hot-spots detection in count data by Poisson assisted smooth sparse tensor decomposition, Journal of Applied Statistics, DOI: 10.1080/02664763.2022.2112557

To link to this article: https://doi.org/10.1080/02664763.2022.2112557

Published online: 25 Aug 2022.

JOURNAL OF APPLIED STATISTICS
https://doi.org/10.1080/02664763.2022.2112557

Taylor & Francis
Taylor & Francis Group

Hot-spots detection in count data by Poisson assisted smooth sparse tensor decomposition

Yujie Zhao [a], Xiaoming Huo [b] and Yajun Mei [b]

[a]Biostatistics and Research Decision Sciences Department, Merck & Co., Inc, North Wales, PA, USA; [b]School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA

ABSTRACT

Count data occur widely in many bio-surveillance and healthcare applications, e.g. the numbers of new patients of different types of infectious diseases from different cities/counties/states repeatedly over time, say, daily/weekly/monthly. For this type of count data, one important task is the quick detection and localization of hot-spots in terms of unusual infectious rates so that we can respond appropriately. In this paper, we develop a method called *Poisson assisted Smooth Sparse Tensor Decomposition (PoSSTenD)*, which not only detect when hot-spots occur but also localize where hot-spots occur. The main idea of our proposed PoSSTenD method is articulated as follows. First, we represent the observed count data as a three-dimensional tensor including (1) a spatial dimension for location patterns, e.g. different cities/countries/states; (2) a temporal domain for time patterns, e.g. daily/weekly/monthly; (3) a categorical dimension for different types of data sources, e.g. different types of diseases. Second, we fit this tensor into a Poisson regression model, and then we further decompose the infectious rate into two components: smooth global trend and local hot-spots. Third, we detect when hot-spots occur by building a cumulative sum (CUSUM) control chart and localize where hot-spots occur by their LASSO-type sparse estimation. The usefulness of our proposed methodology is validated through numerical simulation studies and a real-world dataset, which records the annual number of 10 different infectious diseases from 1993 to 2018 for 49 mainland states in the United States.

ARTICLE HISTORY
Received 15 January 2022
Accepted 8 August 2022

KEYWORDS
Hot-spots detection; spatio-temporal model; tensor decomposition; CUSUM; Poisson regression

1. Introduction

Count data are frequently seen in many real-world applications in fields such as biosurveillance (see [7]), epidemiology and sociology (see [52]). Usually, the count data are collected from multiple data sources and from many spatial locations repeatedly over time, say, daily, monthly, or annually. For instance, the daily number of people infected by some types of diseases in 50 states in the United States. We call this type of data as *multivariate spatio-temporal count data*, which has three domains: (1) the spatial domain recording the locations, (2) the temporal domain recording the time, and (3) the categorical domain

CONTACT Yujie Zhao yujie.zhao@merck.com 351 N Sumneytown Pike, North Wales, PA 19454, USA
This article has been corrected with minor changes. These changes do not impact the academic content of the article.

recording the different types of data sources. When such count data are non-stationary over time, there are two standard research areas: one is model fitting and prediction, and the other is the detection of global/system-wise changes, seeing the classical statistical process control (SPC) or sequential change-point detection literature.

In this paper, we investigate a new research area of multivariate spatio-temporal count data by focusing on the detection and localization of *hot-spots*. Here, the hot-spots are defined as the anomalies/outliers in term of infectious rates occurring in the temporal, spatial, or categorical domains. There are a couple of subtasks: the *hot-spots detection* aims to detect when hot-spots occur (in the temporal domain) and the *hot-spots localization* aims to localize where hot-spots occur (in the spatial and categorical domain).

It is informative to point out that there are two kinds of anomaly detection for multivariate data: one is global-level change (e.g. the *first-order* changes), and the other is local-level hot-spots (e.g. *second-order* changes). The former is the focus of the classical SPC or sequential change-point detection methods, whereas the latter is the main interest of this paper. Here we assume that local-level hot-spots have (1) spatial sparsity, i.e. the local changes are sparse in the spatial and categorical domains; (2) temporal consistency, i.e. the local changes last for a while once they occur.

In the remainder of this section, we present the literature review in Section 1.1 and articulate our proposed method and our contribution in Section 1.2.

1.1. Literature review

In this subsection, we review the existing literature on hot-spots detection based on three types of count data: univariate, multivariate, and multivariate spatio-temporal.

For univariate count data, i.e. when we observe the data of the form $\{y_t\}_{t=1,\ldots,T}$ with $y_t \in \mathbb{N}$ (a set of non-negative integers) denoting the observed count data at time t, there are mainly four methods to detect hot-spots: (1) time series based method (see [21]); (2) control charts based method by using the likelihood functions, such as the cumulative sum (CUSUM) control chart (see [15]) and exponentially weighted moving average (EWMA) control chart (see [15]); (3) scan statistics based method (see [27]), including both the frequentist scan statistics (see [28]) and Bayesian scan statistics (see [33]); (4) Poisson process-based method that is often combined with Bayesian methods (see [20]).

For multivariate count data, i.e. when we observe the data of the form $\{y_{it}\}_{i=1,\ldots,n,t=1,\ldots,T}$ with $y_{it} \in \mathbb{N}$, there are mainly three methods widely-used hot-spots detection: (1) control chart based method, including CUSUM or EWMA control charts (see [29]); (2) Poisson process based method (see [43]); (3) hidden Markov model (HMM) based method (see [8]).

For multivariate spatio-temporal count data, i.e. when the observed data are of the form $y_{ijt} \in \mathbb{N}$ that denotes the observed count data over the i-th spatial location for the j-th type of disease at time t with $i = 1,\ldots,n_1, j = 1,\ldots,n_2, t = 1,\ldots,T$, research on hot-spots detection is rather limited, although there are three types of closely related methods developed for other purposes or other contexts: (1) scan statistics for anomalous clusters, (2) tensor decomposition for Gaussian data, and (3) change point detection for global changes. Since our paper deals with multivariate spatio-temporal count data, below we will provide a more detailed review of these three methods.

For the scan statistics based method, it was first developed in the 1960's (see [32]) and later extended by [25] to detect anomalous clusters in spatio-temporal data. The main idea of scan statistics is to detect the abnormal clusters by utilizing a maximal log-likelihood ratio. It is worth noting that the scan statistics-based method is a parametric method that assumes the data distributions are known. For instance, authors of [41] assumes the negative binomial distribution, [26] investigate the Poisson distribution. A limitation of the scan statistics is that it assumes that the background is independent and identically distributed (i.i.d.) or follows a rather simple probability distribution, which might not be suitable to handle non-stationary spatio-temporal data.

For the tensor decomposition-based detection, one representative is [46] for Gaussian data, and below is the main idea. First, it assumes that the raw data follows the normal distribution and then decomposes it into three components: global trend mean, local hot-spots, and residuals. Then, it uses the Tucker decomposition (see [18]) to decompose the first two components. Finally, it detects the hot-spots by monitoring the model residuals. Similar ideas can also be found in elsewhere, for example, [45]. The limitation of this method is that it does not consider the effect of population size and the normal distribution is clearly inappropriate for count data, especially those small values. Another representative is [11], where the authors first assume the count data follows the Poisson distribution and then decompose the Poisson mean by canonical polyadic decomposition (CPD) (see [18]), and finally localize the hot-spots by small p-values. The limitation of this approach is that it does not remove the global trend mean from the raw data, which leads to its incapacity to detect the local hot-spots. Besides, it can only realize the hot-spots localization, and the analysis of the detection delay is not reported.

For the change-point detection framework, there are two related categories, the *Least Absolute Shrinkage and Selection* (Lasso) based methods and *dimension reduction based methods*. For the Lasso-based change-point detection method, one of the existing research representative is [53]. Note that Lasso has been demonstrated to be an effective method for variable selection to address sparsity issues for high-dimensional data in the past decades since its developments in [42], and thus it is natural to apply it to detect sparse changes in high-dimensional data (see [50,53,54]). While the sparse change of Lasso is similar to the hot-spots, unfortunately, as our extensive simulation studies will demonstrate, the Lasso-based control chart is unable to separate the local hot-spots from the non-stationary global trend mean in the spatio-temporal data. For the dimension reduction-based method, the representative is [10]. The main idea of this type of method is that it first uses principal component analysis (PCA) or other dimension reduction methods to extract the features from the high-dimensional data, and then it combines the reduced dimension information with change-point models to detect the hot-spots. For other dimension-reduction-methods, please see [6,10,36] for more details. The drawbacks of the dimension reduction-based method are the restriction of the change-point detection problem and the failure to consider the spatial sparsity and temporal consistency of hot-spots.

1.2. Overview of our proposed method and its contributions

The essential idea of our proposed method is to combine the tensor decomposition-based method and the Lasso-based method over the multivariate spatio-temporal data. Mathematically speaking, the multivariate spatio-temporal data can be a *tensor* of order three,

which is a multi-dimensional array, i.e. $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. Here n_1 is the number of the spatial locations, n_2 is the number of different diseases or variates, and n_3 is the number of time points. In our proposed method, we assume the tensor $\mathcal{Y}$ comes from the Poisson distribution and then consider the additive model that decomposes its Poisson rate into two components: (1) smooth but non-stationary global trend mean, and (2) sparse local hot-spots. Next, we fit the raw data with a penalty function, i.e. a Lasso type penalty to guarantee the spatial sparsity of hot-spots. This allows us to not only detect when the hot-spots happen over the temporal domain (i.e. hot-spots detection problem) but also localize where and which types/attributes of the hot-spots occur if the change happens (i.e. hot-spots localization problem). We term our proposed decomposition method as Poisson-assisted Smooth Sparse Tensor Decomposition (PoSSTenD).

It is useful to highlight the novelty of our proposed method as compared to those existing methods on multivariate spatio-temporal count data. First, our proposed PoSSTenD method takes into account of the effect of population size to focus on the anomalous infectious diseases rates or Poisson rates, which is important in many real-world applications. Second, our proposed PoSSTenD method can detect hot-spots when the global trend of the spatio-temporal data is dynamic (i.e. non-stationary or non-i.i.d). That is, our method is robust no matter whether the global trend is decreasing, stationary, or increasing, in the sense that it can detect hot-spots with positive or negative local mean shifts on top of the global trend of raw data. In comparison, existing SPC or change-point detection methods often assume that the background is i.i.d. and focus on detecting the anomalies under the static and i.i.d. background. Finally, while our paper focuses only on the tensor of order three arising from our motivating application in 10 annual infectious diseases in the U.S., our proposed hot-spots method can easily be extended to the tensor of order d ($d \geq 3$), as we can simply add corresponding dimensions and bases in the tensor analysis. The capability of extending to high-dimensional tensor data is one of the main advantages of our proposed PoSSTenD method.

We would like to clarify that our contribution lies in the hot-spots detection and localization, instead of model fitting/prediction of the multivariate spatio-temporal count data. For the fitness of the count data, there are lots of well-established methods. One widely used one is the generalized linear regression model (see [16, Chapter 3]). Another popular method is the time series models such as Autoregressive Moving Average (ARMA). In this paper, we will not compare our proposed PoSSTenD method to these methods, because our objective is to detect and localize hot-spots, rather than model fitting.

The remainder of this paper is as follows. In Section 2, we introduce and visualize a motivating dataset. In Section 3, we present our proposed PoSSTenD method, discuss how to estimate model parameters from data, and describe how to use our proposed PoSSTenD method to detect and localize hot-spots. Our proposed method is compared with several benchmark methods in Section 4, demonstrating its usefulness through extensive simulations. The application of our proposed method in the dataset (described in Section 2) is reported in Section 2.

2. Motivating example & tensor background

In this section, we provide a detailed description of our motivating dataset, and some tensor background to help readers better understand our method in Section 3.

2.1. Motivating example

Our motivated data are from CDC on the number of annual new cases infected by 10 different diseases of 49 mainland US states from 1993 to 2018. The 10 infectious diseases are (1) mumps, (2) legionellosis, (3) tuberculosis, (4) syphilis, (5) shigellosis, (6) pertussis, (7) hepatitis A, (8) hepatitis B, (9) rabies, (10) malaria. The dataset is publicly available in https://www.cdc.gov/mmwr/mmwr_nd/index.html (for data from 1993 to 2015) and https://wonder.cdc.gov/nndss/nndss_annual_tables_menu.asp (for data from 2016 to 2018). For illustration, we present the a selected subset of this dataset in Table 1, where the states are listed in alphabetical order.

Mathematically, we can organize the above motivating dataset into a tensor of order three, i.e. $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, where $n_1 = 49$ is the number of states, $n_2 = 10$ is the number of diseases, and $n_3 = 26$ is the number of observed years. For this tensor of order three $\mathcal{Y}$, its (i, j, t)-th entry, i.e. $\mathcal{Y}_{i,j,t}$ is the number of people get infected by the j-th type of disease in the i-th state i in year t.

To better capture the character of the data $\mathcal{Y}$, we provide some data visualization plots in Figure 1. In Figure 1(a), we show the bar plot for the total number of affected subjects for these ten different diseases. Here the j-th bar represents the j-th type of disease and the height of j-th bar is the value of $\sum_{i=1}^{n_1} \sum_{t=1}^{n_3} \mathcal{Y}_{i,j,t}$. In Figure 1(b), we plot the cumulative number of infected subjects of 49 states in a map, where the color of the i-th state is decided by the total value $\sum_{j=1}^{n_2} \sum_{t=1}^{n_3} \mathcal{Y}_{i,j,t}$ and a larger value leads to a deeper color. From this plot, we find that generally speaking, California, Texas, and New York observe a larger number of infected subjects as compared with other states, possibly due to a large population size. In Figure 1(c), we display a time series plot, where the x-axis is the year and the y-axis is the logarithm of the value of $\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathcal{Y}_{i,j,t}$ for $t = 1, 2, \ldots, n_3$. We can see that, there is a trough from 2000 to 2010, but the number of infected people has an increasing trend otherwise.

2.2. Tensor background

In this subsection, we present necessary tensor notations and tensor algebra so as to help readers understand our proposed methodology in Section 3.

