Rapid Detection of Hot-spot by Tensor Decomposition on Space and Circular Time with Application to Weekly Gonorrhea data

Yujie Zhao, Hao Yan, Sarah Holte, Yajun Mei

April 8, 2020

Abstract

In many bio-surveillance and healthcare applications, data sources are measured from many spatial locations repeatedly over time, say, daily/weekly/monthly. In these applications, we are typically interested in detecting hot-spots, which are defined as some structured outliers that are sparse over the spatial domain but persistent over time. In this paper, we propose a tensor decomposition method to detect when and where the hot-spots occur. Our proposed methods represent the observed raw data as a three-dimensional tensor including a circular time dimension for daily/weekly/monthly patterns, and then decompose the tensor into three components: smooth global trend, local hot-spots, and residuals. A combination of LASSO and fused LASSO is used to estimate the model parameters, and a CUSUM procedure is applied to detect when and where the hot-spots might occur. The usefulness of our proposed methodology is validated through numerical simulation and a real-world dataset in the weekly number of gonorrhea cases from 2006 to 2018 for 50 states in the United States.

Key words: Circular time, CUSUM, hot-spot, spatio-temporal model, tensor decomposition

1 Introduction

In many bio-surveillance and healthcare applications, data sources are measured from many spatial locations repeatedly over time, say, daily, weekly,

or monthly. In these applications, we are typically interested in detecting *hot-spots*, which are defined as some structured outliers that are sparse over the spatial domain but persistent over time. A concrete real-world motivating application is the weekly number of gonorrhea cases from 2006 to 2018 for 50 states in the United States, also see the detailed data description in the next section. From the monitoring viewpoint, there are two kinds of changes: one is the global-level trend, and the other is the local-level outliers. Here we are more interested in detecting the so-called hot-spots, which are local-level outliers with the following two properties: (1) spatial sparsity, i.e., the local changes are sparse over the spatial domain; and (2) temporal persistence, i.e., the local changes last for a reasonably long time period unless one takes some actions.

Generally speaking, the hot-spot detection can be thought as detecting sparse anomaly in spatio-temporal data, and there are three different categories of methodologies and approaches in the literature. The first one is LASSO-based control chart that integrates LASSO estimators for change point detection and declares non-zero components of the LASSO estimators as the hot-spot, see Zou and Qiu (2009), Zou et al. (2012), Šaltytė Benth and Šaltytė (2011). Unfortunately, the LASSO-based control chart lacks the ability to separate the local hot-spots from the global trend of the spatio-temporal data. The second category of methods is the dimension reduction based control chart where one monitors the features from PCA or other dimension reduction methods, see De Ketelaere et al. (2015), Louwerse and Smilde (2000), Hu and Yuan (2009). The drawback of PCA or other dimension reduction methods is that it fails to detect sparse anomalies and cannot take full advantage of the spatial location of hot-spot. The third category of anomaly detection methods is the decomposition-based method that uses the regularized regression methods to separate the hot-spots from the background event, see Tran et al. (2012), Yan et al. (2017), Yan et al. (2018). However, these existing approaches investigate structured images or curves data under the assumption that the hot-spots are independent over the time domain.

In this paper, we propose a decomposition-based anomaly detection method for spatial-temporal data when the hot-spots are autoregressive, which is typical for time series data. Our main idea is to represent the raw data as a 3-dimensional tensor: states, weeks, years. To be more specific, at each year, we observe a 50×52 data matrix that corresponds to 50 states and 52 weeks (we ignore the leap years). Next, we propose to decompose the 3-dimension tensor into three components: Smooth global trend, Sparse local hot-spot,

and Residuals, and term our proposed decomposition model as SSR-Tensor. When fitting the observed raw data to our proposed SSR-Tensor model, we develop a penalized likelihood approach by adding two penalty functions: one is the LASSO type penalty to guarantee the sparsity of hot-spots, and the other is the fused-LASSO type penalty for the autoregressive properties of hot-spots or time-series data. By doing so, we are able to (1) detect when the hot-spots occur (i.e., the change point detection problem); and (2) localize where and which type of the hot-spots occur (i.e., the spatial localization problem).

We would like to acknowledge that much research has been done on modeling and prediction of the spatio-temporal data. Some popular time series models are AR, MA, ARMA model, etc., and the parameter can be estimated by Yule-Walker method (Hannan and Quinn, 1979), maximum likelihood estimation or least square method (Hamilton, 1994). In addition, spatial statistics have also been extensively investigated on its own right, see (Reynolds and Madden, 1988; Lichstein et al., 2002; Lan et al., 2004; Elhorst, 2014; Call and Voss, 2016) for examples. When one combines time series with spatial statistics, the corresponding spatio-temporal models generally become more complicated, see (Zhu et al., 2005; Lai and Lim, 2015; Diggle, 2013) for more discussions.

In principle, it is possible to represent the spatio-temporal process as a sequence of random vector $\mathbf{Y}_t$ with weekly observation t, where $\mathbf{Y}$ is p-dimensional vector that characterize the spatial domain (i.e., spatial dimension $p = 50$ in our case study). However, such an approach might not be computationally feasible in the context of hot-spot detection, in which one needs to specify the covariance structure of $\mathbf{Y}_t$, not only over the spatial domain, but also over the time domain. If we wrote all data into a vector, then the dimension of such vector is $50 \times 52 \times 13 = 33,800$, and thus the covariance matrix is of dimension $33,800 \times 33,800$, which is not computationally feasible. Meanwhile, under our proposed SSR-Tensor model, we essentially conduct a dimensional reduction by assuming that such a covariance matrix has a nice sparsity structure, as we reduce the dimensions $50, 52$ and 13 to much smaller numbers, e.g., AR(1) model over the week or year dimension, and local correlation over the spatial domain.

It is useful to point out that while our paper focuses only on 3-dimensional tensor due to our motivating application in gonorrhea, our proposed SSR-Tensor model can easily be extended to any d-dimensional tensor or data with $d \geq 3$, e.g., when we have further information, such as the unemployment

rate, economic performance, and so on. As the dimension d increases, we can simply add more corresponding bases, as our proposed model uses *basis* to describe correlation within each dimension, and utilizes *tensor product* for interaction between different dimensions. The capability of extending to high-dimensional data is one of the main advantages of our proposed SSR-Tensor model. Furthermore, our proposed SSR-Tensor model essentially involves block-wise diagonal covariation matrix, which allows ut to develop computationally efficient methodologies by using tensor decomposition algebra, see Section 5.2 for more technical details.

The remainder of this paper is as follows. Section 2 discusses and visualizes the gonorrhea dataset, which is used as our motivating example and in our case study. Section 3 presents our proposed SSR-Tensor model, and discusses how to estimate model parameters from observed data. Section 4 describes how to use our proposed SSR-Tensor model to find hot-spots, both for temporal detection and for spatial localization. Efficient numerical optimization algorithms are discussed in Section 5. Our proposed methods are then validated through extensive simulations in Section 6 and a case study in gonorrhea dataset in Section 7.

2 Data Description

To protect Americans from serious disease, the National Notifiable Disease Surveillance System (NNDSS) at the Centers for Disease Control and Prevention (CDC) helps public health monitor, control, and prevent about 120 diseases, see its website `https://wwwn.cdc.gov/nndss/infectious-tables.html`. One disease that receives intensive attention in recent years is gonorrhea, due to the possibility of multi-drug resistances. Historically the instances of antibiotic resistance (in gonorrhea) have first been in the west and then move across the country. Since 1965, the CDC has collected the number of cumulative new infected patients every week in a calendar year. There are several changes on report policies or guidelines, and the latest one is year 2006. As a result, we focus on the weekly numbers of new gonorrhea patients during January 1, 2006 and December 31, 2018. The new weekly gonorrhea cases are computed as the difference of the cumulative cases in two consecutive weeks. The last week is dropped during this calculation.

