



Paper Type: Original Article

Covid-19 Pandemic Data Analysis Using Tensor Methods

Dipak Dulal^{1,*} , Ramin Goudarzi Karim² and Carmeliza Navasca³

¹Department of Mathematics, University of Alabama at Birmingham, Birmingham, AL, USA; dpdulal@uab.edu;

²Department of Computational and Information Sciences, Stillman College, Tuscaloosa, AL, USA; rkarim@stillman.edu;

³Department of Mathematics, University of Alabama at Birmingham, Birmingham, AL, USA; cnavasca@uab.edu;

Citation:

Received: 03 January 2024	Dulal, D., Goudarzi Karim, R., and Navasca, C. (2024). Covid-19
Revised: 21 February 2024	Pandemic Data Analysis Using Tensor Methods. Computational
Accepted: 01/03/2024	Algorithms and Numerical Dimensions, 3(1), 17-44.

Abstract

In this paper, we use tensor models to analyze the Covid-19 pandemic data. First, we use tensor models, canonical polyadic, and higher-order Tucker decompositions to extract patterns over multiple modes. Second, we implement a tensor completion algorithm using canonical polyadic tensor decomposition to predict spatiotemporal data from multiple spatial sources and to identify Covid-19 hotspots. We apply a regularized iterative tensor completion technique with a practical regularization parameter estimator to predict the spread of Covid-19 cases and to find and identify hotspots. Our method can predict weekly, and quarterly Covid-19 spreads with high accuracy. Third, we analyze Covid-19 data in the US using a novel sampling method for alternating least-squares. Moreover, we compare the algorithms with standard tensor decompositions concerning their interpretability, visualization, and cost analysis. Finally, we demonstrate the efficacy of the methods by applying the techniques to the New Jersey Covid-19 case tensor data.

Keywords: tensor, tensor completion, tensor decomposition, Covid-19, spatiotemporal data

1|Introduction

Tensor decomposition is a powerful tool in data analysis, computer vision, scientific computing, machine learning, and many other fields. Tensor models for dimensionality reduction have been highly influential in machine



Corresponding Author: Dipak Dulal



<https://doi.org/10.22105/cand.2024.450017.1092>



Licensee System Analytics. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).

learning applications like classification and regression. Tensor decomposition has succeeded in many modern big data analyses [9, 7, 21, 29], Neuro-Science [4].

In this work, we focus on analyzing Covid-19 pandemic data [25, 1] by using tensor decomposition models. Our goals are to predict future infection and locate and identify hotspots from data-set coming from multiple sources across different spatial regions over time. One is interested in determining where and when changes occur in the pattern. The strategy is set up with an optimization that separates the data as follows: let $\mathcal{Y} = \mathcal{L} + \mathcal{S}$ where $\mathcal{Y}$ is the given tensor, $\mathcal{L}$ is a low-rank reconstructed tensor of $\mathcal{Y}$ and $\mathcal{S}$ is the sparse tensor. In video processing, the original video is separated into its background and foreground subspace to detect anomalous activities [9, 29]. The tensor $\mathcal{L}$ is the background, and $\mathcal{S}$ is the foreground. The sparse tensor $\mathcal{S}$ can provide anomalous activities. Similarly, $\mathcal{S}$ will contain the occurrence of hotspots. To achieve this separation, we formulate the following:

$$\min_{a_r, b_r, c_r, \alpha_r} \|\mathcal{C} - \mathcal{L}\|_F^2 + \sigma \|\alpha\|_{\ell_1} \quad (1)$$

where $\mathcal{L} = \sum_{r=1}^R \alpha_r a_r \circ b_r \circ c_r$, $\|\cdot\|_F$ is the tensor Frobenius norm, and $\|\cdot\|_{\ell_1}$ is the vector one norm. The optimization problem (1) is neither convex nor differentiable [10]. In addition, this formulation is amenable to tensor completion problems where missing data can be found. To show the efficacy of our methods for forecasting Covid-19 infections, we use the Covid-19 database in [25].

Most of the algorithms suggested for solving (1) are based on ALS (alternating least squares) [10]. ALS is fast, easy to program, and effective. However, ALS has some drawbacks [18]. There is a need for more efficient and accurate methods, especially in higher order tensors with high modal dimensions. Thus, we propose a sampling method for ALS to maximize its efficiency of ALS (especially for the time-cost minimization) and to allow more considerable tensor data.