First, for the notations throughout the paper, scalars are denoted by lowercase letters (e.g. θ, y), and vectors are denoted by lowercase boldface letters ($\boldsymbol{\theta}, \mathbf{y}$), whose i-th entry is denoted by the subscript ($\boldsymbol{\theta}_i, \mathbf{y}_i$). Matrices are denoted by uppercase boldface letter ($\boldsymbol{\Theta}, \mathbf{Y}$), whose (i, j)-th entry is denoted by the subscript ($\boldsymbol{\Theta}_{i,j}, \mathbf{Y}_{ij}$). Tensors are denoted by curlicue letter ($\vartheta, \mathcal{Y}$). For example, an K-dimension tensor is represented by $\vartheta \in \mathbb{R}^{I_1 \times \ldots \times I_K}$, where I_k represent the mode-K dimension of ϑ for $k = 1, \ldots, K$. Its $(i_1, i_2, \ldots, i_K)$-th entry is denoted by the subscript $(\vartheta_{i_1,i_2,\ldots,i_K}, \mathcal{Y}_{i_1,i_2,\ldots,i_K})$.

Next, we introduce the definition of *slice* in tensor. The slice is a two-dimensional section of a tensor by fixing all but two indices. Let us take a 3-dimension tensor $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ as an example. Its horizontal, lateral, and frontal slides are denoted by $\mathcal{Y}_{i::} (\forall i = 1, \ldots, n_1)$, $\mathcal{Y}_{:j:} (\forall j = 1, \ldots, n_2)$, and $\mathcal{Y}_{::t}$ $(\forall t = 1, \ldots, n_3)$, respectively. A visualization of the aforementioned three types of slides are available in Figure 3 of [51].

Finally, we introduce the *Tucker decomposition*. It can be regarded as an extension of singular value decomposition (SVD) in the matrix. Mathematically, it decomposes a tensor

Table 1. The head of our motivating dataset, which records number of people get infected by ten different types of diseases for all states in U.S. from 1993 to 2018.

State	Year	Mumps	Legionellosis	Tuberculosis	Syphilis	Shigellosis	Pertussis	Hepatitis A	Hepatitis B	Rabies	Malaria
Alabama	1993	22	2	487	2333	375	65	58	107	116	7
Arizona	1993	19	17	231	557	693	70	1493	96	60	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Wyoming	1993	5	7	7	9	22	2	17	34	25	1
Alabama	1994	12	13	433	1933	418	35	93	91	128	9
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Wyoming	2018	1	2	1	42	3	62	5	2	39	2

(a) (b) (c)

Figure 1. Data visualization of our motivating dataset. (a) bar plot of 10 diseases. (b) map of 49 states. (c) time series of 26 years.

$\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_d}$ into a product of a 'core tensor' $\mathcal{C}$ and several matrices $\mathbf{B}_1, \ldots, \mathbf{B}_d$ which correspond to different core scalings along each mode, i.e.

$$\mathcal{Y} = \mathcal{C} \times_1 \mathbf{B}_1 \times_2 \mathbf{B}_2 \ldots \times_d \mathbf{B}_d,$$

where $\times_n$ is the mode-n product for any $n = 1, \ldots, d$ (see [51, Section 2]). And the core tensor $\mathcal{C}$ is of dimension $\mathcal{C} \in \mathbb{R}^{p_1 \times p_2 \times \ldots \times p_d}$. The matrix $\mathbf{B}_i$ is of dimension $\mathbf{B}_i \in \mathbb{R}^{n_i \times p_i}$ for any $i = 1, \ldots, d$. In most of the applications, we assume the matrices $\mathbf{B}_1, \ldots, \mathbf{B}_d$ are known (see [44,45,47]). The high-level criterion to select $\mathbf{B}_1, \ldots, \mathbf{B}_d$ are listed as follows. If the data $\mathcal{Y}$ has a smooth pattern on mode-n, then one can use the either spline basis (see [14,47]) or Gaussian kernel (see [47]) for $\mathbf{B}_n$. If the data $\mathcal{Y}$ has no prior information on mode-n, then one can use the identity matrix for $\mathbf{B}_n$ (see [51]). And we call $\min\{p_1, p_2, \ldots, p_d\}$ as the rank of the Tucker decomposition (see [23]).

3. Our proposed method

In this section, we present our proposed PoSSTenD method, including the optimization algorithm for the model parameter estimation, and statistical procedures for hot-spots detection and localization.

The structure of this section is organized as follows. Subsection 3.1 develops our proposed method. Subsection 3.2 describes the parameters estimation and the algorithm to solve the estimation problems. Subsection 3.3 discusses the procedure to detect and localize the hot-spots.

3.1. Model

In this subsection, we present the mathematical model of our proposed PoSSTenD method. For better illustration, we take the motivating dataset described in Section 2 as an example, which is a tensor of order three, i.e. $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. The (i, j, t)-th entry of $\mathcal{Y}$ is the number of people who are infected by the j-th type of disease in the i-th state in year t. Since the number of infected patients is count data, we assume that it follows the Poisson distribution:

$$\mathcal{Y}_{i,j,t} \sim \text{Poisson}(\mathcal{R}_{i,j,t}\mathcal{N}_{i,j,t}) \tag{1}$$

where $\mathcal{R}_{i,j,t}$ is the infectious rate of the j-th type of disease in the i-th state in year t, and $\mathcal{N}_{i,j,t}$ denotes the corresponding population size, for $i = 1 \ldots n_1, j = 1 \ldots n_2, t = 1 \ldots n_3$.

In our simulation studies and case study, we assume that the population size $\mathcal{N}_{i,j,t}$ is the same over different disease type j, although they can be different under other scenarios, e.g. when certain diseases will affect certain sub-populations depending on gender, age, race, etc.

In our paper, we are interested in detecting and localizing the hot-spots in infectious rate $\mathcal{R}_{i,j,t}$, and thus we further decomposed the logarithm of the infectious rates into two components: the smooth global trend mean $\mathcal{U}_{i,j,t}$ and the local hot-spots $\mathcal{H}_{i,j,t}$. Mathematically, it is an additive model:

$$\log(\mathcal{R}_{i,j,t}) = \mathcal{U}_{i,j,t} + \mathcal{H}_{i,j,t}, \tag{2}$$

where $\mathcal{U}_{i,j,t}$ and $\mathcal{H}_{i,j,t}$ are the (i,j,t)-th entry of tensors $\mathcal{U}$ and $\mathcal{H} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, respectively.

For the first component – the global trend mean $\mathcal{U} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ – we apply Tucker decomposition for dimension reduction of tensors, i.e.

$$\mathcal{U} = \vartheta_m \times_1 \mathbf{B}_{m,1} \times_2 \mathbf{B}_{m,2} \times_3 \mathbf{B}_{m,3}, \tag{3}$$

where $\mathbf{B}_{m,1} \in \mathbb{R}^{n_1 \times p_1}, \mathbf{B}_{m,2} \in \mathbb{R}^{n_2 \times p_2}, \mathbf{B}_{m,3} \in \mathbb{R}^{n_3 \times p_3}$ are the pre-assigned basis describing the within-state correlation, within-disease-type correlation, and within-year correlation in $\mathcal{U}$, respectively. For the selection of $\mathbf{B}_{m,1}, \mathbf{B}_{m,2}, \mathbf{B}_{m,3}$, we postpone its discussion at the end of this subsection. The operators $\times_1, \times_2, \times_3$ are the mode-n product reviewed in Section 2.2 with $n = 1, 2, 3$. These three mode-n products are used to model the between-dimension correlations in $\mathcal{U}$. The tensor $\vartheta_m \in \mathbb{R}^{p_1 \times p_2 \times p_3}$ is an unknown tensor parameter to be estimated.

For the second component – local hot-spots $\mathcal{H} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ – we apply the similar Tucker decomposition as in $\mathcal{U}$:

$$\mathcal{H} = \vartheta_h \times_1 \mathbf{B}_{h,1} \times_2 \mathbf{B}_{h,2} \times_3 \mathbf{B}_{h,3}, \tag{4}$$

where the $\vartheta_h \in \mathbb{R}^{q_1 \times q_2 \times q_3}$ is the unknown parameter to be estimated and basis $\mathbf{B}_{h,1} \in \mathbb{R}^{n_1 \times q_1}, \mathbf{B}_{h,2} \in \mathbb{R}^{n_2 \times q_2}, \mathbf{B}_{h,3} \in \mathbb{R}^{n_3 \times q_3}$ are pre-assigned matrices describing the within-state correlation, within-disease-type correlation, and within-year correlation in $\mathcal{H}$, respectively. The suitable choices of these bases will be discussed in more details in Section 4. The operators $\times_1, \times_2, \times_3$ are the mode-n product used to model the between-dimension correlations in $\mathcal{H}$.

By combining (1), (2), (3), (4), we summarize our method as

$$\begin{cases} \mathcal{Y}_{i,j,t} \sim \text{Poisson}(\mathcal{R}_{i,j,t}\mathcal{N}_{i,j,t}) \\ \log(\mathcal{R}_{i,j,t}) = (\vartheta_m \times_1 \mathbf{B}_{m,1} \times_2 \mathbf{B}_{m,2} \times_3 \mathbf{B}_{m,3})_{i,j,t} + (\vartheta_h \times_1 \mathbf{B}_{h,1} \times_2 \mathbf{B}_{h,2} \times_3 \mathbf{B}_{h,3})_{i,j,t} \end{cases},$$

where $(\vartheta_m \times_1 \mathbf{B}_{m,1} \times_2 \mathbf{B}_{m,2} \times_3 \mathbf{B}_{m,3})_{i,j,t}, (\vartheta_h \times_1 \mathbf{B}_{h,1} \times_2 \mathbf{B}_{h,2} \times_3 \mathbf{B}_{h,3})_{i,j,t}$ denote the (i,j,t)-th entry in tensor $\vartheta_m \times_1 \mathbf{B}_{m,1} \times_2 \mathbf{B}_{m,2} \times_3 \mathbf{B}_{m,3}, \vartheta_h \times_1 \mathbf{B}_{h,1} \times_2 \mathbf{B}_{h,2} \times_3 \mathbf{B}_{h,3} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, respectively.

This tensor representation allows us to develop computationally efficient methods for estimation and prediction. Under tensor algebra, the above model can be rewritten as

$$\begin{cases} \mathbf{y} \dot{\sim} \text{Poisson}(\mathbf{r} \odot \mathbf{n}) \\ \log(\mathbf{r}) = \left(\mathbf{B}_{m,1} \otimes \mathbf{B}_{m,2} \otimes \mathbf{B}_{m,3}\right) \boldsymbol{\theta}_m + \left(\mathbf{B}_{h,1} \otimes \mathbf{B}_{h,2} \otimes \mathbf{B}_{h,3}\right) \boldsymbol{\theta}_h \end{cases}. \tag{5}$$

Here vectors $\mathbf{y}, \mathbf{r}, \mathbf{n} \in \mathbb{R}^{n_1 n_2 n_3}$, $\boldsymbol{\theta}_m \in \mathbb{R}^{p_1 p_2 p_3}$, $\boldsymbol{\theta}_h \in \mathbb{R}^{q_1 q_2 q_3}$ are the vectorized form of tensor $\mathcal{Y}, \mathcal{R}, \mathcal{N} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\vartheta_m \in \mathbb{R}^{p_1 \times p_2 \times p_3}$, $\vartheta_h \in \mathbb{R}^{q_1 \times q_2 \times q_3}$, respectively. The operator $\odot$ is the Hadamard product, i.e. the output of $\mathbf{r} \odot \mathbf{n}$ is a vector of length $n_1 n_2 n_3$, whose i-th entry is $r_i \times n_i$ with r_i, n_i as the i-th elements of vector $\mathbf{r}, \mathbf{n}$, respectively. The operator $\otimes$ is the Kronecker product whose mathematical definition can be found in [48,51]. Besides, the operator $\dot{\sim}$ means that the i-th entry of $\mathbf{y}$ follows the Poisson distribution with parameter $r_i n_i$ for any $i = 1, \ldots, n_1 n_2 n_3$.

In the reminder of this subsection, we articulate the selection of basis for global trend mean $\mathbf{B}_{m,1}, \mathbf{B}_{m,2}, \mathbf{B}_{m,3}$, and basis for local hot-spots $\mathbf{B}_{h,1}, \mathbf{B}_{h,2}, \mathbf{B}_{h,3}$, particularly in our case study.

For the selection of $\mathbf{B}_{m,1}, \mathbf{B}_{m,2}, \mathbf{B}_{m,3}$, given $\mathcal{Y}$'s smooth pattern, we use the cubic B-spline basis with 8, 7, 7 knots in the x-direction (spatial domain), y-direction (disease-categorical domain) and z-direction (temporal domain). This selection strategy is also adopted in Section 6 of [47], and the above three sets of knots are reasonable in our case study. In general, a smaller number of knots may result in the loss of detection accuracy, and a larger number of knots will lead to the selection of normal regions with large noises. As pointed out by [40], as long as the number of knots is sufficiently large to capture the variation of the background, a further increase in knots will have little effect due to the regulation of smoothness. After verifying the above three sets of knots by generalized cross validation (GCV), as suggested by [40], we find they are sufficient large to capture the variation of the global trend mean. Other choice of $\mathbf{B}_{m,1}, \mathbf{B}_{m,2}, \mathbf{B}_{m,3}$ is welcomed, for example, the Gaussian kernel basis (see [47,51]). In this paper, we will use the B-spline basis for illustration of our proposed PoSSTenD method. Given the above selection of $\mathbf{B}_{m,1}, \mathbf{B}_{m,2}, \mathbf{B}_{m,3}$, the rank of the Tucker decomposition of $\mathcal{U}$ is $\min\{p_1, p_2, p_3\} = 4$.

For the selection of $\mathbf{B}_{h,1}, \mathbf{B}_{h,2}, \mathbf{B}_{h,3}$, we set them as the identity matrices, due to the following two reasons. First, there is no prior information about the hot-spots distribution, which makes identity matrices a reasonable choice. Second, we aim at detecting sparse hot-spots. If one in interested in detecting clustered hot-spots, then a spline basis is a better choice (see [45, Section 3.4]). Given the above selection of $\mathbf{B}_{h,1}, \mathbf{B}_{h,2}, \mathbf{B}_{h,3}$, the rank of the Tucker decomposition of $\mathcal{H}$ is $\min\{n_1, n_2, n_3\} = 10$ in the our case. This rank is not reduced much, because we do not want to lose much information considering the hot-spots detection accuracy.