Let us first discuss the spatial patterns of the gonorrhea data among 50 states. For this purpose, we consider the cumulative number of gonorrhea

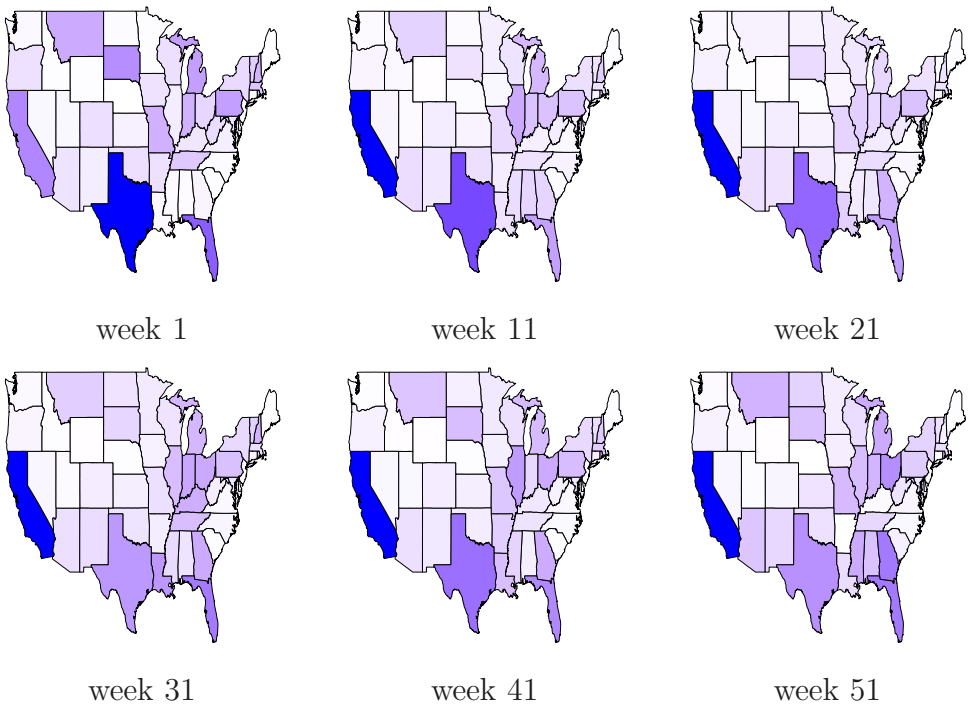

Figure 1: The cumulative number of gonorrhea cases at some selected weeks during years 2006-2018. The deeper the color, the higher number of gonorrhea cases.

cases from week 1 to week 52 by sum up all data during years 2006-2018. Figure 1 plots some selected weeks (#1, #11, #21, #31, #41, #51). In Figure 1, if the state has a deeper and bluer color, then it experiences a higher number of gonorrhea cases. One obvious pattern is that, California and Texas have generally higher number of gonorrhea cases as compared to other states. In addition, the number of gonorrhea cases in the northern US is smaller than that in the southern US.

Next, we consider the temporal pattern of the gonorrhea data set. Figure 2 plots the annual number of gonorrhea cases over the years 2006-2018 in the US. It is evident that there is a global-level decreasing trend during 2010-2013. One possible explanation is the Obamacare, which seems to reduce the risk of infectious diseases. As we mentioned before, we are not interested in detecting this type of global changes, and we focus on the detection of the changes on the local patterns, which are referred to as hot-spots in our

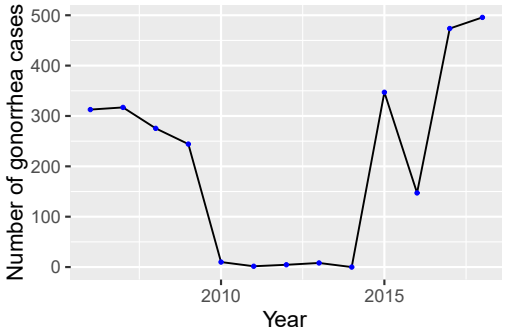

Figure 2: Annual number of gonorrhea cases (in thousands) over the years 2006-2018 in the US

paper.

Moreover, the gonorrhea data consists of weekly data, and thus it is necessary to address the circular patterns over the direction of "week". Figure 3 shows the country-scaled weekly gonorrhea case in the form of "rose" diagram for some selected years. In this figure, each direction represents a given week, and the length represents the number of gonorrhea cases for a given week. It reveals differences in the number of gonorrhea cases across a different week of the year. For instance, in July and August (in the direction of 8 o'clock on the circle), the number of gonorrhea case tends to be larger than other weeks.

3 Proposed Model

In this section, we present our proposed SSR-Tensor model, and postpone the discussion of hot-spot detection methodology to the next section. Owing to the fact that the gonorrhea data is of three dimensions, namely, {state, week, year}, it will likely have complex "within-dimension" and "between-dimension" interaction/correaltion relationship. Within-dimension relationship includes within-state correlation, within-week correlation, and within-year correlation. Between-dimension relationship includes between-state-and-week interaction, between-state-and-year interaction, as well as between-week-and-year interaction. In order to handle these complex "within" and "between" interaction structures, we propose to use the tensor decomposition method, where bases are used to address "within-dimension" correlation, and

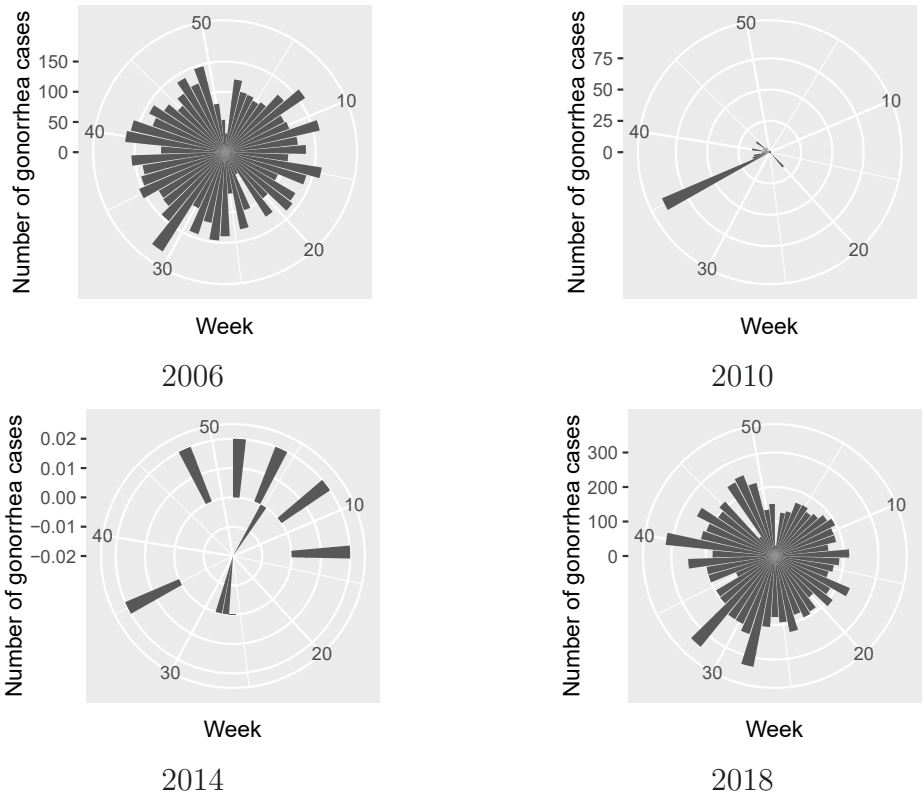

Figure 3: Histograms of the number of gonorrhea cases of Year 2006, 2010, 2014, 2018. Each direction represents a given week, and the length represents the number of gonorrhea cases for a given week.

the tensor product is used for "between-dimension" interaction. Here, the basis is a very important concept where different basis can be chosen for different dimensions. Detailed discussions of the choice of bases are presented in Section 6.2.

For the convenience of notation and easy understanding, we first introduce some basic tensor algebra and notation in Section 3.1. Then Section 3.2 presents our proposed model that is able to characterize the complex correlation structures.

3.1 Tensor Algebra and Notation

In this section, we introduce basic notations, definitions, and operators in tensor (multi-linear) algebra that are useful in this paper. Throughout the paper, scalars are denoted by lowercase letters (e.g., θ), vectors are denoted by lowercase boldface letters ($\boldsymbol{\theta}$), matrices are denoted by uppercase boldface letter ($\boldsymbol{\Theta}$), and tensors by curlicue letter (ϑ). For example, an order-N tensor is represented by $\vartheta \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, where I_k represent the mode-n dimension of ϑ for $k = 1, \ldots, N$.

The mode-n product of a tensor $\vartheta \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ by a matrix $\mathbf{B} \in \mathbb{R}^{J_n \times I_n}$ is a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \ldots I_{n-1} \times J_n \times I_{n+1} \times \ldots I_N}$, denoted as $\mathcal{A} = \vartheta \times_n \mathbf{B}$, where each entry of $\mathcal{A}$ is defined as the sum of products of corresponding entries in $\mathcal{A}$ and $\mathbf{B}$: $\mathcal{A}_{i_1,\ldots,i_{n-1},j_n,i_{n+1},\ldots,i_N} = \sum_{i_n} \vartheta_{i_1,\ldots,i_N} \mathbf{B}_{j_n,i_n}$. Here we use the notation $\mathbf{B}_{j_n,i_n}$ to refer the (j_n, i_n)-th entry in matrix $\mathbf{B}$. The notation $\vartheta_{i_1,\ldots,i_N}$ is used to refer to the entry in tensor ϑ with index $(i_1, \ldots, i_N)$. The notation $\mathcal{A}_{i_1,\ldots,i_{n-1},j_n,i_{n+1},\ldots,i_N}$ is used to refer the entry in tensor $\mathcal{A}$ with index $(i_1, \ldots, i_{n-1}, j_n, i_{n+1}, \ldots, i_N)$.