1.1|Previous Work on Covid-19 Data Analysis

Previous Work on Covid-19 Data Analysis. Here, we mention a non-exhaustive collection of literature on Covid-19 analysis. The work varies from PDE models to machine learning methods for predicting Covid-19 hotspots, patterns, and outbreaks. A parabolic PDE-based predictive model with parameters learned from training data from previous Covid-19 cases has been proposed to predict Covid-19 infections in Arizona [27]. A global optimization of the tensor train is used to explore the parameter space to locate the starting points. The Nelder-Mead simplex, local optimization algorithm is used to optimize the covid-19 modeling problem locally. Eventually, the Runge-Kutta method was applied to solve the PDE for one-step forward prediction. There are several machine learning methods. In supervised machine learning, time series forecasting via Holt's Winter model was used to analyze global Covid-19 data and predict the sum of global Covid-19 cases compared to linear regression and support vector algorithms. Dictionary learning [17] through nonnegative matrix factorization to identify the pattern and predict future covid-19 outbreaks. In addition, machine learning has been applied to a susceptible exposed-infected-removed (SEIR) model with SARS 2003 training data to predict covid-19 outbreaks [30]. Deep learning models such as the attention-based encoder-decoder model [19] have been implemented to forecast the epidemic of covid-19. There needs to be more work on mathematical modeling to detect Covid-19 hotspots; however, a tensor-based anomaly detection method for spatial-temporal data [31] has been proposed to determine hotspots based on an anomaly in pattern.

1.2|Contributions

In this work, we focus on analyzing Covid-19 infection data [25, 1] from New Jersey period 04/01/2020 to 12/26/2021 (NJ-Covid-19). The state of New Jersey was initially chosen since we would like to study the spread of the disease in the most densely populated state. The raw data collected by the New York Times [25, 1] were the cumulative data daily. Preprocessing techniques were applied to the raw spatio-temporal data, and formatting the data in a tensor structure. Standard low-rank tensor decomposition models such as canonical polyadic (CP), Higher Order Orthogonal Iteration (HOOI), as well as tensor rank revealing methods such as the tensor rank approximation method called low-rank approximation of tensor (LRAT) [7] and LRAT with flexible Golub-Kahan [29] are used to approximate the pattern and flow of covid infections. A new sampling method for ALS (SMALS) was applied to the NJ-Covid-19 data. Some of the other contributions are the following:

- Converted each entry of the Covid-19 tensor into the increment rate each week, applied the CP-ALS algorithm, and plotted error with practical threshold using standard deviation and mean. The spike above the threshold line replicates the original spike in Covid-19 infections.
- Tensor completion optimization was formulated to predict future Covid-19 infections up to the next week and quarter using LRAT with flexible Golub-Kahan.
- The proposed SMALS was implemented on the NJ-Covid-19 data. We compare its numerical results with the standard CP decomposition model via the alternating least-squares method to see its efficacy. We tested our spatiotemporal data tensor and other image data of different sizes.

1.3|Outline of the paper

The paper is organized as follows. Section 2 provides some tensor backgrounds, standard tensor decompositions, CP, and well-known numerical techniques like Alternating Least-Squares and Higher-Order Orthogonal Iteration. Then, in Section 3, we include explorations and estimations of the Covid-19 tensor via CP decomposition and Higher Order Orthogonal Iteration (HOOI) and compare and contrast their outputs. Section 4 describes a sampling method for ALS with some numerical results comparison with ALS. Section 5 deals with the sparse tensor model implemented with LRAT with Golub-Kahan [29]. We discuss its application to predicting Covid-19 infection cases and locating and identifying Covid-19 hotspots. Finally, we provide concluding remarks and some future outlooks.

2|Preliminaries

We denote a vector by a bold lower-case letter $\mathbf{a}$. The bold upper-case letter $\mathbf{A}$ represents a matrix, and the symbol of a tensor is a calligraphic letter $\mathcal{A}$. Throughout this paper, we focus on third-order tensors $\mathcal{A} = (a_{ijk}) \in \mathbb{R}^{I \times J \times K}$ of three indices $1 \leq i \leq I$, $1 \leq j \leq J$ and $1 \leq k \leq K$, but these can be easily extended to tensors of arbitrary order greater or equal to three.