3.2. Estimation and optimization

This subsection discusses the parameter estimation of our proposed PoSSTenD method. We will first formulate the parameter estimation as an optimization problem in Subsection 3.2.1, and then develop an efficient algorithm to solve this optimization problem in Subsection 3.2.2.

3.2.1. Estimation of the model parameters

To estimate the two unknown vector $\boldsymbol{\theta}_m \in \mathbb{R}^{p_1 p_2 p_3}, \boldsymbol{\theta}_h \in \mathbb{R}^{q_1 q_2 q_3}$, it is natural to first consider the maximal likelihood estimating (MLE) method. We begin with the likelihood function

$$L(\boldsymbol{\theta}_m, \boldsymbol{\theta}_h) = \prod_{i=1}^{n_1 n_2 n_3} (n_i r_i)^{y_i} e^{-r_i n_i} / y_i!,$$

where n_i, r_i, y_i are the i-th elements of vector $\mathbf{n}, \mathbf{r}, \mathbf{y} \in \mathbb{R}^{n_1 n_2 n_3}$, respectively. Accordingly, the log-likelihood function of $(\boldsymbol{\theta}_m, \boldsymbol{\theta}_h)$, i.e. $\ell(\boldsymbol{\theta}_m, \boldsymbol{\theta}_h) = \log L(\boldsymbol{\theta}_m, \boldsymbol{\theta}_h)$ can derived as

$$\ell(\boldsymbol{\theta}_m, \boldsymbol{\theta}_h) = \sum_{i=1}^{n_1 n_2 n_3} y_i \log(n_i r_i) - n_i r_i - \log(y_i!) \propto \sum_{i=1}^{n_1 n_2 n_3} y_i \log(r_i) - n_i r_i,$$

where $\propto$ is done by removing the constant unrelated to $(\boldsymbol{\theta}_m, \boldsymbol{\theta}_h)$. If we denote the $\ell^*(\boldsymbol{\theta}_m, \boldsymbol{\theta}_h) := \sum_{i=1}^{n_1 n_2 n_3} y_i \log(r_i) - n_i r_i$ then by plugging (5) in, we have

$$\ell^*(\boldsymbol{\theta}_m, \boldsymbol{\theta}_h) = \sum_{i=1}^{n_1 n_2 n_3} y_i (\mathbf{B}_m \boldsymbol{\theta}_m + \mathbf{B}_h \boldsymbol{\theta}_h)_i - n_i e^{(\mathbf{B}_m \boldsymbol{\theta}_m + \mathbf{B}_h \boldsymbol{\theta}_h)_i}.$$

By denoting the i-th row of matrix $\mathbf{B}_m, \mathbf{B}_h$ as $\mathbf{x}_i^\top, \mathbf{z}_i^\top$, respectively, we can rewrite the above equation as

$$\ell^*(\boldsymbol{\theta}_m, \boldsymbol{\theta}_h) = \sum_{i=1}^{n} y_i \left(\mathbf{x}_i^\top \boldsymbol{\theta}_m + \mathbf{z}_i^\top \boldsymbol{\theta}_h\right) - n_i e^{\left(\mathbf{x}_i^\top \boldsymbol{\theta}_m + z_i^\top \boldsymbol{\theta}_h\right)},$$

where $n = n_1 n_2 n_3$. Given the above log-likelihood function, on the one hand, we would like to minimize it with respect to $\boldsymbol{\theta}_m, \boldsymbol{\theta}_h$. On the other hand, we would like to detect the hot-spots with sparsity. Considering the trade-off between the maximization of a log-likelihood function $\ell^*(\boldsymbol{\theta}_m, \boldsymbol{\theta}_h)$ and the detection of sparse hot-spots $\boldsymbol{\theta}_h$, we propose to estimate the model parameter $(\boldsymbol{\theta}_m, \boldsymbol{\theta}_h)$ under the penalized likelihood ration framework, which yields the following Lasso-type optimization problem:

$$F(\boldsymbol{\theta}_m, \boldsymbol{\theta}_h) = \sum_{i=1}^{n_1 n_2 n_3} \left[-y_i \left(\mathbf{x}_i^\top \boldsymbol{\theta}_m + \mathbf{z}_i^\top \boldsymbol{\theta}_h\right) + n_i e^{\left(\mathbf{x}_i^\top \boldsymbol{\theta}_m + \mathbf{z}_i^\top \boldsymbol{\theta}_h\right)}\right] + \lambda \|\boldsymbol{\theta}_h\|_1. \tag{6}$$

Here the first term is the negative log-likelihood function and the second term is a ℓ_1 regularization term, which gives sparsity of the hot-spots estimators. The parameter $\lambda > 0$ trades off the minimization of the negative log-likelihood function and the sparsity of the estimators, and is often chosen by cross-validation of the training data.

3.2.2. Optimization algorithm

In this subsection, we discuss the optimization algorithm to minimize the objective function $F(\boldsymbol{\theta}_m, \boldsymbol{\theta}_h)$ in (6). The main tools we use to minimize $F(\boldsymbol{\theta}_m, \boldsymbol{\theta}_h)$ is the Iteratively Reweighted Least Square (IRLS) method (see [12, Chapter 4.4]) and the Fast Iterative Shrinkage Threshold algorithm (FISTA) (see [1]).

The proposed optimization algorithm is an iterative algorithm with two loops: the outer-loop for updating $\boldsymbol{\theta}_m$ and the inner-loop for updating $\boldsymbol{\theta}_h$. The optimization procedure is

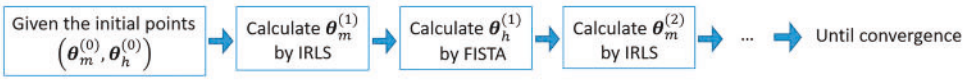

Figure 2. The pipeline of our algorithm to minimize $F(\theta_m, \theta_h)$ with respect to θ_m, θ_h.

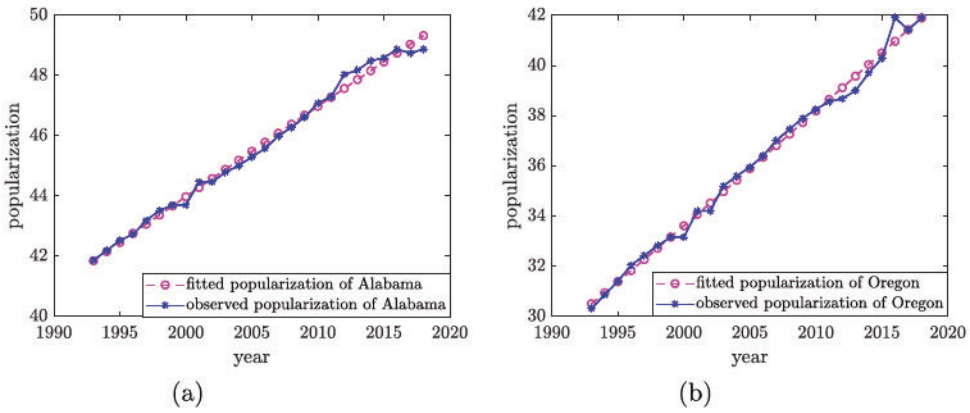

Figure 3. Population and estimate in Alabama and Georgia during 1993–2018. (a) Alabama. (b) Oregon.

visualized in Figure 2 and described as follows. We begin with the initial point $(\theta_m^{(0)}, \theta_h^{(0)})$. Then, in the first outer-loop, we minimize $F(\theta_m, \theta_h^{(0)})$ with respect to θ_m by the IRLS algorithm. In this way, we update θ_m from $\theta_m^{(0)}$ to $\theta_m^{(1)}$. Next, we minimize $F(\theta_m^{(1)}, \theta_h)$ with respect to θ_h by the FISTA algorithm, which allows us to update θ_h from $\theta_h^{(0)}$ to $\theta_h^{(1)}$. The iteration of the FISTA algorithm is called the inner-loop under the first outer-iteration. The above outer-loop and inter-loop will be repeated until the convergence. The detailed pseudo-code is summarized in Algorithm 1.

Algorithm 1: Pseudo-code of the optimization algorithm in our proposed PoSS-TenD method.

Input: $\mathbf{y}, \mathbf{n} \in \mathbb{R}^{n_1 n_2 n_3}, \mathbf{B}_m \in \mathbb{R}^{n_1 n_2 n_3 \times p_1 p_2 p_3}, \mathbf{B}_h \in \mathbb{R}^{n_1 n_2 n_3 \times q_1 q_2 q_3}, \lambda > 0$.
Output: The estimator of θ_m, θ_h, noted as $\widehat{\theta}_m, \widehat{\theta}_h$.

```
Initialization θ_m^(0), θ_h^(0).
Outer Iteration:
    update θ_m from θ_m^(k−1) to θ_m^(k) by the IRLS algorithm
    Inner Iteration:
        update θ_h from θ_h^(k−1) to θ_h^(k) by the FISTA algorithm
    end
end
```

In the reminder of this section, we discuss the detailed implementation of the IRLS algorithm and FISTA algorithm, as shown in the aforementioned Algorithm 1.

First, the implementation of the IRLS algorithm to estimate the parameter of the global trend mean $\boldsymbol{\theta}_m$ is summarized in Proposition 3.1.

Proposition 3.1: *In Algorithm* 1, *given* $(\boldsymbol{\theta}_m^{(k-1)}, \boldsymbol{\theta}_h^{(k-1)})$, *one can update* $\boldsymbol{\theta}_m^{(k)}$ *by IRLS algorithm, i.e.*

$$\boldsymbol{\theta}_m^{(k)} = \left(\mathbf{X}^\top \mathbf{W} \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{W} \underbrace{\left[\mathbf{X}\boldsymbol{\theta}_m^{(k-1)} + \mathbf{W}^{-1}\left(\mathbf{y} - \boldsymbol{\gamma}^{(k-1)}\right)\right]}_{\boldsymbol{\eta}^{(k-1)}}.$$

Here the matrix $\mathbf{X}^\top$ *is of dimension* $p_1p_2p_3 \times n_1n_2n_3$. *And its i-th column, denoted as* $\mathbf{x}_i$, *is the transpose of i-th row of matrix* $\mathbf{B}_m$. *The matrix* $\mathbf{W} \in \mathbb{R}^{n_1n_2n_3 \times n_1n_2n_3}$ *is a diagonal matrix. And its* (i,i)*-th entry is* $n_i e^{(\mathbf{x}_i^\top \boldsymbol{\theta}_m^{(k-1)} + \mathbf{z}_i^\top \boldsymbol{\theta}_h^{(k-1)})}$. *The vector* $\mathbf{y}$ *is of length* $n_1n_2n_3$, *whose i-th entry is* y_i. *The vector* $\boldsymbol{\gamma}^{(k-1)}$ *is of length* $n_1n_2n_3$. *And its i-th entry is* $n_i e^{(\mathbf{x}_i^\top \boldsymbol{\theta}_m^{(k-1)} + \mathbf{z}_i^\top \boldsymbol{\theta}_h^{(k-1)})}$, *where* $\mathbf{z}_i^\top$ *is the i-th row of the matrix* $\mathbf{B}_h$.

Proof: The proof of the above proposition is shown in Appendix 1. ■

Next, we discuss the implementation of the FISTA algorithm to estimate the parameter of the local hot-spots $\boldsymbol{\theta}_h$. Suppose in the k-th iteration, we already get the iterative estimator $\boldsymbol{\theta}_m^{(k)}$ by Proposition 3.1. Given that, we can updating $\boldsymbol{\theta}_h$ from $\boldsymbol{\theta}_h^{(k-1)}$ to $\boldsymbol{\theta}_h^{(k)}$ by the minimizing the following optimization problem with respect to $\boldsymbol{\theta}_h$:

$$F(\boldsymbol{\theta}_m^{(k)}, \boldsymbol{\theta}_h) = \underbrace{\sum_{i=1}^{n}\left[-y_i\left(\mathbf{x}_i^\top \boldsymbol{\theta}_m^{(k)} + \mathbf{z}_i^\top \boldsymbol{\theta}_h\right) + n_i e^{\left(\mathbf{x}_i^\top \boldsymbol{\theta}_m^{(k)} + \mathbf{z}_i^\top \boldsymbol{\theta}_h\right)}\right]}_{g(\boldsymbol{\theta}_h)} + \underbrace{\lambda \left\|\boldsymbol{\theta}_h\right\|_1}_{h(\boldsymbol{\theta}_h)}. \tag{7}$$

Essentially speaking, the above optimization problem is an optimization problem with a ℓ_1 regularization term. There is a large body of literature available on the analysis of it, where both researchers in optimization research and statistics devote to it.

In optimization research, one well-known method is the Iterative Shrinkage Threshold Algorithm (ISTA) proposed by [9], where they approximate $g(\boldsymbol{\theta}_h)$ by its second-order Taylor expansion. For the Hessian matrix, to speed up the algorithm, they approximate it by a diagonal matrix, whose diagonal entry is the maximal eigenvalue of the Hessian matrix. After the quadratic approximation of the objective function, a proximal mapping is used, where the soft-thresholding operator can be easily applied. [9] is a representative of this type of method, and the early foundational work of proximal gradient descent can be found in [2]. As the techniques mature, they became widely used in several different fields. As a result, they have been referred to by a diverse set of names, including proximal algorithm, proximal point, and so on. In the survey of [38], it can be seen that, many other widely-known methods – including, Majorization-Minimization (MM) and Alternating Direction Method of Multipliers (ADMM) – also fall into the proximal framework. Following [1,9] proposes another algorithm called FISTA, which can be regarded as an accelerated version of [9]. The main difference between ISTA and FISTA is that FISTA uses more historical information, i.e. FISTA takes advantage of the gradient of the previous two

estimators, while ISTA only uses the gradient of the previous one estimator. Accordingly, the convergence rate of FISTA is $O(1/k^2)$ iterations, which is faster than $O(1/k)$ achieved by ISTA (see [49]).

In statistics, researchers also do lots of works. One of the well-known methods is proposed by [13] with an R package called *glmnet*. The main tool used is the coordinate descent algorithm. The coordinate descent method has been proposed several times for the optimization with ℓ_1 regularization term, e.g. see discussions in [13] and its convergence rate is $O(1/k)$ (see [49]).

After comparing the existing algorithm to solve (7), we select FISTA in our paper due to its faster convergence. The detailed implementation of the FISTA algorithm can be found in the following proposition.