The mode-n unfold of the tensor $\vartheta \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ is denoted by $\vartheta_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times \ldots I_{n-1} \times I_{n+1} \times I_N)}$, where the column vector of $\vartheta_{(n)}$ are the mode-n vector of ϑ. The mode-n vector of ϑ are defined as the I_n dimensional vector obtained from ϑ by varying the index i_n while keeping all the other indices fixed. For example, $\vartheta_{:,2,3}$ is a model-1 vector.

A very useful technique in the tensor algebra is the Tucker decomposition, which decomposes a tensor into a core tensor multiplied by matrices along each mode: $\mathcal{Y} = \vartheta \times_1 \mathbf{B}^{(1)} \times_2 \mathbf{B}^{(2)} \cdots \times_N \mathbf{B}^{(N)}$, where $\mathbf{B}^{(n)}$ is an orthogonal $I_n \times I_n$ matrix and is a principal component mode-n for $n = 1, \ldots, N$. Tensor product can be represented equivalently by a Kronecker product, i.e., $\text{vec}(\mathcal{Y}) = (\mathbf{B}^{(N)} \otimes \cdots \otimes \mathbf{B}^{(1)})\text{vec}(\vartheta)$, where $\text{vec}(\cdot)$ is the vectorized operator. Finally, the definition of Kronecker product is as follow: Suppose $\mathbf{B}_1 \in \mathbb{R}^{m \times n}$

and $\mathbf{B}_2 \in \mathbb{R}^{p\times q}$ are matrices, the Kronecker product of these matrices, denoted by $\mathbf{B}_1 \otimes \mathbf{B}_2$, is an $mq \times nq$ block matrix defined by

$$\mathbf{B}_1 \otimes \mathbf{B}_2 = \begin{bmatrix} b_{11}\mathbf{B}_2 & \cdots & b_{1n}\mathbf{B}_2 \\ \vdots & \ddots & \vdots \\ b_{m1}\mathbf{B}_2 & \cdots & b_{mn}\mathbf{B}_2 \end{bmatrix}.$$

3.2 Our Proposed SSR-Tensor Model

Our proposed SSR-Tensor model is built on tensors of order three, as it is inspired by the gonorrhea data, which can be represented as a three-dimension tensor $\mathcal{Y}_{n_1\times n_2\times T}$ with $n_1 = 50$ states, $n_2 = 51$ weeks, and $T = 13$ years. Note that the i-th,j$-th$, and k-th slice of the 3-D tensor along the dimension of state, week, and year can be achieved as $\mathcal{Y}_{i::}, \mathcal{Y}_{:j:}, \mathcal{Y}_{::k}$ correspondingly, where $i = 1\cdots n_1$, $j = 1\cdots n_2$ and $k = 1\cdots T$. For simplicity, we denote $\mathbf{Y}_k = \mathcal{Y}_{::k}$. We further denote $\mathbf{y}_k$ as the vectorized form of $\mathbf{Y}_k$, and $\mathbf{y}$ as the vectorized form of $\mathcal{Y}$.

The key idea of our proposed model is to separate the global trend from the local pattern by decomposing the tensor $\mathbf{y}$ into three parts, namely the smooth global trend $\boldsymbol{\mu}$, local hot-spot $\mathbf{h}$, and residual $\mathbf{e}$, i.e. $\mathbf{y} = \boldsymbol{\mu} + \mathbf{h} + \mathbf{e}$. For the first two of the components (e.g. the global trend mean and local hot-spots), we introduce basis decomposition framework to represent the structure of the within correlation in the global background and local hot-spot, also see Yan et al. (2018).

To be more concrete, we assume that global trend mean and local hot-spot can be represented as $\boldsymbol{\mu} = \mathbf{B}_m\boldsymbol{\theta}_m$ and $\boldsymbol{h} = \mathbf{B}_h\boldsymbol{\theta}_h$, where $\mathbf{B}_m$ and $\mathbf{B}_h$ are two bases that will discussed below, and $\boldsymbol{\theta}_m$ and $\boldsymbol{\theta}_h$ are the model coefficients vector of length n_1n_2T and needed to be estimated (see Section 5). Here the subscript of m and h are abbreviations for mean and hot-spot. Next, it is useful to discuss how to choose the bases $\mathbf{B}_m$ and $\mathbf{B}_h$, so as to characterize the complex "within" and "between" correlation or interaction structures. For the "within" correlation structures, we propose to use pre-specified bases, $\mathbf{B}_{m,s}$ and $\mathbf{B}_{h,s}$, for within-state correlation in global trend and hot-spot, where the subscript of s is an abbreviation for states. Similarly, $\mathbf{B}_{m,w}$ and $\mathbf{B}_{h,w}$ are the pre-specified bases for within-correlation of the same week, whereas $\mathbf{B}_{m,y}$ and $\mathbf{B}_{h,y}$ are the bases for within-time correlation over time. As for the "between" interaction, we use tensor product to describe it,

i.e, $\mathbf{B}_m = \mathbf{B}_{m,s} \otimes \mathbf{B}_{m,w} \otimes \mathbf{B}_{m,y}$ and $\mathbf{B}_h = \mathbf{B}_{h,s} \otimes \mathbf{B}_{h,w} \otimes \mathbf{B}_{h,y}$. This Kronecker product has been proved to have better computational efficiency in the tensor response data Kolda and Bader (2009). Mathematically speaking, all these bases are matrices, which is pre-assigned in our paper. And the choice of bases in shown in Section 6.2. With the well-structured "within" and "between" interaction, our proposed model can be written as:

$$\mathbf{y} = (\mathbf{B}_{m,s} \otimes \mathbf{B}_{m,w} \otimes \mathbf{B}_{m,y})\boldsymbol{\theta}_m + (\mathbf{B}_{h,s} \otimes \mathbf{B}_{h,w} \otimes \mathbf{B}_{h,y})\boldsymbol{\theta}_h + \mathbf{e}, \qquad (1)$$

where $\mathbf{e} \sim N(0, \sigma^2\mathbf{I})$ is the random noise. Mathematically speaking, both $\mathbf{B}_{m,s}$ and $\mathbf{B}_{h,s}$ are $n_1 \times n_1$ matrix, $\mathbf{B}_{m,w}$ and $\mathbf{B}_{h,w}$ are $n_2 \times n_2$ matrix and $\mathbf{B}_{m,y}$ and $\mathbf{B}_{h,y}$ are $T \times T$ matrix, respectively.

Mathematically, our proposed model in (1) can be rewritten into a tensor format:

$$\mathcal{Y} = \vartheta_m \times_3 \mathbf{B}_{m,y} \times_2 \mathbf{B}_{m,w} \times_1 \mathbf{B}_{m,s} + \vartheta_h \times_3 \mathbf{B}_{h,y} \times_2 \mathbf{B}_{h,w} \times_1 \mathbf{B}_{h,s} + \mathbf{e}, \qquad (2)$$

where ϑ_m and ϑ_h is the tensor format of $\boldsymbol{\theta}_m$ and $\boldsymbol{\theta}_h$ with dimensional $n_1 \times n_2 \times T$. Accordingly, the $((k-1)n_1n_2 + (i-1)n_1 + j)$-th entry of $\boldsymbol{\theta}_h$, $\boldsymbol{\theta}_m$ can estimate the global mean and hot-spot in i-th state and j-th week in k-th year respectively. The tensor representation in equation (2) allows us to develop computationally efficient methods for estimation and prediction.