The three kinds of matricization for third-order $\mathcal{A}$ are $A_{(1)}$, $A_{(2)}$ and $A_{(3)}$, according to respectively arranging the column, row, and tube fibers to be columns of matrices, which are defined by fixing every index but one and denoted by $a_{:jk}$, $a_{i:k}$ and $a_{ij\cdot}$, respectively. We also consider the vectorization for $\mathcal{A}$ to obtain a row vector $\mathbf{a}$ such the elements of $\mathcal{A}$ are arranged according to k varying faster than j and j varying faster than i , i.e., $\mathbf{a} = (a_{111}, \dots, a_{11K}, a_{121}, \dots, a_{12K}, \dots, a_{1J1}, \dots, a_{1JK}, \dots)$. Kronecker Product of two matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$ is denoted as $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{mp \times nq}$ and obtained as the product of each element of $\mathbf{A}$ and the matrix $\mathbf{B}$. Khatri-Rao Product of two matrices $\mathbf{A}$ and $\mathbf{B}$ with the same columns are the column-wise Kronecker product: $\mathbf{A} \odot \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1 \dots \mathbf{a}_R \otimes \mathbf{b}_R]$. The outer product of a rank-one third order tensor is denoted as $\mathbf{a} \circ \mathbf{b} \circ \mathbf{c} \in \mathbb{R}^{I \times J \times K}$ of three nonzero vectors $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ is a rank-one tensor with elements $a_i b_j c_k$ for all the indices, i.e., the matricizations of $\mathbf{a} \circ \mathbf{b} \circ \mathbf{c} \in \mathbb{R}^{I \times J \times K}$ are rank-one matrices. A canonical polyadic (CP) decomposition of $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$ expresses $\mathcal{A}$ as a sum of rank-one outer products:

$$\mathcal{A} = \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \quad (2)$$

where $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$, $\mathbf{c}_r \in \mathbb{R}^K$ for $1 \leq r \leq R$ and $\circ$ is the outer product. The outer product $\mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$ is a rank-one component and the integer R is the number of rank-one components in tensor $\mathcal{A}$. The minimal number R such that the decomposition (2) holds is the rank of tensor $\mathcal{A}$, which is denoted by $\text{rank}(\mathcal{A})$. For any tensor $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$, $\text{rank}(\mathcal{A})$ has an upper bound $\min\{IJ, JK, IK\}$ [11]. In fact, tensor rank is NP-hard over $\mathbb{R}$ and $\mathbb{C}$ [6].

Another standard tensor decomposition is higher-order Tucker (HOT) decomposition. HOT is a generalization of matrix SVD where $m \times n$ matrix M has a factorization $U\Sigma V^T$, where U and V are orthogonal matrices and Σ is a diagonal matrix with $\sigma_{ij} = 0$ if $i \neq j$ and otherwise $\sigma_{ii} \geq 0$. Its generalization in third-order tensors is

$$\mathcal{T} = \mathcal{G} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)}$$

where $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is the given tensor, $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ is the core tensor and $U^{(i)} \in \mathbb{R}^{I_i \times R_i}$ for $i = 1, 2, 3$ is an orthogonal matrix. The Tucker contracted product $\times_1$ is defined as $\mathcal{G} \times_1 U^{(1)} = \sum_{r_1} \mathcal{G}_{r_1 r_2 r_3} U_{i_1 r_1}^{(1)} \in \mathbb{R}^{I_1 \times R_2 \times R_3}$.

2.1|Standard Least-Squares Optimization for Tensor Decomposition

Tensor decompositions like CP and HOOI [3, 10] are considered to be generalizations of the singular value decomposition (SVD) and principal component analysis (PCA) of a matrix. To achieve CP from a given third order $\mathcal{X}$, an optimization problem is solved to find the finite sum of rank one tensor of 3^{rd} order approximating $\mathcal{X}$:

$$\min_{A,B,C} \|\mathcal{X} - \hat{\mathcal{X}}\|_F \quad (3)$$

with

$$\hat{\mathcal{X}} = \sum_{r=1}^R a_r \circ b_r \circ c_r \quad (4)$$

where A, B, C are factor matrices with a_r, b_r, c_r are their respective column vectors. This nonlinear optimization can be divided into subproblems of linear least squares with respect to the factor matrices. This is called the Alternating Least Squares (ALS).