Proposition 3.2: *In Algorithm* 1, *given* $(\boldsymbol{\theta}_m^{(k)}, \boldsymbol{\theta}_h^{(k-1)})$, *one can update* $\boldsymbol{\theta}_h$ *from* $\boldsymbol{\theta}_h^{(k-1)}$ *to* $\boldsymbol{\theta}_h^{(k)}$ *by the FISTA algorithm. The updating scheme of the FISTA algorithm is summarized as follows. For a fixed* $k > 0$, *and any* $s = 0, 1, \ldots$, *one has*

$$\begin{aligned}
\boldsymbol{\theta}_h^{(k)[s]} &= S\left(\boldsymbol{\alpha}^{[s]} - \frac{\partial}{\partial \boldsymbol{\theta}_m} g(\boldsymbol{\theta}_m^{(k)}, \boldsymbol{\alpha}^{[s]})/L, \lambda/L\right) \\
t_{s+1} &= \left(1 + \sqrt{1 + 4t_s^2}\right)/2 \\
\boldsymbol{\alpha}^{[s+1]} &= \boldsymbol{\theta}_h^{(k)[s]} + \frac{t_s - 1}{t_{s+1}}\left(\boldsymbol{\theta}_h^{(k)[s]} - \boldsymbol{\theta}_h^{(k)[s-1]}\right),
\end{aligned}$$

where $\boldsymbol{\theta}_h^{(k)[s]}$ *is the iterative estimator of* $\boldsymbol{\theta}_h$ *after s inner loops under the k-th outer loops. The function* $S(\cdot, \cdot)$ *is a thresholding function, i.e.* $S(x, a) = (x - a)\mathbb{1}\{x \geq a\} + (x + a)\mathbb{1}\{x \leq -a\}$ *for any* $x \in \mathbb{R}$ *and* $a > 0$. *The vector* $\boldsymbol{\alpha}^{[s]}$ *is an auxiliary vector and is initialized as* $\boldsymbol{\theta}_h^{(k-1)}$. *The hyper parameter* t_s *and L can be initialized as* $t_1 = 1$ *and set as the maximal eigenvalue of the matrix* $\frac{\partial^2}{\partial \boldsymbol{\theta}_h \partial \boldsymbol{\theta}_h^\top} g(\boldsymbol{\theta}_h)$, *respectively.*

By combining Propositions 3.1 and 3.2, we update $\boldsymbol{\theta}_m$ and $\boldsymbol{\theta}_h$ iteratively until convergence. We summarize the details of our algorithm for parameter estimation as follows:

3.3. Hot-spots detection and localization

In this subsection, we discuss the detection and localization of the hot-spots. For the ease of presentation, we first discuss the hot-spots detection, i.e. detect when a hot-spot occurs in Subsection 3.3.1. Then, in Subsection 3.3.2, we consider the localization of the hot-spots, i.e. determine which states and which type of diseases are involved for the detected hot-spots.

3.3.1. Hot-spots detection

To detect when the hot-spots occur, we propose to monitor the time series of the Pearson residuals of count data, $(y_t(\lambda) - \widehat{\mu}_t(\lambda))/\sqrt{\mu_t(\lambda)}$, over time $t = 1, \ldots, n_3$. If it has a shift toward the direction of the estimated hot-spots $\mathbf{h}_t$, then it is highly likely that there

Algorithm 2: Estimation of $\boldsymbol{\Theta}_m, \boldsymbol{\Theta}_h$ by IRLS & FISTA algorithm.

Input: $\mathbf{y}, \mathbf{n} \in \mathbb{R}^{n_1 n_2 n_3}$, $\mathbf{B}_m \in \mathbb{R}^{n_1 n_2 n_3 \times p_1 p_2 p_3}$, $\mathbf{B}_h \in \mathbb{R}^{n_1 n_2 n_3 \times q_1 q_2 q_3}$, $\lambda > 0, t_1$.

Output: The estimators of $\boldsymbol{\Theta}_m, \boldsymbol{\Theta}_h$, noted as $\widehat{\boldsymbol{\Theta}}_m, \widehat{\boldsymbol{\Theta}}_h$.

Initialization $t_1 = 1, \boldsymbol{\Theta}_m^{(0)}, \boldsymbol{\Theta}_h^{(0)}$.

▶ ***Outer-Iteration:*** ◀ **for** $k = 0, 1, 2, 3, \ldots, K$ **do**

$\mathbf{W} = diag(e^{\mathbf{x}_1^\top \boldsymbol{\Theta}_m^{(k)} + \mathbf{z}_1^\top \boldsymbol{\Theta}_h^{(k)}}, \ldots, e^{\mathbf{x}_{n_1 n_2 n_3}^\top \boldsymbol{\Theta}_m^{(k)} + \mathbf{z}_{n_1 n_2 n_3}^\top \boldsymbol{\Theta}_h^{(k)}})$

$\boldsymbol{\gamma}^{(k)} = (e^{\mathbf{x}_1^\top \boldsymbol{\Theta}_m^{(k)} + \mathbf{z}_1^\top \boldsymbol{\Theta}_h^{(k)}}, \ldots, e^{\mathbf{x}_{n_1 n_2 n_3}^\top \boldsymbol{\Theta}_m^{(k)} + \mathbf{z}_{n_1 n_2 n_3}^\top \boldsymbol{\Theta}_h^{(k)}})^\top$

$\boldsymbol{\eta}^{(k)} = \mathbf{X}\boldsymbol{\Theta}_m^{(k)} + \mathbf{W}^{-1}(\widetilde{\mathbf{y}} - \boldsymbol{\gamma}^{(k)})$

$\boldsymbol{\Theta}_m^{(k+1)} = (\mathbf{X}^\top \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W} \boldsymbol{\eta}^{(k)}$

$\boldsymbol{\beta}^{(k)[0]} = \boldsymbol{\Theta}_h^{(k)}$

$\boldsymbol{\alpha}^{[1]} = \boldsymbol{\Theta}_h^{(k)}$

$t_1 = 1$

▶ ***Inner-Iteration:*** ◀ **for** $s = 1,2, \ldots, S$ **do**

$\boldsymbol{\Theta}_h^{(k)[s]} = S(\boldsymbol{\alpha}^{[s]} - \frac{\partial}{\partial \boldsymbol{\Theta}_m} g(\boldsymbol{\Theta}_m^{(k+1)}, \boldsymbol{\alpha}^{[s]})/L, \lambda/L)$

$t_{s+1} = (1 + \sqrt{1 + 4t_s^2})/2$

$\boldsymbol{\alpha}^{[s+1]} = \boldsymbol{\Theta}_h^{(k)[s]} + \frac{t_s - 1}{t_{s+1}} (\boldsymbol{\Theta}_h^{(k)[s]} - \boldsymbol{\Theta}_h^{(k)[s-1]})$

end

$\boldsymbol{\Theta}_h^{(k+1)} = \boldsymbol{\Theta}_h^{(k)[S]}$

end

$\widehat{\boldsymbol{\Theta}}_m = \boldsymbol{\Theta}_m^{(K)}$

$\widehat{\boldsymbol{\Theta}}_h = \boldsymbol{\Theta}_h^{(K)}$

is a hot-spot. Mathematically speaking, we develop a control chart based on the following hypothesis test problem:

$$H_0 : \mathbf{r}_t(\lambda) = 0 \text{ v.s. } H_1 : \mathbf{r}_t(\lambda) = \delta \widehat{\mathbf{h}}_t(\lambda) \quad (\delta > 0), \tag{8}$$

where $\mathbf{r}_t(\lambda) = \mathbf{y}_t(\lambda) - \widehat{\boldsymbol{\mu}}_t(\lambda)$ is the residual after removing the global trend mean under the penalty parameters λ and the vector $\widehat{\boldsymbol{\mu}}_t(\lambda) = \text{vec}(\widehat{\mathcal{U}}_{::t}(\lambda)), \widehat{\mathbf{h}}_t(\lambda) = \text{vec}(\widehat{\mathcal{H}}_{::t}(\lambda))$ are the estimated global trend men and local hot-spots in t-th year. Here, we add (λ) to emphasize that, $\widehat{\boldsymbol{\mu}}_t(\lambda), \widehat{\mathbf{h}}_t(\lambda)$ are the global trend mean and local hot-spots estimation under penalty parameter λ respectively.

The motivation of the above hypothesis test is articulated as follows. When there are no hot-spots, the residual $\mathbf{r}_t(\lambda)$ is exactly the model noises. However, when hot-spots exist, the residual $\mathbf{r}_t$ includes both hot-spots and noises. By including the hot-spots information of $\widehat{\mathbf{h}}_t(\lambda)$ in the alternative hypothesis, we hope to provide a direction in the alternative hypothesis space, which allows one to construct a test with more power (see [53]).

Next, we construct the likelihood ratio test in the above-mentioned hypotheses testing problem. By [17], the test statistics monitoring upward shift is

$$P_t^+(\lambda) = \widehat{\mathbf{h}}_t^+(\lambda)^\top \mathbf{r}_t(\lambda) \Big/ \sqrt{\widehat{\mathbf{h}}_t^+(\lambda)^\top \widehat{\mathbf{h}}_t^+(\lambda)}$$

where $\widehat{\mathbf{h}}_t^+(\lambda)$ only takes the positive part of $\widehat{\mathbf{h}}_t(\lambda)$ with other entries as zero, because our objective is to detect positive hot-spots. The superscript '+' emphasizes that we aim at detecting upward shift. In other words, we focus on the hot-spots that have increasing means, partly because increasing infectious rates are generally more harmful to the societies and communities. If one is also interested in detecting decreasing mean shifts, one could modify it by using a two-sided test.

It remains to discuss how to choose λ suitably in our test. We propose to follow [53] to calculate a series of $P_t^+(\lambda)$ under different combination of $\lambda \in \Gamma = \{\lambda^{(1)}, \ldots, \lambda^{(n_\lambda)}\}$ and then select the λ with the largest power. The final chosen test statistics, denoted as $\widetilde{P}_t^+(\lambda_t^*)$, can be computed by

$$\widetilde{P}_t^+(\lambda_t^*) = \max_{\lambda \in \Gamma} \frac{P_t^+(\lambda) - E(P_t^+(\lambda))}{\sqrt{\mathrm{Var}(P_t^+(\lambda))}}, \tag{9}$$

where $E(P_t^+(\lambda))$, $\mathrm{Var}(P_t^+(\lambda))$ respectively are the mean and variance of $P_t^+(\lambda)$ under H_0 (e.g. for phase-I in-control samples). Here $\lambda_t^* \in \Gamma$ is the penalty parameter maximizing the above equation.

With the test statistic available, we detect when hot-spots occur based on the widely used Cumulative Sum (CUSUM) Control Chart (see [30,35]). At each time t, we recursively compute the CUSUM statistics as

$$W_t^+ = \max\{0, W_{t-1}^+ + \widetilde{P}_t^+(\lambda_t^*) - d^*\}, \tag{10}$$

with the initial value $W_{t=0}^+ = 0$, where d^* is a constant and can be chosen according to the degree of the shift that we want to detect. Then we declare that a hot-spot might occurs whenever $W_t^+ > L$ for some pre-specified control limit L.

Note that the CUSUM statistics W_t^+ leads to the optimal control chart to detecting a mean shift from μ_0 to $\mu_1 = 2d^* - \mu_0$ for normally distributed data (see [30]). When the data are not normally distributed, the optimality properties might not hold, but it can still be a reasonable control chart. Also, it is important to choose the control limit L in the CUSUM control chart suitably, and the detailed discussion will be presented in Section 4 for our simulation studies and in Section 5 for our case study.

3.3.2. Hot-spots localization

In this subsection, we discuss how to localize the hot-spots if the CUSUM control chart in (10) raises an alarm at year t^*. In other words, we want to determine where and which infectious rates may account for the hot-spots. To do so, we propose to utilize the matrix $\widehat{\mathcal{H}}_{::t^*}(\lambda_{t^*})$, which is the hot-spots estimation in t^*-th year. If the (i, j, t^*)-th entry in $\widehat{\mathcal{H}}_{::t^*}(\lambda_{t^*})$ is non-zero, then we declare that there is a hot-spot for the j-th type of disease in the i-th state at the t^*-th year.

The mathematical procedure to derive $\widehat{\mathcal{H}}_{::t^*}(\lambda_{t^*})$ is as follows. First, $\widehat{\mathcal{H}}(\lambda_{t^*})$ is the tensor format of $\widehat{\mathbf{h}}(\lambda_{t^*}) = \mathbf{B}_h \widehat{\boldsymbol{\theta}}_h(\lambda_{t^*})$, where $\widehat{\boldsymbol{\theta}}_h(\lambda_{t^*})$ is the minimizer in (6) with penalty parameter as λ_{t^*}. Second, $\widehat{\mathcal{H}}_{::t^*}(\lambda_{t^*})$ is the t^*-th frontal slices of $\widehat{\mathcal{H}}(\lambda_{t^*})$.

It is possible that this approach might lead to a relatively high false positive rate (FPR), since some non-zero entries might not be statistically significant. Two possible ways to improve our approach are (1) to conduct the significant test, or (2) to set up a pre-specified threshold and only keep the positive entries that are larger than the threshold. In our paper,

we use the second approach, i.e. set up a pre-specified threshold for hot-spots localization. As for the selection of the threshold, there are several choices:

- Hard-thresholding: if the (i,j,t^*)-th entry in the refined estimation of hot-spots $\widehat{\mathcal{H}}_{::t^*}(\lambda_{t^*}) = \widehat{\mathcal{H}}_{::t^*}(\lambda_{t^*})\mathbb{1}\{\widehat{\mathcal{H}}_{::t^*}(\lambda_{t^*}) > h_{\text{hard}}\}$ is larger than zero, then we declare there is a hot-spot in the j-th type of disease in the i-th state at the t^*-th year.
- Soft-thresholding: if the (i,j,t^*)-th entry in the refined estimation of hot-spots $\widehat{\mathcal{H}}_{::t^*}(\lambda_{t^*}) = \max\{\widehat{\mathcal{H}}_{::t^*}(\lambda_{t^*}) - h_{\text{soft}}, 0\}$ is larger than zero, then we declare there is a hot-spot in the j-th type of disease in the i-th state at the t^*-th year.
- Order-thresholding: if the (i,j,t^*)-th entry in the refined estimation of hot-spots $\widehat{\mathcal{H}}_{::t^*}(\lambda_{t^*}) = \widehat{\mathcal{H}}_{::t^*}(\lambda_{t^*})\mathbb{1}\{\widehat{\mathcal{H}}_{::t^*}(\lambda_{t^*}) \geq h_{\text{order},r}\}$ is larger than zero, then we declare there is a hot-spot in the j-th type of disease in the i-th state at the t^*-th year. Here $h_{\text{order},r}$ is the r-th largest order statistics of $\{\widehat{\mathcal{H}}_{i,j,t^*}(\lambda_{t^*}) : \forall(i,j),\ s.t.\ \widehat{\mathcal{H}}_{i,j,t^*}(\lambda_{t^*}) > 0\}$.