3.3 Estimation of Hot-spots

With the proposed SSR-Tensor model above, we can now discuss the estimation of hot-spot parameters $\boldsymbol{\theta}$'s (including $\boldsymbol{\theta}_m$, $\boldsymbol{\theta}_h$) in our model in (1) or (2) from the data via the penalized likelihood function. We propose to add two penalties in our estimation. First, because hot-spots rarely occur, we assume that $\boldsymbol{\theta}_h$ is sparse and the majority of entries in the hot-spot coefficient $\boldsymbol{\theta}_h$ are zeros. Thus we propose to add the penalty $R_1(\boldsymbol{\theta}_h) = \lambda\|\boldsymbol{\theta}_h\|_1$ to encourage the sparsity property of $\boldsymbol{\theta}_h$. Second, we assume there is temporal continuity of the hot-spots, as the usual phenomenon of last year is likely to affect the performance of hot-spot in this year. Thus, we add the second penalty $R_2(\boldsymbol{\theta}_h) = \lambda_2\|\mathbf{D}\boldsymbol{\theta}_h\|_1$ to ensure the yearly continuity of the hot-spot, where $\mathbf{D} = \mathbf{D}_s \otimes \mathbf{D}_w \otimes \mathbf{D}_y$ with $\mathbf{D}_s$ as identical matrix of dimension $n_1 \times n_1$, and $T \times T$ matrix $\mathbf{D}_y = \begin{bmatrix} 1 & -1 & & & \\ & \ddots & \ddots & & \\ & & 1 & -1 & \\ & & & 1 & \end{bmatrix}$, $n_2 \times n_2$ matrix

$\mathbf{D}_w = \begin{bmatrix} 1 & -1 & & & \\ & \ddots & \ddots & & \\ & & 1 & -1 & \\ -1 & & & 1 & \end{bmatrix}$. With the formula of $\mathbf{D}_y$, the hot-spot has the property of yearly continuity. By the formula of $\mathbf{D}_w$, the hot-spot has a weekly circular pattern.

By combining both penalties, we propose to estimate the parameters via the following optimization problem:

$$\begin{aligned} &\arg\min_{\boldsymbol{\theta}_m, \boldsymbol{\theta}_h} \|\boldsymbol{e}\|^2 + \lambda_1 \|\boldsymbol{\theta}_h\|_1 + \lambda_2 \|\mathbf{D}\boldsymbol{\theta}_h\|_1 \qquad (3) \\ &\text{subject to } \ \boldsymbol{y} = (\mathbf{B}_{m,s} \otimes \mathbf{B}_{m,w} \otimes \mathbf{B}_{m,y})\boldsymbol{\theta}_m + (\mathbf{B}_{h,s} \otimes \mathbf{B}_{h,w} \otimes \mathbf{B}_{h,y})\boldsymbol{\theta}_h + \mathbf{e}, \end{aligned}$$

where $\boldsymbol{\theta}_m = \text{vec}(\boldsymbol{\theta}_{m,1}, \ldots, \boldsymbol{\theta}_{m,t}, \ldots, \boldsymbol{\theta}_{m,T})$ and $\boldsymbol{\theta}_h = \text{vec}(\boldsymbol{\theta}_{h,1}, , \ldots, \boldsymbol{\theta}_{h,t}, \ldots, \boldsymbol{\theta}_{h,T})$. The choice of the turning parameters λ_1, λ_2 will be discussed in Section 4.

Note that there are two penalties in equation (3): $\lambda_1 \|\boldsymbol{\theta}_h\|_1$ is the LASSO penalty to control both the sparsity of the hot-spots and $\lambda_2 \|\mathbf{D}\boldsymbol{\theta}_h\|_1$ is the fused LASSO penalty (Tibshirani et al., 2005) to control the temporal consistency of the hot-spots. Traditional algorithms often involve the storage and computation of the matrix $\mathbf{B}_m$ and $\mathbf{B}_h$, which is of the dimension $n_1 n_2 n_3 \times n_1 n_2 n_3$. Thus they might work to solve the optimization problem in equation (3) when the dimensions are small, but they will be computationally infeasible as the dimensions grow. To address this computational challenge, we propose to simplify the computational complexity by modifying the matrix algebra in traditional algorithm into tensor algebra, and will discuss how to optimize the problem in equation (3) computationally efficiently in Section 5.

4 Hot-spot Detection

This section focuses on the detection of the hot-spot, which includes the detection and identification of the year (when), the state (where) and the week (which) of the hot-spots. In our case study, we focus on the upward shift of the number of gonorrhea cases, since the increasing gonorrhea is generally more harmful to the societies and communities. Of course, one can also detect the downward shift with a slight modification of our proposed algorithms by multiplying -1 to the raw data.

For the purpose of easy presentation, we first discuss the detection of the hot-spot, i.e., detect when hot-spot occurs in Subsection 4.1. Then, in Subsection 4.2, we consider the localization of the hot-spot, i.e., determine which states and which weeks are involved for the detected hot-spots.

4.1 Detect When the Hot Spot Occurs

To determine when the hot-spot occurs, we consider the following hypothesis test and set up the control chart for the hot-spot detection (4).

$$H_0 : \widetilde{\mathbf{r}}_t = 0 \quad v.s. \quad H_1 : \widetilde{\mathbf{r}}_t = \delta \widehat{\mathbf{h}}_t \quad (\delta > 0), \tag{4}$$

where $\widetilde{\mathbf{r}}_t$ is the expected residuals after removing the mean. The essence of this test is that, we want to detect whether $\widetilde{\mathbf{r}}_t$ has a mean shift in the direction of $\widehat{\mathbf{h}}_t$, estimated in Section 5. To test this hypotheses, the likelihood ratio test is applied to the residual $\mathbf{r}_t$ at each time t, i.e. $\mathbf{r}_t = \mathbf{y}_t - \boldsymbol{\mu}_t$, where it assumes that the residuals $\mathbf{r}_t$ is independent after removing the mean and its distribution before and after the hot-spot remains the same. Accordingly, the test statistics monitoring upward shift is designed as $P_t^+ = \widehat{\mathbf{h}}_t'^+ \mathbf{r}_t / \sqrt{\widehat{\mathbf{h}}_t'^+ \widehat{\mathbf{h}}_t^+}$ (Hawkins, 1993), where $\widehat{\mathbf{h}}_t^+$ only takes the positive part of $\widehat{\mathbf{h}}_t$ with other entries as zero. Here we put a superscript "+" to emphasis that it aims for upward shift.

The choices of the penalty parameters λ_1, λ_2 are described as follows. In order to select the one with the most power, we propose to calculate a series of P_t^+ under different combination of (λ_1, λ_2) from the set $\Gamma = \{(\lambda_1^{(1)}, \lambda_2^{(1)}) \cdots (\lambda_1^{(n_\lambda)}, \lambda_2^{(n_\lambda)})\}$. For better illustration, we denote the test statistics under penalty parameter (λ_1, λ_2) as $P_t^+(\lambda_1, \lambda_2)$. The test statistics (Zou and Qiu, 2009) with the most power to detect the change, noted as $\widetilde{P}_t^+$, can be computed by

$$\widetilde{P}_t^+ = \max_{(\lambda_1, \lambda_2) \in \Gamma} \frac{P_t^+(\lambda_1, \lambda_2) - E(P_t^+(\lambda_1, \lambda_2))}{\sqrt{Var(P_t^+(\lambda_1, \lambda_2))}}, \tag{5}$$

where $E(P_t^+(\lambda_1, \lambda_2))$, $Var(P_t^+(\lambda_1, \lambda_2))$ respectively are the mean and variance of $P_t(\lambda_1, \lambda_2)$ under H_0 (e.g. for phase-I in-control samples).

Note that the penalty parameter (λ_1, λ_2) to realize the maximization in equation (5) is generally different under different time t. To emphasize such

dependence of time t, denote by $(\lambda^*_{1,t}, \lambda^*_{2,t})$ the parameter pair that attains the maximization in equation (5) at time t, i.e,

$$(\lambda^*_{1,t}, \lambda^*_{2,t}) = \arg \max_{(\lambda_1,\lambda_2)\in\Gamma} \frac{P_t^+(\lambda_1,\lambda_2) - E(P_t^+(\lambda_1,\lambda_2))}{\sqrt{Var(P_t^+(\lambda_1,\lambda_2))}}. \tag{6}$$

Thus, the series of the test statistics for the hot-spot at time t is $\widetilde{P}_t^+(\lambda^*_{1,t}, \lambda^*_{2,t})$ where $t = 1 \cdots T$.

With the test statistic available, we design a control chart based on the CUSUM procedure due to the following reasons: (1) we are interested in detecting the change with the temporal continuity, therefore, aligns with the objective of CUSUM. (2) In the view of social stability, we want to keep gonorrhea at a target value without sudden changes, which makes the CUSUM chart is a natural better fit.

To be more specific, in the CUSUM procedure, we compute the CUSUM statistics recursively by

$$W_t^+ = \max\{0, W_{t-1}^+ + \widetilde{P}_t^+(\lambda^*_{1,t}, \lambda^*_{2,t}) - d\},$$

and $W_{t=0}^+ = 0$, where d is a constant and can be chosen according to the degree of the shift that we want to detect. Next, we set the control limit L to achieve a desirable ARL for in-control samples. Finally, whenever $W_t^+ > L$ at some time $t = t^*$, we declare that a hot-spot occurs at time t^*.