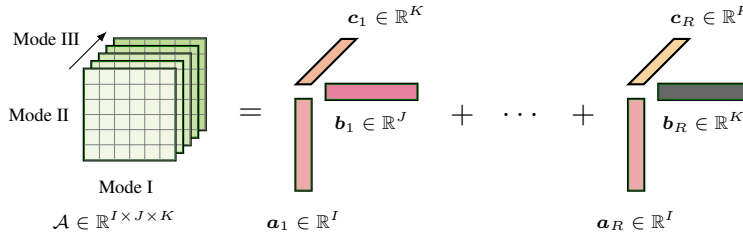


FIGURE 1. CP-Decomposition Architecture

ALS is an iterative method for finding the CP decomposition of a given tensor. The nonlinear optimization problem is:

$$\min_{A,B,C} \left\| \mathcal{X} - \sum_{r=1}^R a_r \circ b_r \circ c_r \right\|_F^2 \quad (5)$$

where A, B and C are factor matrices containing the columns $a_r \in \mathcal{R}^I$, $b_r \in \mathcal{R}^J$ and $c_r \in \mathcal{R}^K$ respectively. The problem can be reduced to linear least squares problems at each iteration with an initial guess A^0, B^0, C^0 , the sequences A^k, B^k, C^k are generated by solving each sub-problems [13, 32]. Given the

initial guess A^0, B^0, C^0 , the factor matrices are updated by the following: update A via

$$A^{k+1} = \arg \min_{A \in \mathcal{R}^{I \times R}} \frac{1}{2} \| X_{(1)} - A(C^k \odot B^k)^T \|_F^2 \quad (6)$$

update B via

$$B^{k+1} = \arg \min_{B \in \mathcal{R}^{J \times R}} \frac{1}{2} \| X_{(2)} - B(C^k \odot A^k)^T \|_F^2 \quad (7)$$

and update C via

$$C^{k+1} = \arg \min_{C \in \mathcal{R}^{K \times R}} \frac{1}{2} \| X_{(3)} - C(B^k \odot A^k)^T \|_F^2 \quad (8)$$

These updating schemes are repeated until convergence; See figure 1, Algorithm 1. The local convergence and uniqueness of the ALS technique were discussed in this work [26].

Algorithm 1 [20]CP-ALS for 3-Way Tensor Decomposition

Require: Tensor $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$ rank R , Maximum Iterations N .

Ensure: CP Decomposition $\lambda \in \mathbb{R}^{R \times 1}$, $A \in \mathbb{R}^{I \times R}$, $B \in \mathbb{R}^{J \times R}$, $C \in \mathbb{R}^{K \times R}$.

- 1: Initialize A, B, C ;
 - 2: for $i = 1 \dots N$ DO
 - 3: $A \leftarrow X_{(1)}(C \odot B)(C^T C * B^T B)^\dagger$
 - 4: Normalize the columns of A (storing norms in vector λ .)
 - 5: $B \leftarrow X_{(2)}(C \odot A)(C^T C * A^T A)^\dagger$
 - 6: Normalize the columns of B (storing norms in vector λ .)
 - 7: $C \leftarrow X_{(3)}(B \odot A)(B^T B * A^T A)^\dagger$
 - 8: Normalize the columns of C (storing norms in vector λ .)
 - 9: If convergence is met then,
 - 10: break for loop
 - 11: end if
 - 12: end for
 - 13: return λ, A, B, C ;
-

Algorithm 2 The CP-ALS Algorithm for General Case

Require: Initialize with factor matrices $A^0 \in \mathbb{R}^{I \times R}$, $B^0 \in \mathbb{R}^{J \times R}$, $C^0 \in \mathbb{R}^{K \times R}$

Ensure: Factor matrices $A^{(n)} \in \mathbb{R}^{I_n \times R}$ for $n = 1, 2, \dots, N$.