In this paper, for simplicity, we use the order-thresholding as an illustration, since it has been used in other contents, such as outlier detection (see [31]), testing problem (see [22]), etc..

4. Monte Carlo simulations

In this section, we report the numerical simulation results of our proposed method as well as its comparison with several benchmark methods in the literature. To better present our results, we divide this section into several subsections. Subsection 4.1 includes the data generation mechanism for our simulation studies, and Subsection 4.2 presents the benchmark methods for the comparison purpose. The performance of hot-spots detection and localization are reported in Subsection 4.3.

4.1. Data generation mechanism

This subsection describes the data generation mechanism we use in the numerical studies. In our simulation, the (i,j,t)-th entry of tensor $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is generated as follows:

$$\begin{cases} \mathcal{Y}_{i,j,t} \sim \text{Poisson}(\mathcal{N}_{i,j,t}\mathcal{R}_{i,j,t}) \\ \mathcal{R}_{i,j,t} = 0.2 + \delta\mathbb{1}\{t > \tau\}\,\mathbb{1}\left\{(i,j) \in S\right\} \end{cases}. \tag{11}$$

In the first line of the above generative model, $\mathcal{Y}_{i,j,t}, \mathcal{N}_{i,j,t}, \mathcal{R}_{i,j,t}$ is the (i,j,t)-th entry of tensor $\mathcal{Y}, \mathcal{N}, \mathcal{R} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, whose physical meaning is the number of infected patients, the population size, and the infectious rate in the i-th state in year t under the j-th type of disease, respectively. In this simulation, to match the dimension of or motivating dataset, we set $n_1 = 49, n_2 = 10, n_3 = 26$.

In the second line of the above generative model in (11), the parameter δ measures the magnitude of the hot-spots. In our paper, we use $\delta = 0.05$ as the small hot-spots and $\delta = 0.2$ as the large hot-spots. The reasons we declare $\delta = 0.05$ ($\delta = 0.2$) as small (large) hot-spots is that, the average Kullback-Leibler divergence (see [24]) is 0.3474 (4.6388). And $\mathbb{1}\{x \in A\}$ is an indictor function, i.e. $\mathbb{1}\{x \in A\} = 1$ if $x \in A$ and $\mathbb{1}\{x \in A\} = 0$ otherwise. The first indictor function $\mathbb{1}\{t > \tau\}$ means that the hot-spots only happens after the τ-th

year. And in our paper, we set $\tau = 15$. The second indictor function $\mathbb{1}\{(i,j) \in S\}$ means that the hot-spots only occurs in certain spatial location set S. In our paper, S is randomly sampled without replacement in each simulation and the number of elements of S only accounts for 10% of the total number of elements of $\vartheta_{h,:,:,t}$.

It remains to discuss how to model the population size $\mathcal{N}_{i,j,t}$. We propose to follow the common practice to model the curve of the population growth by the logistic model (see [37]):

$$\mathcal{N}_{i,j,t} = \frac{\phi_{i,j,1}}{1 + \exp[-(t - \phi_{i,j,2})/\phi_{i,j,3}]} + \epsilon_{i,j,t} \tag{12}$$

where $E(\epsilon_{i,j,t}) = 0$ and $Var(\epsilon_{i,j,t}) = \sigma^2$. Here $\phi_{i,j,1}$ indicates an asymptotic upper limit of population size in the i-th state, $\phi_{i,j,2}$ is the middle point of the S-shaped curve in the i-th state, and $\phi_{i,j,3}$ is the scale adjustment of time periods in the i-th state.

In our numerical studies, we consider two scenarios for the population size $\mathcal{N}$: (1) the population size with increasing trend and (2) the population size with decreasing trend. In the above two scenarios, we assume one state share the uniform characteristic in the population over different type of diseases, i.e. $\phi_{i,j_1,1} = \phi_{i,j_2,1}, \phi_{i,j_1,2} = \phi_{i,j_2,2}, \phi_{i,j_1,3} = \phi_{i,j_2,3}$ and $\mathcal{N}_{i,j_1,t} = \mathcal{N}_{i,j_2,t}$ for any $j_1 \neq j_2$. In our paper, we first fit (12) to the observed population sizes in our motivating dataset by a nonlinear least-squares method (we treat year 1993 as time 1, and the population sizes are in the units of 10,000). Using the mathematical software Matlab R2018b, the estimated parameters for the logistic model (12) are summarized in Appendix 2. Figure 3 plots the actual observed population sizes and the estimated growth curve in New Mexico during 1993-2018. From the plot, one sees that the two curves are close to each other, implying that the logistic model is reasonable in our application.

In our simulation, we will consider the following two extreme scenarios of the population sizes, depending on whether the population size in all states are increasing or decreasing. This might be unrealistic, but can indicate the performance in more practical scenarios, where the population can be the interested study group, such as specific age/gender/race sub-groups.

- There is an increasing trend in the population. The increasing population is generated from $\mathcal{N}_{i,j,t} = \frac{\hat{\phi}_{i,j,1}}{1+\exp[-(t-\hat{\phi}_{i,j,2})/\hat{\phi}_{i,j,3}]} + \epsilon_{i,j,t}$, where $\{(\hat{\phi}_{i,j,1}, \hat{\phi}_{i,j,2}, \hat{\phi}_{i,j,3})\}$ is estimated by fitting the real-world population into (12).
- There is a decreasing trend in the population. The decreasing population is generated from $\mathcal{N}_{i,j,t} = \frac{\hat{\phi}_{i,j,1}/a_{i,j}}{1+\exp[(t-\hat{\phi}_{i,j,2})/\hat{\phi}_{i,j,3}]} + 1 + \epsilon_{i,j,t}$, where $\{(\hat{\phi}_{i,j,1}, \hat{\phi}_{i,j,2}, \hat{\phi}_{i,j,3})\}$ is estimated by fitting the real-world population into (12). Here the parameter $\{a_{i,j}\}$ is necessary to make sure that the simulated initial population size is the same as the observed value.

4.2. Benchmark methods

In this subsection, we present the description and implement of five benchmark or baseline methods that will be used to compare with our method.

The first benchmark method is the scan statistics method in [34], which is a Bayesian extension of Kulldorff's scan statistic and we denote it as 'NMC-scan-stat'. The reason for

us to choose NMC-scan-stat is that it has large power to detect clusters and the fast runtime [34]. In our paper, we use the R function called *scanbayesnegbin()* from the package *scanstatistics*. Because *scanbayesnegbin()* can only handle one type of disease one time, we apply *scanbayesnegbin()* to ten diseases separately and set the probability of an outbreak as 0.02/10. Because scan-statistics-based method does not give the clear calculation of *average run length under the in-control status* (ARL_0) and ARL_1, so we can only use the probability of an outbreak as 0.02/10 to define the control limit to achieve similar ARL_0 with other benchmarks.

The second benchmark method is the smooth sparse decomposition method proposed by [46], and we denote it as 'YPS-SSD'. The YPS-SSD method assumes the raw data (or the raw data after transformation) follows normal distribution and decompose it into the functional mean, sparse anomalies, and random noises. Then YPS-SSD detect the hot-spots by monitoring the model residual through Shewhart control chart. Although the main idea in YPS-SSD looks similar to our proposed PoSSTenD, there are two significant difference. The first difference is that, YPS-SSD focus the normal distributed tensor, while our PoSSTenD method focus on the Poisson distributed tensor. In other words, YPS-SSD ignore the effect of the population size, and aims at detecting the hot-spots in the number of infected events. However, for our PoSSTenD method, we take the population size into consideration and target on detecting the hot-spots in infectious rates. The second difference is that, YPS-SSD utilizes Shewhart control chart to monitor the hot-spots, which is not sensitive to the small hot-spots. Yet, our proposed PoSSTenD method uses CUSUM control chart, which has more capability to detect small hot-spots. For the selection of basis in YPS-SSD and our PoSSTenD method, we use the same basis for a fair comparison, i.e. $\mathbf{B}_{m,1} \in \mathbb{R}^{n_1 \times p_1}$, $\mathbf{B}_{m,2} \in \mathbb{R}^{n_2 \times p_2}$, $\mathbf{B}_{m,3} \in \mathbb{R}^{n_3 \times p_3}$ are B-spline basis with order 3 and over equally spaced knots $\{1, 8, 15, \ldots, 50\}$, $\{1, 9.1667, 17.3333, \ldots, 50\}$, $\{1, 9.1667, 17.3333, \ldots, 50\}$, respectively (see [19]) and $\mathbf{B}_{h,1}, \mathbf{B}_{h,2}, \mathbf{B}_{h,3}$ are all set as identity matrix.

The third benchmark method is the Lasso-based method proposed by [53] and we denote it as 'ZQ-Lasso'. The main idea of ZQ-Lasso is to integrate the multivariate Exponentially Weighted Moving Average (EWMA) charting scheme. Under the assumption that the hot-spots are sparse, the Lasso model is applied to the EWMA statistics. If the Mahalanobis distance between the expected response (the Lasso estimator) and observed values is larger than a pre-specified control limit, temporal hot-spots are detected, with non-zero entries of the Lasso estimator are declared as spatial hot-spots. For the control limits and the penalty parameters of the Lasso-based method, we use the same criterion as our proposed PoSSTenD method.

The fourth benchmark is the dimension-reduction method proposed by [10], and we denote it as 'DBS-PCA'. The DBS-PCA method uses PCA to extract a set of uncorrelated new features that are linear combinations of original variables. Note that DBS-PCA fails to localize the spatial hot-spots, and it can only detect the temporal change-point when the PCA-projected Mahalanobis distance is larger than a pre-specified control limit. For this control limit, we set it by using the same criterion as our proposed PoSSTenD method. In both our simulations and case study, we select three principle components, since they can explain more than 90% cumulative percentage of variance (CPV).

Finally, the fifth benchmark is the traditional T^2 control chart (see [39]) method and we denote it as 'T^2'. The control limit of T^2 is set by using the same criterion as our proposed

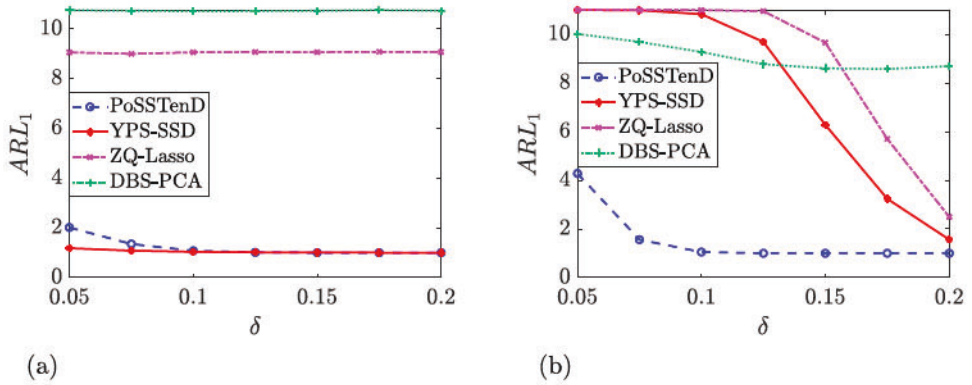

Figure 4. The ARL_1 plot of our proposed PoSSTenD method, and the three benchmark methods, i.e. YPS-SSD method, ZQ-Lasso method and DBS-PCA method. (a) population with an increasing trend. (b) population with a decreasing trend.

PoSSTenD method. Since the T^2 control chart method is a well-defined method, we skip the detailed description, and more details can be found in [39].

4.3. Simulation results

In this subsection, we compare our proposed PoSSTenD method with five benchmark methods with the focus on the performance of hot-spots detection in Section 4.3.1 and localization 4.3.2. The five benchmark methods are NMC-scan-stat method proposed by [34], YPS-SSD proposed by [46], ZQ-Lasso proposed by [53], DBS-PCA proposed by [10], and the traditional Hotelling T^2 control chart reviewed in [39]. All simulation results below are based on 1000 Monte Carlo replications.

4.3.1. Comparison of hot-spots detection

In this section, we compare the performance of hot-spots detection, i.e. the detection delay after the occurrence of hot-spots. The criterion we use is ARL_1. Because ARL_1 measures the delay after the change occurs, the smaller the ARL_1, the better detection performance. The comparison results are visualized in Figure 4 (the numerical numbers to generate this plot are available in Appendix 3).

For our proposed PoSSTenD method (marked with blue), it has a small ARL_1 no larger than 4.2780 under both two scenarios. This indicates that the PoSSTenD method can provide a rapid alarm after the hot-spots occur, even if there are increasing or decreasing global trends. This good performance is mainly due to its ability to remove the global trend and capture the small hot-spots by the CUSUM control chart.

For the NMC-scan-stat method, it is hard to estimate the exact ARL_1 because it focuses on the hot-spots localization, not sequential change point detection. So we will not report its ARL_1.

For the YPS-SSD method (marked with red), it can successfully detect hot-spots when there is an increasing trend in population. For example, its ARL_1 is around 2 when $\delta = 0.05$ and there is an increasing trend in population. However, when the hot-spots are small and there is a decreasing trend in population, then YPS-SSD has very limited power to detect

the hot-spots. For example, its ARL_1 is around 11 when $\delta = 0.05$ and there is a decreasing trend in population. If $ARL_1 \approx 11$, it is bad, because it means the YPS-SSD method is unable to detect any hot-spots among a total of 26 years with hot-spots occurrence after the 15-th year. This is because YPS-SSD targets on the detection of hot-spots among the number of infectious people, rather than the infectious rates. So when the hot-spots are small and there is a decreasing trend in population, the numbers of infectious people have a very stable trend, which makes it difficult for YPS-SSD to detect the hot-spots.