4.2 Localize Where and Which the Hot Spot Occur?

After the hot-spot t^* has been detected by the CUSUM control chart in the previous section, the next step is to localize where and which crime rate may account for this hot-spot. To do so, we propose to utilize the vector

$$\widehat{\mathbf{h}}_{\lambda^*_{1,t^*},\lambda^*_{2,t^*}} = \mathbf{B}_h \widehat{\boldsymbol{\theta}}_{h,\lambda^*_{1,t^*},\lambda^*_{2,t^*}}$$

at the declared hot-spot time t^* and the corresponding parameter $\lambda^*_{1,t^*}, \lambda^*_{2,t^*}$ in equation (6). For the numerical computation purpose, it is often easier to directly work with the tensor format of the hot-spot $\widehat{\mathbf{h}}_{\lambda^*_{1,t^*},\lambda^*_{2,t^*}}$, denoted as $\widehat{\mathcal{H}}_{\lambda^*_{1,t^*},\lambda^*_{2,t^*}}$, which is a tenor of dimension $n_1 \times n_2 \times T$. If the (i, j, t^*)-th entry in $\widehat{\mathcal{H}}_{\lambda^*_{1,t^*},\lambda^*_{2,t^*}}$ is non-zero, then we declare that there is a hot-spot for the j-th crime rate type in the i-th state in t^*-th year.

5 Optimization Algorithm

In this section, we will develop an efficient optimization algorithm for solving the optimization problem in equation (3). For notion convenience, we adjust the notation above a little bit. Because $\boldsymbol{\theta}_m, \boldsymbol{\theta}_h$ in equation (3) is solved under penalty $\lambda_1 R_1(\boldsymbol{\theta}_h) + \lambda_2 R_2(\boldsymbol{\theta}_h)$, we change $\boldsymbol{\theta}_m$, $\boldsymbol{\theta}_h$ into $\boldsymbol{\theta}_{m,\lambda_1,\lambda_2}, \boldsymbol{\theta}_{h,\lambda_1,\lambda_2}$ to emphasis the penalty parameter λ_1 and λ_2. Accordingly, $\boldsymbol{\theta}_{h,0,\lambda_2}$ refers to the estimator only under the second penalty $\lambda_2 R_2(\boldsymbol{\theta}_h)$, i.e,

$$\boldsymbol{\theta}_{h,0,\lambda_2} = \arg\min_{\boldsymbol{\theta}_m, \boldsymbol{\theta}_h} \{\|\mathbf{e}\|_2^2 + \lambda R_2(\boldsymbol{\theta}_h)\}. \tag{7}$$

The structure of this section is that, we first develop the procedure of our proposed method in Subsection 5.1 and then gives the computational complexity in Subsection 5.2.

5.1 Procedure of Our Algorithm

In the optimization problem shown in equation (3), there are two unknown vectors, namely $\boldsymbol{\theta}_{m,\lambda_1,\lambda_2}$, $\boldsymbol{\theta}_{h,,\lambda_1,\lambda_2}$. To simplify the optimization above, we first figure out the close-form correlation between $\boldsymbol{\theta}_{m,\lambda_1,\lambda_2}$ and $\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}$. Then, we solve the optimization by modifying the matrix algebra in FISTA(Beck and Teboulle, 2009) into tensor algebra. The key to realize it is the proximal mapping of $\lambda_1 R_1(\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}) + \lambda_2 R_2(\boldsymbol{\theta}_{h,\lambda_1,\lambda_2})$. To address it, we first aims at the proximal mapping of $\lambda_2 R_2(\boldsymbol{\theta}_{h,0,\lambda_1})$, where SFA via gradient descent (Liu et al., 2010) is used. And then the proximal mapping of $\lambda_1 R_1(\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}) + \lambda_2 R_2(\boldsymbol{\theta}_{h,\lambda_1,\lambda_2})$ can be solved with a close-form correlation between it and the proximal mapping of $\lambda_2 R_2(\boldsymbol{\theta}_{h,0,\lambda_2})$.

There are three subsections in this section, where each subsection represents one step in our proposed algorithm.

5.1.1 Estimate the mean parameter

To begin with, we first simplify the optimization problem in equation (3), i.e., figure out the close-form correlation between $\boldsymbol{\theta}_{m,\lambda_1,\lambda_2}$ and $\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}$.

Although there are two sets of parameters $\boldsymbol{\theta}_{m,\lambda_1,\lambda_2}$ and $\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}$ in the model, we note that given $\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}$, the parameter $\boldsymbol{\theta}_{m,\lambda_1,\lambda_2}$ is involved in the standard least squared estimation and thus can be solved in the closed-form solution, see equation (8) in the proposition below.

Proposition 1. Given $\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}$, the closed-form solution of $\boldsymbol{\theta}_{m,\lambda_1,\lambda_2}$ is given by:

$$\boldsymbol{\theta}_{m,\lambda_1,\lambda_2} = (\mathbf{B}'_m\mathbf{B}_m)^{-1}(\mathbf{B}'_m y - \mathbf{B}'_m\mathbf{B}_h\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}). \tag{8}$$

It remains to investigate how to estimate the parameter $\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}$. After plugging in (8) into (3), the optimization problem for estimating $\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}$ becomes

$$\arg\min_{\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}} \|\mathbf{y}^* - \mathbf{X}\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}\|_2^2 + \lambda_1\|\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}\|_1 + \lambda_2\|\mathbf{D}\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}\|_1, \tag{9}$$

where $\mathbf{y}^* = [\mathbf{I} - \mathbf{H}_m]\,\mathbf{y}$, $\mathbf{X} = [\mathbf{I} - \mathbf{H}_m]\,\mathbf{B}_h$ and $\mathbf{H}_m = \mathbf{B}_m(\mathbf{B}'_m\mathbf{B}_m)^{-1}\mathbf{B}'_m$ is the projection matrix.

Due to the high dimension, we need to develop an efficient and precise optimization algorithm to optimize(3). Obviously, (9) is a typical sparse optimization problem. However, most of the sparse optimization frameworks focus on optimizing Eq. (7).

$$\arg\min_{\boldsymbol{\theta}_{h,0,\lambda_2}} \|\mathbf{y}^* - \mathbf{X}\boldsymbol{\theta}_{h,\lambda_1,0}\|_2^2 + \lambda_1\|\boldsymbol{\theta}_{h,\lambda_1,0}\|_1, \tag{10}$$

such as Daubechies et al. (2004), Beck and Teboulle (2009), Friedman et al. (2010) and so on, where iterative updating rule are used base either on the gradient information or the proximal mapping. In most cases, the algorithms above works, however, two challenges occur in our paper:

1. When the dimension of $\mathbf{X}$ (of size $n_1n_2T \times n_1n_2T$) become increasingly large, it is difficult for the computer to store and memorize it.

2. When the penalty term is $\lambda_1\|\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}\|_1 + \lambda_2\|\boldsymbol{D}\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}\|_1$, instead of only $\lambda_1\|\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}\|_1$, direct application of the proximal mapping of $\lambda_1\|\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}\|_1$ is not workable.

Therefore, directly applying these above algorithms(Beck and Teboulle (2009), Daubechies et al. (2004), Friedman et al. (2010)) to our case is not feasible. To extend the existing research, we proposed an iterative algorithm in Algorithm 1 and we explain the approach to solve the proximal mapping of $\lambda_1\|\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}\|_1 + \lambda_2\|\boldsymbol{D}\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}\|_1$ in Section 5.1.2.