For ZQ-Lasso (marked with pink), it has a relatively larger ARL_1. For example, its ARL_1 is around 9 when there is an increasing trend in the population. And its ARL_1 is above 9 when *delta* ≤ 0.15 and there is a decreasing trend in the population. This is not surprising because ZQ-Lasso is unable to separate the global trend and local hot-spots.

For the DBS-PCA method (marked with green), its ARL_1 is also large compared with other methods. Specifically, its ARL_1 is above 8 for both two scenarios. And the T^2 method fails to detect the hot-spots in all scenarios within the entire $n_3 = 26$ (simulated) years. The reason for the unsatisfying results of DBS-PCA and T^2 is that they are designed based on the global mean change, which cannot take into account the non-stationary global mean trend and the sparsity of the hot-spots.

In conclusion, our proposed PoSSTenD method has short detection delay, no matter there is increasing/decreasing population trend. Yet, for the other benchmarks, their detection performance is not robust to the population trend. Especially when there is a decreasing population trend, they all have relatively long detection delays.

4.3.2. Comparison of hot-spots localization

In this section, we compare the performance on hot-spots localization. The evaluation criterion are: (1) precision, defined as the proportion of detected hot-spots that are true hot-spots; (2) recall, defined as the proportion of the hot-spots that are correctly identified; (3) F-measure, a single criterion that combines the precision and recall by calculating their harmonic mean. In this paper, we classify a overall good method if it has a high precision, together with a high recall and a high F-measure. The localization performance of the compared six methods is available in Table 2.

For our proposed PoSSTenD method, its localization performance is satisfactory no matter the population size is increasing or decreasing over time. For instance, when the population size is decreasing over time and $\delta = 0.2$, our method has 81.13% precision, 54.52% recall, and 64.41% F-measure, which outperforms the benchmark methods. The only weakness of our proposed PoSSTenD method is that it has a relatively high false positive rate when δ is small. For example, its precision, recall, and F-measure are 46.10%, 15.55% and 23.25% when $\delta = 0.05$ and the population is increasing. Though the above performance is not excellent, it is still satisfying given the challenging to detect sparse hot-spots among high-dimensional data, let along there are time-varifying global trend mean.

For the NMC-scan-stat method, it has much lower precision than our proposed PoSSTenD method. For example, its precision, recall and F-measure are 2.50%, 26.69% and 4.51% when $\delta = 0.2$ and the population is decreasing. Its limited power is mainly caused by the property of scan statistics, which tends to detect clustered hot-spots. Accordingly, the NMC-scan-stat method detects fewer hot-spots and misses true hot-spots. It is worth noting that the precision/recall/F-measure might be overestimated for the NMC-scan-stat

Table 2. Hot-spots detection and localization performance by five methods: our proposed method, NMC-scan-stat, YPS-SSD, ZQ-Lasso, DBS-PCA and T^2 under the population with neural growth and positive hot-spots.

	Population with increasing trend				Population with decreasing trend			
	$\delta = 0.05$							
Methods	Precision	Recall	F-measure	ARL_1	Precision	Recall	F-measure	ARL_1
PoSSTenD	0.4610	**0.1555**	**0.2325**	2.0150	**0.3393**	**0.1945**	**0.2470**	**4.2780**
	(0.0775)	(0.0257)	(0.0381)	(1.4610)	(0.1638)	(0.0937)	(0.1187)	(3.6105)
NMC-scan-stat	0.0113	0.1193	0.0203	–	0.0110	0.1134	0.0198	–
	(0.0144)	(0.1655)	(0.0256)	(-)	(0.0141)	(0.1575)	(0.0253)	(-)
YPS-SSD	0.4184	0.1176	0.1836	**1.1806**	0.0000	0.0000	0.0000	11.0000
	(0.0678)	(0.0190)	(0.0296)	(0.5138)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
ZQ-Lasso	**0.9926**	0.1003	0.1822	9.0609	0.0000	0.0000	0.0000	11.0000
	(0.0168)	(0.0013)	(0.0023)	(1.0593)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
DBS-PCA	–	–	–	10.7515	–	–	–	10.0200
	–	–	–	(0.6330)	–	–	–	(2.0995)
T^2	–	–	–	11.0000	–	–	–	11.0000
	–	–	–	(0.0000)	–	–	–	(0.0000)
	$\delta = 0.2$							
PoSSTenD	**0.7530**	**0.3165**	**0.4453**	**1.0000**	**0.8113**	**0.5352**	**0.6441**	**1.0000**
	(0.0614)	(0.0269)	(0.0351)	(0.0000)	(0.0539)	(0.0428)	(0.0420)	(0.0000)
NMC-scan-stat	0.0211	0.2289	0.0381	–	0.0250	0.2669	0.0451	–
	(0.0162)	(0.1985)	(0.0287)	(-)	(0.0176)	(0.2034)	(0.0312)	(-)
YPS-SSD	0.5664	0.1617	0.2515	1.0010	0.9376	0.2436	0.3866	1.5630
	(0.0678)	(0.0194)	(0.0299)	(0.0316)	(0.2136)	(0.0563)	(0.0889)	(2.1750)
ZQ-Lasso	0.9971	0.1011	0.1836	9.0730	0.8990	0.0899	0.1635	2.4990
	(0.0083)	(0.0016)	(0.0027)	(1.0406)	(0.3015)	(0.0301)	(0.0548)	(2.9734)
DBS-PCA	–	–	–	10.7330	–	–	–	8.6920
	–	–	–	(0.7939)	–	–	–	(3.6631)
T^2	–	–	–	11.0000	–	–	–	11.0000
	–	–	–	(0.0000)	–	–	–	(0.0000)

[a] The above results are based on 1000 simulations.

method: we record all the Monte Carlo runs, even if it is a false alarm because scan-stat fails to report ARL_1.

For the YPS-SSD method, it has a high false positive rate in all scenarios, especially when the magnitude of hot-spots is small ($\delta = 0.05$). For example, its recall are 24.36% when $\delta = 0.2$ and the population is decreasing. Besides, we notice that its precision, recall, and F-measure are all zero when $\delta = 0.05$ and there is a decreasing trend in population. This is because that the YPS-SSD method fails to detect when the hot-spots occur, and thus fails to further localize where the hot-spots occur.

For the ZQ-Lasso method, it has high precision when the population is increasing, or $\delta = 0.2$. However, it also has high false positive rate. For example, its precision, recall, F-measure are 99.71%, 10.11% and 18.36 when $\delta = 0.2$ and the population is increasing. We don't think its performance is overall good, since it has unsatisfactory performance in the sense of small recall. The physical meaning of the 99.71%-precision and 10.11%-recall is that, the ZQ-Lasso method detects almost 'everywhere' as hot-spots, so it has around 100% precision. However, as shown in Section 4.1, there are only 10% of them are true hot-spots, so the recall is 10%. In practice, it is not preferred because it gives too much false alarm. The aforementioned unsatisfactory performance is due to two reasons. First, ZQ-Lasso lacks the ability to separate the global trend mean and local hot-spots, so ZQ-Lasso tends

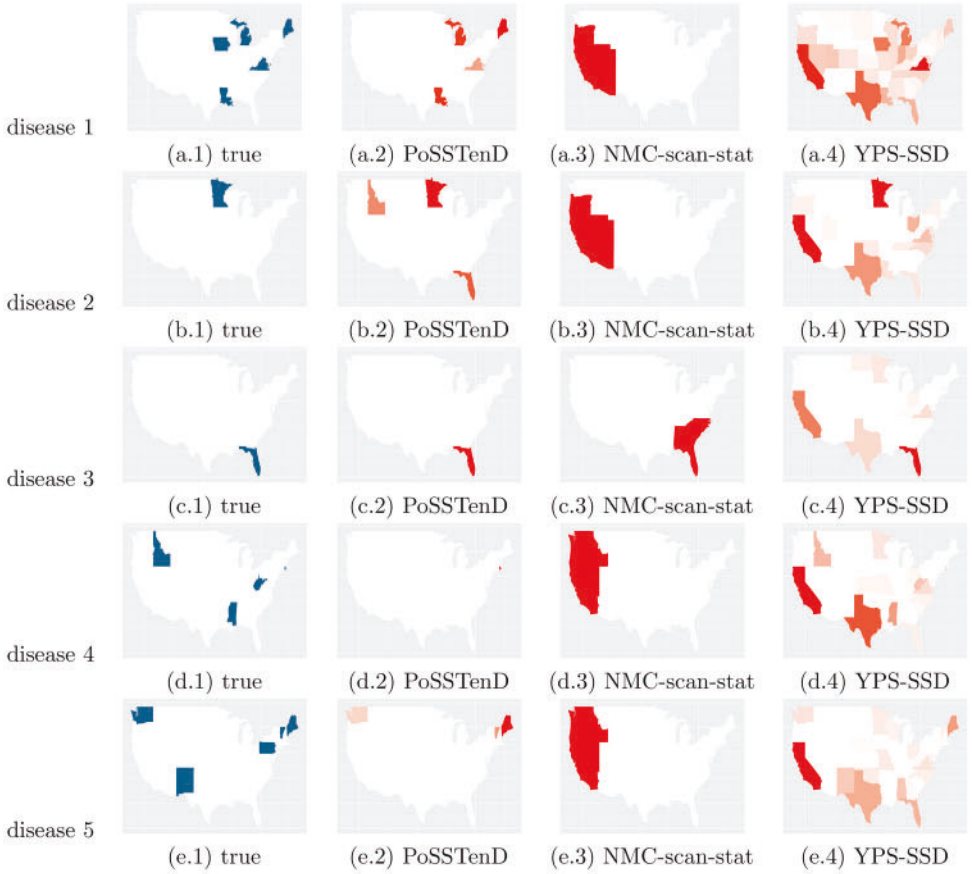

Figure 5. Comparison of true hot-spots and hot-spots detected by our PoSSTenD method and YPS-SSD method under the decreasing population size and large hot-spots $\delta = 0.2$. (a.1) true. (a.2) PoSSTenD. (a.3) NMC-scan-stat & (a.4) YPS-SSD (b.1) true. (b.2) PoSSTenD. (b.3) NMC-scan-stat. (b.4) YPS-SSD (c.1) true. (c.2) PoSSTenD. (c.3) NMC-scan-stat. (c.4) YPS-SSD (d.1) true. (d.2) PoSSTenD. (d.3) NMC-scan-stat. (d.4) YPS-SSD (e.1) true & (e.2) PoSSTenD. (e.3) NMC-scan-stat. (e.4) YPS-SSD.

to localize all states as hot-spots which cause lots of false alarm. Second, ZQ-Lasso aims at hot-spots localization of the number of infectious people, instead of infectious rates. Thus, when there is an increasing trend of the population, ZQ-Lasso mistakenly regards it as the potential hot-spots in the number of infectious people.

For the DBS-PCA and T^2 methods, their localization performance is not reported since they can only detect when hot-spots occur and are incapable to localize where hot-spots occur.

Moreover, we visualize the hot-spots localization results from one Monte Carlo simulation in Figure 5 under the scenario when $\delta = 0.2$ and population is decreasing over time. Different rows refer to different types of disease. And we select the first five disease as representative. In the first column, the blue states are the true hot-spots (true positive), whereas the white states are the normal states (true negative). In the second, third, and fourth columns, the red states are the detected hot-spots (true positive + false positive) by

our proposed PoSSTenD method, NMC-scan-stat, and YPS-SSD method respectively. Different color represents how likely it is hot-spots: the darker red, the more likely it is. From Figure 5, we can see that the NMC-scan-stat method tends to detect clustered hot-spots, however, there is no clear pattern for the hot-spots detected by our proposed PoSSTenD method and YPS-SSD method. The difference between our proposed PoSSTenD method and the YPS-SSD method is that we have a lower false positive rate and a higher true positive rate. This is because that, our proposed PoSSTenD method is designed to detect and localize hot-spots in the infectious rate, which is aligned with the data generation mechanism introduced in Section 4.1. However, the YPS-SSD method detects and localizes hot-spots in the number of infectious people, which leads to an inaccurate localization in our simulations.

In conclusion, our proposed PoSSTenD method has satisfying localization performance no matter the population is increasing or decreasing. For the other benchmarks, they either have relatively high false positive rate, or high false negative rate, or lack the ability to localize the hot-spots. Besides, they are also not robust to the population trend, especially when there is an increasing population trend.

4.3.3. Validation of model fitness

In this section, we demonstrate that our proposed PoSSTenD method leads to a reasonably well estimation for the global trend mean. The criterion we use is the Squared-Root of Mean Square Error (SMSE). Table 3 shows the SMSE of the PoSSTenD, and YPS-SSD method. We do not report the SMSE of the other baseline methods (NMC-scan-stat, ZQ-Lasso, DBS-PCA, and T^2), becasue they cannot model the global trend mean. It is clear from Tables 3 that, our proposed PoSSTenD method performs well in terms of the background fitness in both two scenarios.

5. Case study

In this section, we apply our proposed PoSSTenD method to the infectious disease dataset described in Section 2. There are only two missing values, and we handle them by the mean imputation with a reasonable assumption that they are missing at random. If one encounters the missing-not-at-random case, one can use the methods in [3–5]. After the missing data is addressed, we use PoSSTenD for hot-spots detection and localization and compare its performance with other benchmarks as in Section 4.2.

First, we compare the performance of the detection delay. For all the methods, we set the control limits so that the average run length to false alarm constraint $ARL_0 = 50$ via Monte Carlo simulation under the assumption that data from the first 15 years are in control. For the setting of the parameters and the selection of basis, they are the same as that in Section 4. For our proposed PoSSTenD method, we build a CUSUM control chart utilizing the test statistic in Section 3.3.1, which is shown in Figure 6. From this figure, we can see that the hot-spots are detected in 2017 by our proposed PoSSTenD method. For the benchmark methods for comparison, we also use YPS-SSD (see [46]), ZQ-Lasso (see [53]), DBS-PCA (see [10]) and T^2 (see [39]) to our motivating dataset and summarize the performance of the detection of a hot-spots in Table 4. Note that the value in Table 4 is the first year that raises alarm, i.e. $\min_{t=1986,\ldots,2014}\{t : W_t^+ > L\}$. Our proposed PoSSTenD method, YPS-SSD, and ZQ-Lasso all raise alarms of hot-spots in the year 2017, while other benchmarks

Table 3. Fitting error of our method and YPS-SSD (decreasing population size and positive hot-spots).