5.1.2 Proximal Mapping

The main tool we use to solve the optimization problem in equation (9) is a variation of proximal mapping. Denote that $F(\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}) = \frac{1}{2}\|\mathbf{y}^* - \mathbf{X}\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}\|_2^2$. And in the i-th iteration, the according recursive estimator of $\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}$ is noted as $\boldsymbol{\theta}^{(i)}_{h,\lambda_1,\lambda_2}$. Besides,an auxiliary variable $\boldsymbol{\eta}^{(i)}$ is introduced to update from $\boldsymbol{\theta}^{(i)}_{h,\lambda_1,\lambda_2}$ to $\boldsymbol{\theta}^{(i+1)}_{h,\lambda_1,\lambda_2}$ through

$$\begin{aligned}
\boldsymbol{\theta}^{(i+1)}_{h,\lambda_1,\lambda_2} &= \arg\min_{\boldsymbol{\theta}} F(\boldsymbol{\eta}^{(i)}) + \frac{\partial}{\partial \boldsymbol{\theta}_{h,\lambda_1,\lambda_2}} F(\boldsymbol{\eta}^{(i)}) \left(\boldsymbol{\theta} - \boldsymbol{\eta}^{(i)}\right) + \\
&\quad \lambda_1\|\boldsymbol{\theta}\|_1 + \lambda_2\|\mathbf{D}\boldsymbol{\theta}\|_1 + \frac{L}{2}\|\boldsymbol{\theta} - \boldsymbol{\eta}^{(i)}\|_2^2 \\
&= \arg\min_{\boldsymbol{\theta}} \left[\frac{1}{2}\left[\boldsymbol{\theta} - \left(\boldsymbol{\eta}^{(i)} - \frac{\partial}{L\partial\boldsymbol{\theta}} F(\boldsymbol{\eta}^{(i)})\right)\right]^2 + \lambda_1\|\boldsymbol{\theta}\|_1 + \lambda_2\|\mathbf{D}\boldsymbol{\theta}\|_1\right] \\
&\triangleq \pi^{\lambda_1}_{\lambda_2}(\mathbf{v})
\end{aligned}$$

where $\mathbf{v} = \boldsymbol{\eta}^{(i)} - \frac{\partial}{L\partial\boldsymbol{\theta}} F(\boldsymbol{\eta}^{(i)})$, $\boldsymbol{\eta}^{(i)} = \boldsymbol{\theta}^{(i)}_{h,\lambda_1,\lambda_2} + \frac{t_{i-2}-1}{t_{i-1}}(\boldsymbol{\theta}^{(i)}_{h,\lambda_1,\lambda_2} - \boldsymbol{\theta}^{(i-1)}_{h,\lambda_1,\lambda_2})$ and $t_{-1} = t_0 = 1$, $t_{i+1} = \frac{1+\sqrt{1+4t_i^2}}{2}$

Because it is difficult to solve $\pi^{\lambda_1}_{\lambda_2}(\mathbf{v})$ directly, we aim to solve $\pi^{0}_{\lambda_2}(\mathbf{v})$ first. And proved by Liu et al. (2010), there is a close-form correlation between $\pi^{\lambda_1}_{\lambda_2}(\mathbf{v})$ and $\pi^{0}_{\lambda_2}(\mathbf{v})$, which is shown in Proposition 2.

Proposition 2. The close form relationship between $\pi^{\lambda_1}_{\lambda_2}(\mathbf{v})$ and $\pi^{0}_{\lambda_2}(\mathbf{v})$ is

$$\pi^{\lambda_1}_{\lambda_2}(\mathbf{v}) = \text{sign}(\pi^{0}_{\lambda_2}(\mathbf{v})) \odot \max\{|\pi^{0}_{\lambda_2}(\mathbf{v})| - \lambda_1, 0\}. \tag{11}$$

where $\odot$ is an element-wise product operator.

With the proximal mapping function in Proposition 2, we can now develop

the algorithm shown in Algorithm 1.

Algorithm 1: Iterative updating based on tensor decomposition

Input: $\mathbf{y}^*, \mathbf{B}_s, \mathbf{B}_w, \mathbf{B}_y, \mathbf{D}_s, \mathbf{D}_w, \mathbf{D}_y, K, L, \lambda_1, \lambda_2, L_0, M_1, M_2$

Output: $\boldsymbol{\theta}_{h,\lambda_1,\lambda_2}$

initialization;

$\boldsymbol{\Theta}^{(1)} = \boldsymbol{\Theta}^{(0)}, t_{-1} = 1, t_0 = 1, L = L_0$

for $i = 1 \cdots M_1$ **do**

$\mathcal{N}^{(i)} = \mathcal{N}^{(i)} + \frac{t_{i-2}-1}{t_{i-1}}(\boldsymbol{\Theta}^{(i)} - \boldsymbol{\Theta}^{(i-1)})$

$$\mathcal{V} = \mathcal{N}^{(i)} - \frac{1}{L}\mathcal{N}^{(i)} \times_1 (\mathbf{P}'_s\mathbf{P}_s) \times_2 (\mathbf{P}'_w\mathbf{P}_w) \times_3 (\mathbf{P}'_y\mathbf{P}_y) - \frac{1}{L}\mathcal{Y}^* \times_1 \mathbf{P}'_s \times_2 \mathbf{P}'_w \times_3 \mathbf{P}'_y$$

for $j = 0 \cdots M_2$ **do**

$$\mathcal{G}^{(i)} = \left(\mathcal{Z}^{(j)} \times_1 (\mathbf{D}'_s\mathbf{D}_s) \times_2 (\mathbf{D}'_w\mathbf{D}_w) \times_3 (\mathbf{D}'_y\mathbf{D}_y)\right)) - (\mathcal{V} \times_1 \mathbf{D}_s \times_2 \mathbf{D}_w \times_3 \mathbf{D}_y)$$

$\mathcal{Z}^{(j+1)} = P\left(\mathcal{Z}^{(j)} - \mathcal{G}^{(j)}/L\right)$

end

$\pi^0_{\lambda_2}(\mathcal{V}) = \mathcal{V} - (\mathcal{Z}^{(M_2)}) \times_1 \mathbf{D}_s \times_2 \mathbf{D}_w \times_3 \mathbf{D}_y$

$\pi^{\lambda_1}_{\lambda_2}(\mathcal{V}) = \mathbf{sign}(\pi^0_{\lambda_2}(\mathcal{V})) \odot \mathbf{max}\{\left|\pi^0_{\lambda_2}(\mathcal{V})\right| - \lambda_1, 0\}$

$t_{i+1} = \frac{1+\sqrt{1+4t_i^2}}{2}$

end

$\widehat{\boldsymbol{\Theta}}_{h,\lambda_1,\lambda_2} = \pi^{\lambda_1}_{\lambda_2}(\mathcal{V})$

$\widehat{\boldsymbol{\theta}}_{h,\lambda_1,\lambda_2} = \mathbf{vector}(\widehat{\boldsymbol{\Theta}}_{h,\lambda_1,\lambda_2})$ $\boldsymbol{v} = \mathbf{vector}(\mathcal{V})$

vector$(\cdot)$ is a function that unfolding a order-3 tensor of dimension $n_1 \times n_2 \times n_3$ into a vector $n_1 n_2 n_3$.

5.2 Computational Complexity

This section discusses the computational complexity of our proposed algorithm. Suppose the raw data is structured into a tensor of order three with dimensional $n_1 \times n_2 \times n_3$, then the computation complexity of our propose

method is of order $O\left(n_1 n_2 n_3 \max\{n_1, n_2, n_3\}\right)$ (see Proposition 3).

Proposition 3. The computational complexity of Algorithm 1 is of order $O\left(n_1 n_2 n_3 \max\{n_1, n_2, n_3\}\right)$.

Proof. The main computational load in Algorithm 1 is on the calculation of $\mathbf{v}$ (line 4), $\mathbf{g}^{(i)}$(line 5) and $\pi^0_{\lambda_2}(\mathbf{v})$ (line 7). We will take the calculation of $\mathbf{v}$ in line 4 in the algorithm as an example. To begin with, we focus on the computational complexity of

$$\mathcal{N}^{(i)} \times_1 (\mathbf{P}'_s \mathbf{P}_s) \times_2 (\mathbf{P}'_w \mathbf{P}_w) \times_3 (\mathbf{P}'_y \mathbf{P}_y)). \tag{12}$$

For better illustration, we denote tensor$(\boldsymbol{\eta}^{(i)})$ as $\mathcal{N}^{(i)}$ and $\mathcal{N}^{(i)} \times_1 (\mathbf{P}'_s \mathbf{P}_s)$ as tensor $\mathcal{L}_1$. According to the tensor algebra (Kolda and Bader, 2009, Section 2.5),

$$\mathcal{L}_1 = \mathcal{N}^{(i)} \times_1 (\mathbf{P}'_s \mathbf{P}_s) \Longleftrightarrow \mathcal{L}_{1(1)} = \mathbf{P}'_s \mathbf{P}_s \mathcal{N}^{(i)}_{(1)}.$$

Therefore, the computational complexity of equation (12) is the same as two-matrix multiplication with order $n_1 \times n_1$ and $n_1 \times n_1 n_2$, which is of order $O\left(n_1 n_2 n_3 (2n_1 - 1)\right)$.