Fitting error	$\delta = 0.05$	$\delta = 0.075$	$\delta = 0.1$	$\delta = 0.125$	$\delta = 0.15$	$\delta = 0.175$	$\delta = 0.2$
	Population with increasing trend						
PoSSTenD	0.0259×10^5	0.0259×10^5	0.0601×10^5	0.0601×10^5	0.0601×10^5	0.0988×10^5	0.0988×10^5
	(0.0150×10^3)	(0.0143×10^3)	(0.0397×10^3)	(0.0401×10^3)	(0.0423×10^3)	(0.0723×10^3)	(0.0702×10^3)
YPS-SSD	1.3218×10^5	1.3370×10^5	1.3520×10^5	1.3698×10^5	1.3899×10^5	1.4098×10^5	1.4282×10^6
	(1.1723×10^3)	(1.6122×10^3)	(2.0530×10^3)	(2.6108×10^3)	(3.0255×10^3)	(3.7379×10^3)	(4.0804×10^3)
	Population with decreasing trend						
PoSSTenD	1.3202×10^4	1.3198×10^4	1.3188×10^4	1.7187×10^4	1.7157×10^4	2.1140×10^4	2.1091×10^4
	(65.6439)	(68.9465)	(65.4763)	(0.0915×10^3)	(0.0934×10^3)	(0.1134×10^3)	(0.1201×10^3)
YPS-SSD	4.7511×10^4	4.8117×10^4	4.9009×10^4	5.0127×10^4	5.1698×10^4	5.3398×10^4	5.5376×10^4
	(745.1110)	(870.4066)	(963.7134)	(1.1771×10^3)	(1.3612×10^3)	(1.5949×10^3)	(1.8306×10^3)

[a] The above results are based on 1000 simulations
[b] The generated population size is of order 10^5.

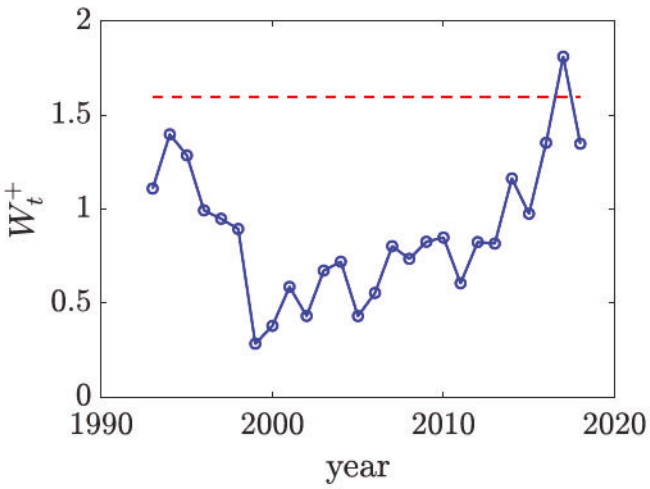

Figure 6. Control chart of our proposed method.

Table 4. Detection of change-point year in infectious rate dataset.

Methods	PoSSTenD	NMC-scan-stat	YPS-SSD	ZQ-Lasso	DBS-PCA	T^2
Year when an alarm is raised	2017	–	2017	2017	None	None

Notes: The label 'Year when an alarm is raised' is first year that raises alarm, i.e. $\min_{t=2008,\ldots,2018}\{t : W_t^+ > L\}$, where W_t^+ is the CUSUM statistics defined in Equation (10), and L is control limits to achieve the average run length to false alarm constraint $ARL_0 = 50$ via Monte Carlo simulation under the assumption that data from the first 15 years are in control.

fail to detect any hot-spots (we do not represent the hot-spots year of NMC-scan-stat, as it does not report the hot-spots). While nobody knows the ground truth when hot-spots occur in this real-world dataset, our numerical simulation experiences suggest that year 2017 is likely a hot-spot.

Next, after the temporal detection of hot-spots, we need to further localize the hot-spots in the sense that we need to find out which state and which type of disease may lead to the occurrence of the temporal hot-spot in 2017. Because for the baseline methods, DBS-PCA and T^2 can only realize the detection of temporal changes, and ZQ-Lasso declares all states as hot-spots, we only show the localization of spatial hot-spots by our proposed PoSSTenD method, NMC-scan-stat and YPS-SSD method in Figure 7. In Figure 7, one row is one type of disease and we select three types of diseases as representatives, i.e. mumps, syphilis, and pertussis. The first column is the raw data of the number of infected people in 2017, the second, third, and fourth columns are the hot-spots localized by our proposed PoSSTenD method, NMC-scan-stat, and YPS-SSD method, respectively. It can be seen from Figure 7 that, our proposed PoSSTenD method realizes more sparse hot-spots localization. Given the simulation results in Section 4, our proposed PoSSTenD method has very high precision, recall, and F measure, we can declare that these states are highly likely to be hot-spots. For NMC-scan-stat, it trends to localize clustered hot-spots, which has a very low recall. For YPS-SSD, since it has relative lower precision and recall than our method, there are more false-alarm than our method. This is reasonable for us to conclude that, our method has better performance in hot-spots localization than the selected benchmarks.

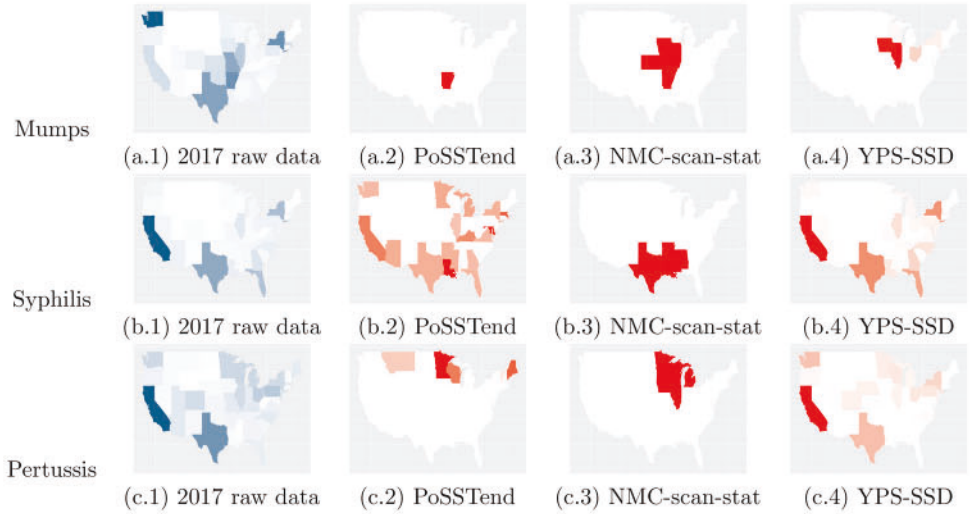

Figure 7. Hot-spots detection result of mumps, syphilis and pertussis in 2017 by our proposed PoSSTenD method (second column), NMC-scan-stat (third column) and YPS-SSD method (last column). The first column is the raw data of the number of infected people of mumps, syphilis and pertussis in 2017. (a.1) 2017 raw data. (a.2) PoSSTend. (a.3) NMC-scan-stat. (a.4) YPS-SSD (b.1) 2017 raw data. (b.2) PoSSTend. (b.3) NMC-scan-stat. (b.4) YPS-SSD (c.1) 2017 raw data. (c.2) PoSSTend. (c.3) NMC-scan-stat. (c.4) YPS-SSD.

Acknowledgments

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

Disclosure statement

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

Funding

This project is partially supported by Dr Huo's funding in the Transdisciplinary Research Institute for Advancing Data Science (TRIAD), http://triad.gatech.edu, which is a part of the TRIPODS program at NSF and locates at Georgia Tech, enabled by the NSF [grant number CCF-1740776]. Dr Huo is also supported in part by NSF [grant number DMS-2015363]. Dr Mei's research was supported in part by NSF [grant number DMS-2015405], and in part by the National Center for Advancing Translational Sciences of the National Institutes of Health under Award Number UL1TR000454.

ORCID

Yujie Zhao http://orcid.org/0000-0003-2896-4955
Yajun Mei http://orcid.org/0000-0002-1015-990X

References

[1] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM. J. Imaging. Sci. 2 (2009), pp. 183–202.

[2] L.M. Brègman, *Relaxation method for finding a common point of convex sets and its application to optimization problems*, Doklady Akademii Nauk. 171 (1966), pp. 1019–1022. Russian Academy of Sciences.
[3] J. Chen and F. Fang, *Semiparametric likelihood for estimating equations with non-ignorable nonresponse by non-response instrument*, J. Nonparametr. Stat. 31 (2019), pp. 420–434.
[4] J. Chen, F. Fang, and Z. Xiao, *Semiparametric inference for estimating equations with nonignorably missing covariates*, J. Nonparametr. Stat. 30 (2018a), pp. 796–812.
[5] J. Chen, J. Shao, and F. Fang, *Instrument search in pseudo-likelihood approach for nonignorable nonresponse*, Ann. Inst. Stat. Math. 73 (2021a), pp. 519–533.
[6] J. Chen, B. Xie, and J. Shao, *Pseudo likelihood and dimension reduction for data with nonignorable nonresponse*, Stat. Theory Relat. Fields. 2 (2018b), pp. 196–205.
[7] Q. Chen, J. Chen, Y. Zhou, L. Huang, Y. Tang, J. Li, and J. Zhang, *Natural history and associated early life factors of childhood asthma: A population registry-based cohort study in denmark*, BMJ. Open. 11 (2021b), pp. e045728.
[8] D. Conesa, M. Martínez-Beneito, R. Amorós, and A. López-Quílez, *Bayesian hierarchical poisson models with a hidden markov structure for the detection of influenza epidemic outbreaks*, Stat. Methods. Med. Res. 24 (2015), pp. 206–223.
[9] I. Daubechies, M. Defrise, and C. De Mol, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Commun. Pure Appl. Math.: A Journal Issued by the Courant Institute of Mathematical Sciences 57 (2004), pp. 1413–1457.
[10] B. De Ketelaere, M. Hubert, and E. Schmitt, *Overview of PCA-based statistical process-monitoring methods for time-dependent, high-dimensional data*, J. Qual. Technol. 47 (2015), pp. 318–335.
[11] M.E. Eren, J.S. Moore, and B.S Alexandro, *Multi-dimensional anomalous entity detection via poisson tensor factorization*, 2020 IEEE International Conference on Intelligence and Security Informatics (ISI), IEEE, 2020, pp. 1–6.
[12] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*, Vol. 1, Springer series in statistics New York, 1, Springer, New York, 2001.
[13] J. Friedman, T. Hastie, and R. Tibshirani, *Regularization paths for generalized linear models via coordinate descent*, J. Stat. Softw. 33 (2010), pp. 1.
[14] M.R. Gahrooei, H. Yan, K. Paynabar, and J. Shi, *Multiple tensor-on-tensor regression: An approach for modeling processes with heterogeneous sources of data*, Technometrics 63 (2021), pp. 147–159.
[15] S.W. Han, K. -L. Tsui, B. Ariyajunya, and S.B. Kim, *A comparison of cusum, ewma, and temporal scan statistics for detection of increases in poisson rates*, Qual. Reliab. Eng. Int. 26 (2010), pp. 279–289.
[16] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations*, CRC Press, 2015.
[17] D.M. Hawkins, *Regression adjustment for variables in multivariate quality control*, J. Qual. Technol. 25 (1993), pp. 170–182.
[18] F.L. Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, J. Math. Phys. 6 (1927), pp. 164–189.
[19] L Hunyadi, B-splines. 2020. Available at https://www.mathworks.com/matlabcentral/fileexchange/27374-b-splines.
[20] A. Ihler, J. Hutchins, and P Smyth, *Adaptive event detection with time-varying poisson processes*, Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006, pp. 207–216.
[21] R. Killick and I. Eckley, *Changepoint: An R package for changepoint analysis*, J. Stat. Softw. 58 (2014), pp. 1–19.
[22] M.H. Kim and M.G. Akritas, *Order thresholding*, Ann. Stat. 38 (2010), pp. 2314–2350.
[23] T.G. Kolda and B.W. Bader, *Tensor decompositions and applications*, SIAM Rev. 51 (2009), pp. 455–500.
[24] S. Kullback and R.A. Leibler, *On information and sufficiency*, Ann. Math. Stat. 22 (1951), pp. 79–86.

[25] M. Kulldorff, *A spatial scan statistic*, Commun. Stat.-Theor. Meth. 26 (1997), pp. 1481–1496.
[26] M. Kulldorff, *Prospective time periodic geographical disease surveillance using a scan statistic*, J. R. Stat. Soc.: Ser. A (Stat. Soc.). 164 (2001), pp. 61–72.
[27] M. Kulldorff, L. Huang, and K. Konty, *A scan statistic for continuous data based on the normal probability model*, Int. J. Health. Geogr. 8 (2009), pp. 1–9.
[28] M. Kulldorff and N. Nagarwalla, *Spatial disease clusters: Detection and inference*, Stat. Med. 14 (1995), pp. 799–810.
[29] K. Liu, R. Zhang, and Y. Mei, *Scalable sum-shrinkage schemes for distributed monitoring large-scale data streams*, Stat. Sin. 29 (2019), pp. 1–22.
[30] Lorden, *Procedures for reacting to a change in distribution*, Ann. Math. Stat. 42 (1971), pp. 1897–1908.
[31] H. Nagaraja, *Some nondegenerate limit laws for the selection differential*, Ann. Stat. 10 (1982), pp. 1306–1310.
[32] J.I. Naus, *Clustering of Random Points in Line and Plane*, Harvard University Press, 1963.
[33] D. Neill, A. Moore, and G. Cooper, *A Bayesian spatial scan statistic*, Adv. Neural. Inf. Process. Syst. 18 (2005), pp. 1003–1010.
[34] D.B. Neill, A.W. Moore, and G.F. Cooper, *A Bayesian spatial scan statistic*, Adv. Neural. Inf. Process. Syst. 18 (2006), pp. 1003–1010.
[35] E.S. Page, *Continuous inspection schemes*, Biometrika 4/1 (1954), pp. 100–115.
[36] K. Paynabar, J. Jin, and M. Pacella, *Monitoring and diagnosis of multichannel nonlinear profile variations using uncorrelated multilinear principal component analysis*, IIE. Trans. 45 (2013), pp. 1235–1247.
[37] J. Pinheiro and D. Bates, *Mixed-effects Models in S and S-PLUS*, Springer Science & Business Media, 2006.
[38] N.G. Polson, J.G. Scott, and B.T. Willard, *Proximal algorithms in statistics and machine learning*, Stat. Sci. 30 (2015), pp. 559–581.
[39] P. Qiu, *Introduction to Statistical Process Control*, Chapman and Hall/CRC, 2013.
[40] D. Ruppert, *Selecting the number of knots for penalized splines*, J. Comput. Graph. Stat. 11 (2002), pp. 735–757.
[41] T. Tango, K. Takahashi, and K. Kohriyama, *A space–time scan statistic for detecting emerging outbreaks*, Biometrics 67 (2011), pp. 106–115.
[42] R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. R. Stat. Soc. Ser. B (Methodol.) 58 (1996), pp. 267–288.
[43] M. Turcotte, J. Moore, N. Heard, and A McPhall, *Poisson factorization for peer-based anomaly detection*, 2016 IEEE Conference on Intelligence and Security Informatics (ISI), IEEE, 2016, pp. 208–210.
[44] H. Yan, K. Paynabar, and J. Shi, *Image-based process monitoring using low-rank tensor decomposition*, IEEE. Trans. Autom. Sci. Eng. 12 (2014), pp. 216–227.
[45] H. Yan, K. Paynabar, and J. Shi, *Anomaly detection in images with smooth background via smooth-sparse decomposition*, Technometrics 59 (2017), pp. 102–114.
[46] H. Yan, K. Paynabar, and J. Shi, *Real-time monitoring of high-dimensional functional data streams via spatio-temporal smooth sparse decomposition*, Technometrics 60 (2018a), pp. 181–197.
[47] H. Yan, K. Paynabar, and J. Shi, *Real-time monitoring of high-dimensional functional data streams via spatio-temporal smooth sparse decomposition*, Technometrics 60 (2018b), pp. 181–197.
[48] Y Zhao, *New progress in hot-spots detection, partial-differential-equation-based model identification and statistical computation*, PhD thesis, Georgia Institute of Technology, 2021.
[49] Y. Zhao, and X Huo, *A homotopic method to solve the lasso problems with an improved upper bound of convergence rate*, preprint (2020). Available at arXiv:2010.13934.
[50] Y. Zhao, X. Huo, and Y Mei, *Identification of underlying dynamic system from noisy data with splines*, preprint (2021a). Available at arXiv:2103.10231.
[51] Y. Zhao, H. Yan, S. Holte, and Y. Mei, *Rapid detection of hot-spots via tensor decomposition with applications to crime rate data*, J. Appl. Stat. 49 (2021b), pp. 1–27.

[52] Y. Zhao, H. Yan, S.E. Holte, R.P. Kerani, and Y Mei, *Rapid detection of hot-spot by tensor decomposition with application to weekly gonorrhea data*, International Workshop on Intelligent Statistical Quality Control, Springer, 2019, pp. 265–286.
[53] C. Zou and P. Qiu, *Multivariate statistical process control using lasso*, J. Am. Stat. Assoc. 104 (2009), pp. 1586–1596.
[54] C. Zou, F. Tsung, and Z. Wang, *Monitoring profiles based on nonparametric regression methods*, Technometrics 50 (2008), pp. 512–526.

Appendices

Appendix 1. Proof for Proposition 3.1

Proof: According to Newton-Raphson method, we update from $\boldsymbol{\theta}_m^{(k+1)}$ to $\boldsymbol{\theta}_m^{(k)}$ as follows:

$$\boldsymbol{\theta}_m^{(k+1)} = \boldsymbol{\theta}_m^{(k)} - \left[\frac{\partial^2}{\partial\boldsymbol{\theta}_m\partial\boldsymbol{\theta}_m^\top}F(\boldsymbol{\theta}_m,\boldsymbol{\theta}_h)\right]^{-1}\frac{\partial}{\partial\boldsymbol{\theta}_m}F(\boldsymbol{\theta}_m,\boldsymbol{\theta}_h), \tag{A1}$$

where we have

$$\begin{aligned}
\frac{\partial}{\partial\boldsymbol{\theta}_m}F(\boldsymbol{\theta}_m,\boldsymbol{\theta}_h) &= \sum_{i=1}^{n} -y_i\mathbf{x}_i + n_i e^{(\mathbf{x}_i^\top\boldsymbol{\theta}_m+\mathbf{z}_i^\top\boldsymbol{\theta}_h)}\mathbf{x}_i \triangleq -\mathbf{X}^\top\mathbf{y}+\mathbf{X}^\top\boldsymbol{\gamma}\\
\frac{\partial^2}{\partial\boldsymbol{\theta}_m\partial\boldsymbol{\theta}_m^\top}F(\boldsymbol{\theta}_m,\boldsymbol{\theta}_h) &= \sum_{i=1}^{n} n_i e^{(\mathbf{x}_i^\top\boldsymbol{\theta}_m+\mathbf{z}_i^\top\boldsymbol{\theta}_h)}\mathbf{x}_i\mathbf{x}_i^\top \triangleq \mathbf{X}^\top\mathbf{W}\mathbf{X}\\
\frac{\partial}{\partial\boldsymbol{\theta}_h}F(\boldsymbol{\theta}_m,\boldsymbol{\theta}_h) &= \sum_{i=1}^{n} -\frac{1}{N_i}y_i\mathbf{z}_i + e^{(\mathbf{x}_i^\top\boldsymbol{\theta}_m+\mathbf{z}_i^\top\boldsymbol{\theta}_h)}\mathbf{z}_i \triangleq -\mathbf{Z}^\top\mathbf{y}+\mathbf{Z}^\top\boldsymbol{\eta},
\end{aligned}$$

where the i-th entry of vector $\boldsymbol{\gamma}$ is $n_i e^{(\mathbf{x}_i^\top\boldsymbol{\theta}_m^{(k-1)}+\mathbf{z}_i^\top\boldsymbol{\theta}_h^{(k-1)})}$ and matrix $\mathbf{W}$ is a diagonal matrix whose (i,i)-th entry is $n_i e^{(\mathbf{x}_i^\top\boldsymbol{\theta}_m^{(k-1)}+\mathbf{z}_i^\top\boldsymbol{\theta}_h^{(k-1)})}$. So we can rewrite Equation (A1) as follows:

$$\begin{aligned}
\boldsymbol{\theta}_m^{(k+1)} &= \boldsymbol{\theta}_m^{(k)} - \left[\frac{\partial^2}{\partial\boldsymbol{\theta}_m\partial\boldsymbol{\theta}_m^\top}F(\boldsymbol{\theta}_m,\boldsymbol{\theta}_h)\right]^{-1}\frac{\partial}{\partial\boldsymbol{\theta}_m}F(\boldsymbol{\theta}_m,\boldsymbol{\theta}_h)\\
&= \boldsymbol{\theta}_m^{(k)} + \left(\mathbf{X}^\top\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^\top(\mathbf{y}-\boldsymbol{\gamma})\\
&= \left(\mathbf{X}^\top\mathbf{W}\mathbf{X}\right)^{-1}\left(\mathbf{X}^\top\mathbf{W}\mathbf{X}\right)\boldsymbol{\theta}_m^{(k)} + \left(\mathbf{X}^\top\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^\top(\mathbf{y}-\boldsymbol{\gamma})\\
&= \left(\mathbf{X}^\top\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{W}\left[\mathbf{X}\boldsymbol{\theta}_m^{(k)} + \mathbf{W}^{-1}(\mathbf{y}-\boldsymbol{\gamma})\right]\\
&\triangleq \left(\mathbf{X}^\top\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{W}\boldsymbol{\eta},
\end{aligned}$$

where $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\theta}_m^{(k)} + \mathbf{W}^{-1}(\mathbf{y}-\boldsymbol{\gamma})$. ■

Appendix 2. Estimation results for the logistic model of population size in Section 4

The estimation of $\{\phi_{i,j,1},\phi_{i,j,2},\phi_{i,j,3}\}_{i=1,\ldots,49}$ is summarized in Table A1.

Appendix 3. Table to generate Figure 4

In this section, we present the table to generate Figure 4 in Table A2.

Table A1. Estimation of $\{\phi_{i,j,1}, \phi_{i,j,2}, \phi_{i,j,3}\}_{i=1,\ldots,49}$ in Section 4.

	State	$\widehat{\phi}_{i,j,1}$	$\widehat{\phi}_{i,j,2}$	$\widehat{\phi}_{i,j,3}$		State	$\widehat{\phi}_{i,j,1}$	$\widehat{\phi}_{i,j,2}$	$\widehat{\phi}_{i,j,3}$
$i = 1$	(Alabama	80.6404	−4.0135	66.0162	$i = 26$	Nebraska	90.1254	168.2284	109.1338
$i = 2$	(Arizona	82.7195	2.6681	12.4089	$i = 27$	Nevada	35.6347	6.3429	11.1336
$i = 3$	Arkansas	37.4245	−15.4787	28.1058	$i = 28$	New Hampshire	13.8539	−13.8127	10.6643
$i = 4$	California	541.8391	−8.4830	33.6786	$i = 29$	New Jersey	92.5943	−21.7473	13.7113
$i = 5$	Colorado	98.7504	17.0426	28.2669	$i = 30$	New Mexico	24.5653	−12.2919	20.4325
$i = 6$	Connecticut	38.0542	−33.2342	20.1424	$i = 31$	New York	187.9958	−10.3970	3.8098
$i = 7$	Delaware	18.8717	23.0528	40.3679	$i = 32$	North Carolina	209.7977	25.8509	34.7033
$i = 8$	District of Columbia	84.8814	234.3312	85.0780	$i = 33$	North Dakota	61.2268	266.2989	120.0412
$i = 9$	Florida	472.3076	32.7741	35.0153	$i = 34$	Ohio	120.8197	-61.9571	26.2627
$i = 10$	Georgia	128.0025	−1.0137	17.0737	$i = 35$	Oklahoma	170.7639	136.1955	92.1323
$i = 11$	Idaho	47.9991	46.7379	38.0065	$i = 36$	Oregon	80.6280	22.5818	43.4817
$i = 12$	Illinois	130.1208	−16.1579	8.3610	$i = 37$	Pennsylvania	119.5912	−156.0891	4.3750
$i = 13$	Indiana	87.0352	−26.3047	42.7584	$i = 38$	Rhode Island	10.6925	−17.1867	7.8247
$i = 14$	Iowa	110.5464	168.3283	154.6580	$i = 39$	South Carolina	287.8320	116.8317	59.0815
$i = 15$	Kansas	35.6595	−33.1733	38.3685	$i = 40$	South Dakota	63.0675	210.4712	100.9319
$i = 16$	Kentucky	61.6822	−21.2109	47.4401	$i = 41$	Tennessee	126.5228	18.9025	45.5198
$i = 17$	Louisiana	138.2943	170.8833	212.9227	$i = 42$	Texas	1296.6000	79.1413	42.5022
$i = 18$	Maine	13.9479	−37.1852	19.3080	$i = 43$	Utah	87.5383	44.0237	32.6645
$i = 19$	Maryland	107.4051	10.3277	56.6712	$i = 44$	Vermont	287.8320	116.8317	59.0815
$i = 20$	Massachusetts	172.8622	72.9025	113.4860	$i = 45$	Virginia	287.8320	116.8317	59.0815
$i = 21$	Michigan	100.0291	−9.4310	3.8436	$i = 46$	Washington	72.4669	0.2911	6.1436
$i = 22$	Minnesota	79.6624	−10.0102	41.2455	$i = 47$	West Virginia	40.7546	388.1061	1747.3000
$i = 23$	Mississippi	30.6879	−20.8025	12.1407	$i = 48$	Wisconsin	63.7490	−29.2415	23.1078
$i = 24$	Missouri	68.6103	−26.4275	23.9467	$i = 49$	Wyoming	60.6138	173.6913	66.5942
$i = 25$	Montana	65.8753	179.3000	92.5485					

Table A2. ARL_1 of our proposed PoSSTenD method, YPS-SSD method and ZQ-Lasso under population with decreasing trend.

Fitting error	$\delta = 0.05$	$\delta = 0.075$	$\delta = 0.1$	$\delta = 0.125$	$\delta = 0.15$	$\delta = 0.175$	$\delta = 0.2$
				Population with increasing trend			
PoSSTenD	2.0150	1.3500	1.0780	1.0140	1.0010	1.0000	1.0000
	(1.4610)	(0.6049)	(0.3000)	(0.1175)	(0.0316)	(0.0000)	(0.0000)
YPS-SSD	1.1806	1.0830	1.0350	1.0160	1.0080	1.0030	1.0010
	(0.5138)	(0.3259)	(0.2093)	(0.1542)	(0.1093)	(0.0547)	(0.0316)
ZQ-Lasso	9.0609	9.0000	9.0620	9.0730	9.0670	9.0820	9.0730
	(1.0593)	(1.1096)	(1.0580)	(1.0162)	(1.0704)	(1.0865)	(1.0406)
DBS-PCA	10.7515	10.7360	10.7090	10.7220	10.7340	10.7620	10.7330
	(0.6330)	(0.7778)	(0.8666)	(0.7832)	(0.7442)	(0.8295)	(0.7939)
				population with decreasing trend			
PoSSTenD	4.2780	1.5580	1.0500	1.0000	1.0000	1.0000	1.0000
	(3.6105)	(1.1908)	(0.2357)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
YPS-SSD	11.0000	10.9900	10.8300	9.6910	6.2810	3.2390	1.5630
	(0.0000)	(0.3162)	(1.2620)	(3.2690)	(4.8147)	(3.9708)	(2.1750)
ZQ-Lasso	11.0000	11.0000	11.0000	10.9550	9.6600	5.7020	2.4990
	(0.0000)	(0.0000)	(0.0000)	(0.6383)	(3.3238)	(4.7103)	(2.9734)
DBS-PCA	10.0200	9.6900	9.2750	8.7720	8.6040	8.5750	8.6920
	(2.0995)	(2.6342)	(3.0684)	(3.4392)	(3.6128)	(3.6477)	(3.6631)

[a] The above results are based on 1000 simulations.