After the calculation of $\mathcal{L}_1$, equation (12) is reduced to

$$\mathcal{L}_1 \times_2 (\mathbf{P}'_w \mathbf{P}_w) \times_3 (\mathbf{P}'_y \mathbf{P}_y)). \tag{13}$$

Similarly, denotes $\mathcal{L}_2 = \mathcal{L}_1 \times_2 (\mathbf{P}'_w \mathbf{P}_w)$, then

$$\mathcal{L}_2 = \mathcal{L}_1 \times_2 (\mathbf{P}'_w \mathbf{P}_w) \Longleftrightarrow \mathcal{L}_{2(2)} = \mathbf{P}'_w \mathbf{P}_w \mathcal{N}_{(2)}.$$

Therefore, the computational complexity of equation (13) is the same as two-matrix multiplication with order $n_2 \times n_2$ and $n_2 \times n_1 n_3$, which is of order $O\left(n_1 n_2 n_3 (2n_2 - 1)\right)$.

After the calculation of $\mathcal{L}_2$, equation (13) is reduced to

$$\mathcal{L}_2 \times_3 (\mathbf{P}'_y \mathbf{P}_y)). \tag{14}$$

Similarly, denotes $\mathcal{L}_3 = \mathcal{L}_2 \times_2 (\mathbf{P}'_y \mathbf{P}_y)$, then

$$\mathcal{L}_3 = \mathcal{L}_2 \times_3 (\mathbf{P}'_y \mathbf{P}_y) \Longleftrightarrow \mathcal{L}_{3(3)} = \mathbf{P}'_w \mathbf{P}_w \mathcal{N}_{(3)}.$$

Therefore, the computational complexity of equation (13) is the same as two-matrix multiplication with order $n_3 \times n_3$ and $n_3 \times n_1 n_2$, which is of order $O\left(n_1 n_2 n_3 (2n_3 - 1)\right)$.

By combining all these blocks built above, we conclude that the computational complexity of equation (12) is of order $O(n_1 n_2 n_3 (\max\{n_1, n_2, n_3\}))$.

In the same way, the computational complexity in line 5 and 7 of Algorithm 1 is also of order $O(n_1 n_2 n_3 (\max\{n_1, n_2, n_3\}))$. Thus, the computational complexity of Algorithm is of order $O(n_1 n_2 n_3 (\max\{n_1, n_2, n_3\}))$. □

6 Simulation

In this section, we conduct simulation studies to evaluate our proposed methodologies by comparing with several benchmark methods in the literature. The structure of this section is as follows. We first present the data generation mechanism for our simulations in Subsection 6.1, then discuss the performance of hot-spot detection and localization in Subsection 6.2.

6.1 Generative Model in Simulation

In our simulation, at each time index $t(t = 1 \cdots T)$, we generate a vector $\mathbf{y}_t$ of length $n_1 n_2$ by

$$\mathbf{y}_{i,t} = (\mathbf{B}\boldsymbol{\theta}_t)_i + \delta \mathbb{1}\{t \geq \tau\} \mathbb{1}_i\{i \in S_h\} + \mathbf{w}_{i,t}, \tag{15}$$

where $\mathbf{y}_{i,t}$ denotes the i-th entry in vector $\mathbf{y}_t$, $(\mathbf{B}\boldsymbol{\theta}_t)_i$ denotes the i-th entry in vector $\mathbf{B}\boldsymbol{\theta}_t$, and δ denotes the change magnitude. Here $\mathbb{1}(A)$ is the indicator function, which has the value 1 for all elements of A and the value 0 for all elements not in A, and $\mathbf{w}_{i,t}$ is the i-th entry in the white noise vector whose entries are independent and follow $N(0, 0.1^2)$ distribution.

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 week may lead to the occurrence of temporal hot-spot. Because the baseline methods, PCA and T2, can only realize the detection of temporal changes, we only show the localization of spatial hot-spot by SSR-Tensor, SSD (Yan et al., 2018), ZQ lasso (Zou and Qiu, 2009). For the anomaly setup, $\mathbb{1}\{t \geq \tau\}$ indicates that the spatial hot-spots only occur after the temporal hot-spot τ. This ensures that the simulated hot-spot is temporal consistent. The second indicator function $\mathbb{1}_i\{i \in S_h\}$ shows that only those entries whose location index belongs set S_h are assigned as local hot-spots. This ensures that the simulated hot-spot is sparse. Here we assume the change happens at $\tau = 50$ among total $T = 100$ years. And the spatial

hot-spots index set is formed by the combination of states Conn, Ohio, West Va, Tex, Hawaii and week from 1-10 and 41-51.

To match the dimension in the case study, we choose $n_1 = 50, n_2 = 51$. As for the three terms on the right side of equation (15), they serve for the global trend mean, local sparse anomaly and white noise respectively. In our simulation, the matrix $\mathbf{B}$ is $\mathbf{B}_{m,s} \otimes \mathbf{B}_{m,w} \otimes \mathbf{B}_{m,y}$ with the same choice as that in Section 3.2.

Besides, in each of these two scenarios, we further consider two sub-cases, depending on the value of change magnitude δ in equation (15): one is $\delta = 0.1$ (small shift) and the other is $\delta = 0.5$ (large shift).

6.2 Hot-spot Detection Performance

In this section, we compare the performance of our proposed method (denoted as 'SSR-tensor') for detection of hot-spot with some benchmark methods. Specifically, we compare our proposed method with Hotelling T^2 control chart (Qiu, 2013) (denoted as 'T2'), LASSO-based control chart proposed by Zou and Qiu (2009) (denoted as 'ZQ LASSO'), PCA-based control chart proposed by De Ketelaere et al. (2015) (denoted as 'PCA') and SSD proposed by Yan et al. (2018) (denoted as 'SSD'). Note that there are two main differences between our SSR-tensor method and the SSD method in Yan et al. (2018). First, SSR-Tensor has the autoregressive or fussed LASSO penalty in equation (3) so as to ensure the temporal continuity of the hot-spot. Second, SSD uses the Shewhart control chart to monitor temporal changes, while SSR-Tensor utilizes CUSUM instead, which is more sensitive for a small shift.

For the basis choices of our proposed method, to model the spatial structure of the global trend, we choose $\mathbf{B}_{m,1}$ as the kernel matrix to describe the smoothness of the background, whose (i, j) entry is of value $\exp\{-d^2/(2c^2)\}$ where d is the distance between the i-th state and j-th state and c is the bandwidth chosen by cross-validation. In addition, we choose identical matrices for the yearly basis and weekly basis since we do not have any prior information. Moreover, we use the identity matrix for the spatial and temporal basis of the hot-spots. For SSD in Yan et al. (2018), we will use the same spatial and temporal basis in order to have a fair comparison.

For evaluation, we will compute the following four criteria: (i) precision, defined as the proportion of detected anomalies that are true hot-spots; (ii) recall, defined as the proportion of the anomalies that are correctly identi-

methods	small shift $\delta = 0.1$				large shift $\delta = 0.5$			
	precision	recall	F measure	ARL	precision	recall	F measure	ARL
SSR-tensor	**0.0824**	**0.9609**	**0.5217**	**1.6420**	**0.0822**	**0.9633**	**0.5228**	**1.0002**
	(0.0025)	(0.0536)	(0.0270)	(0.7214)	(0.0022)	(0.0549)	(0.0277)	(0.0144)
SSD	0.0404	0.9820	0.5112	7.4970	0.0412	1.0000	0.5206	1.0000
	(0.0055)	(0.1330)	(0.0692)	(9.4839)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
ZQ LASSO	0.0412	1.000	0.5206	9.5890	0.0412	1.0000	0.5206	8.8562
	(0.0000)	(0.0000)	(0.0000)	(7.5414)	(0.0000)	(0.0000)	(0.0000)	(7.1169)
PCA	-	-	-	28.7060	-	-	-	32.0469
	-	-	-	(16.9222)	-	-	-	(17.4660)
T2	-	-	-	50.0000	-	-	-	50.0000
	-	-	-	(0.0000)	-	-	-	(0.0000)

Table 1: Scenario 1 (decreasing global trend): Comparison of hot-spot detection under small shift and large shift

fied; (iii) F measure, a single criterion that combines the precision and recall by calculating their harmonic mean; and (iv) the corresponding average run length (ARL_1), a measure on the average detection delay in the special scenario when the change occurs at time $t = 1$. All simulation results below are based on 1000 Monte Carlo simulation replications.

Table 1 shows the merits of our methodology mainly lies on the higher precision and shorter ARL_1. For example, when the shift is very small, i.e., $\delta = 0.1$, the ARL_1 of our SSR-Tensor method is only 1.6420 compared with 7.4970 of SSD and 9.5890 of ZQ-LASSO. The reason for SSR-Tensor has shorter ARL_1 than that of SSD is that, SSD use Shewhart control chart to detect temporal changes, which make it insensitive for a small shift. While for SSR-Tensor, it applies the CUSUM control chart, which is capable to detect the shift of small size. The reason for both SSR-Tensor and SSD have shorter ARL_1 than that of ZQ-LASSO, PCA and T2 is that ZQ-LASSO fails to capture the global trend mean. Yet, the data generated in our simulation has both decreasing and circular global trend, which makes it hard for ZQ-LASSO to model well.

7 Case Study

In this section, we apply our proposed SSR-tensor model and hot-spot detection/localization method to the weekly gonorrhea dataset in Section 2. For the purpose of comparison, we also consider other benchmark methods mentioned in Section 6), and consider two performance criteria: one is the temporal detection of hot-spots (i.e., which year it occurs) and the other

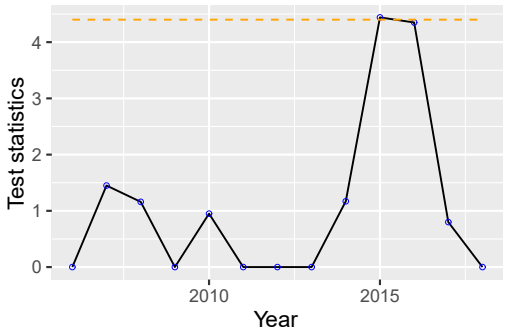

Figure 4: CUSUM Control chart of gonorrhea dataset during years 2006-2018.

is the localization of the hot-spots (i.e., which state and which week might involve the alarm).

7.1 When the temporal changes happen?

Here we consider the performance on the temporal detection of hot-spots of our proposed method and other benchmark methods. For our proposed SSR-Tensor method, we build a CUSUM control chat utilizing the test statistic in Subsection 4.1, which is shown in Figure 4. From this plot, we can see that the hot-spots are detected at 10-th year, i.e., 2016.

For the purpose of comparison, we also apply the benchmark methods, SSD (Yan et al., 2018), ZQ LASSO (Zou and Qiu, 2009), PCA (De Ketelaere et al., 2015) and T2(Qiu, 2013), into the gonorrhea dataset. Unfortunately, all benchmark methods are unable to raise any alarms, but our proposed SSR-tensor method raises the first hot-spot alarm in year 2016.

7.2 Which state and week the spatial hot-spots occur?

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 week may lead to the occurrence of temporal hot-spot. Because the baseline methods, SSD, ZQ-LASSO, PCA, and T2, can only realize the detection of temporal changes, we only show the localization of spatial hot-spot by SSR-Tensor, which is visualized in Figure 5.

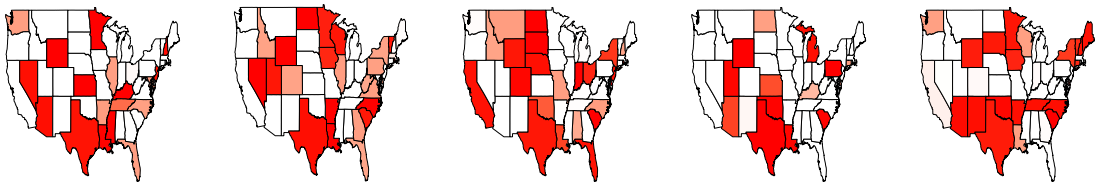

Figure 5: Hot-spot detection result of circular pattern of W.S. CENTRAL(Arkansas, Louisiana, Oklahoma, Texas)

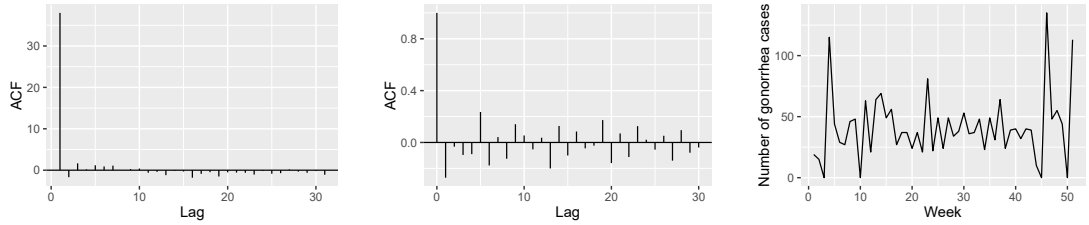

Figure 6: Auto-correlation of all US (left) & Kans.(middle) in 2016 and time series plot of Kansas in 2016 (right)

There are some circular patterns in specific areas. For example, CENTRAL(Ark, La, Okla, Tex) tends to have a circular pattern every 11 weeks, which is shown in Figure 5 . Besides, there are also some circular pattern for a certain state, for instance, Kansas has the bi-weekly pattern as shown in Figure 6. To validate the bi-weekly circular pattern of Kansas, we plot the time series plot of Kansas in 2016 as well as the auto-correlation function plot in Figure 5. Besides, the auto-correlation function plot in the left panel of Figure 6 serves as a baseline. It can be seen from the middle and right plot of Figure 6 that, Kansas has some bi-weekly or tri-weekly circular pattern.

References

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202.

Call, M. A. and Voss, P. R. (2016). Spatio-temporal dimensions of child poverty in america, 1990–2010. *Environment and Planning A*, 48(1):172–191.

Daubechies, I., Defrise, M., and De Mol, C. (2004). An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11):1413–1457.

De Ketelaere, B., Hubert, M., and Schmitt, E. (2015). Overview of pca-based statistical process-monitoring methods for time-dependent, high-dimensional data. *Journal of Quality Technology*, 47(4):318–335.

Diggle, P. J. (2013). *Statistical analysis of spatial and spatio-temporal point patterns*. CRC Press.

Elhorst, J. P. (2014). Spatial panel data models. In *Spatial econometrics*, pages 37–93. Springer.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1.

Hamilton, J. D. (1994). *Time series analysis*, volume 2. Princeton university press Princeton, NJ.

Hannan, E. J. and Quinn, B. G. (1979). The determination of the order of an autoregression. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 190–195.

Hawkins, D. M. (1993). Regression adjustment for variables in multivariate quality control. *Journal of Quality Technology*, 25(3):170–182.

Hu, K. and Yuan, J. (2009). Batch process monitoring with tensor factorization. *Journal of Process Control*, 19(2):288–296.

Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review*, 51(3):455–500.

Lai, T. L. and Lim, J. (2015). Asymptotically efficient parameter estimation in hidden markov spatio-temporal random fields. *Statistica Sinica*, pages 403–421.

Lan, H., Zhou, C., Wang, L., Zhang, H., and Li, R. (2004). Landslide hazard spatial analysis and prediction using gis in the xiaojiang watershed, yunnan, china. *Engineering geology*, 76(1-2):109–128.

Lichstein, J. W., Simons, T. R., Shriner, S. A., and Franzreb, K. E. (2002). Spatial autocorrelation and autoregressive models in ecology. *Ecological monographs*, 72(3):445–463.

Liu, J., Yuan, L., and Ye, J. (2010). An efficient algorithm for a class of fused lasso problems. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–332. ACM.

Louwerse, D. and Smilde, A. (2000). Multivariate statistical process control of batch processes based on three-way models. *Chemical Engineering Science*, 55(7):1225–1235.

Qiu, P. (2013). *Introduction to statistical process control*. Chapman and Hall/CRC.

Reynolds, K. and Madden, L. (1988). Analysis of epidemics using spatio-temporal autocorrelation. *Phytopathology*, 78(2):240–246.

Šaltytė Benth, J. and Šaltytė, L. (2011). Spatial–temporal model for wind speed in lithuania. *Journal of Applied Statistics*, 38(6):1151–1168.

Tran, L., Navasca, C., and Luo, J. (2012). Video detection anomaly via low-rank and sparse decompositions. In *2012 Western New York Image Processing Workshop*, pages 17–20. IEEE.

Yan, H., Paynabar, K., and Shi, J. (2017). Anomaly detection in images with smooth background via smooth-sparse decomposition. *Technometrics*, 59(1):102–114.

Yan, H., Paynabar, K., and Shi, J. (2018). Real-time monitoring of high-dimensional functional data streams via spatio-temporal smooth sparse decomposition. *Technometrics*, 60(2):181–197.

Zhu, J., Huang, H.-C., and Wu, J. (2005). Modeling spatial-temporal binary data using markov random fields. *Journal of Agricultural, Biological, and Environmental Statistics*, 10(2):212.

Zou, C., Ning, X., and Tsung, F. (2012). Lasso-based multivariate linear profile monitoring. *Annals of Operations Research*, 192(1):3–19.

Zou, C. and Qiu, P. (2009). Multivariate statistical process control using lasso. *Journal of the American Statistical Association*, 104(488):1586–1596.