- 1: for $i = k \dots N$ DO
- 2:

$$(A^{(n)})^{k+1} = \arg \min_{A^{(n)}} \frac{1}{2} \| X_{(n)} - A((A^{(N)})^k \odot \dots \odot (A^{(n+1)})^k \odot (A^{(n-1)})^{k+1} \odot \dots \odot (A^{(1)})^{k+1})^T \|_F^2$$

- 3: end for
 - 4: return $A^{(n)} \in \mathbb{R}^{I_n \times R}$;
-

2.2|Higher Order Orthogonal Iteration (HOOI)

HOOI is an iterative algorithm that computes low rank decomposition of a given tensor [22]. To achieve the reconstruction of $\mathcal{T}$ in the HOOI format, i.e. $\mathcal{T} = \mathcal{G} \times_1 U^{(1)} \times_2 U^{(2)} \dots \times_N U^{(N)}$, the problem formulation is

$$\min_{U^{(1)}, U^{(2)}, \dots, U^{(N)}} \|\mathcal{T} - \mathcal{G} \times_1 U^{(1)} \times_2 U^{(2)} \dots \times_N U^{(N)}\|_F^2$$

where $\mathcal{T} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ and $U^{(i)} \in \mathbb{R}^{I_i \times R_i}$ is a low rank matrix for $i = 1, \dots, N$ with $I_i \geq R_i$. Equivalently, the optimization can be reformulated as a maximization of $\|\mathcal{G}\|_F^2$ with $\mathcal{G}$ is given by

$$\mathcal{G} = \mathcal{A} \times_1 U^{(1)T} \times_2 U^{(2)T} \dots \times_N U^{(N)T} \quad (9)$$

where the core tensor $\mathcal{G} \in \mathbb{R}^{R_1 \times \dots \times R_N}$ tensor for $i = 1, \dots, N$. See Figures 2.

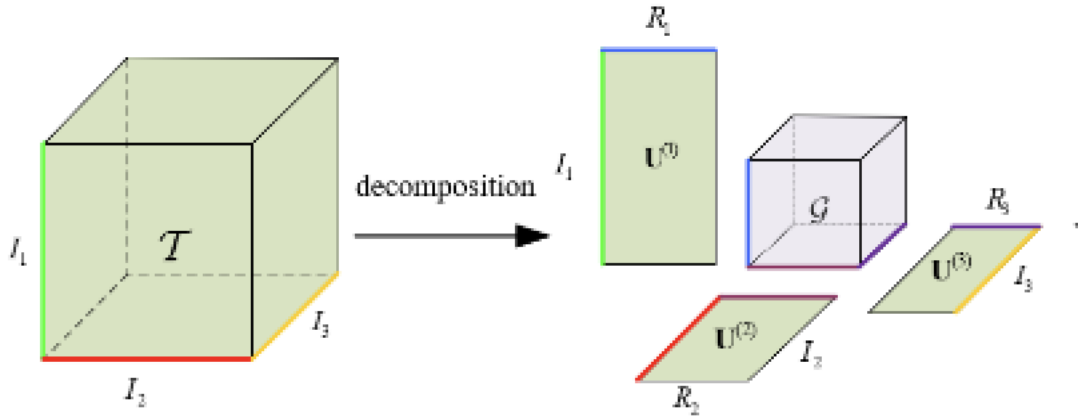


FIGURE 2. HOOI Architecture of 3-Way Tensor [16]

Algorithm 3 [20]Higher-Order Orthogonal Iteration(HOOI)

Require: :Tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ and ranks $R_1, \dots, R_N$

Ensure: :Tucker Factors: $U_1 \in \mathbb{R}^{I_1 \times R_1}, U_2 \in \mathbb{R}^{I_2 \times R_2}, \dots, U_N \in \mathbb{R}^{I_N \times R_N}$

and Core tensor $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \dots \times R_N}$

1: Initialize $U_1, \dots, U_N$; (random or given by HSVD)

2: while convergence criterion not met do

3: for $n = 1, \dots, N$ do

4: $\mathcal{W} \leftarrow \mathcal{X} \times_N U_N^T \dots \times_{n+1} U_{n+1}^T \times_{n-1} U_{n-1}^T \dots \times_1 U_1^T$

5: $[U\Sigma V] \leftarrow SVD(\mathcal{W}_{(n)})$

6: $U_n \leftarrow U(:, 1 : R_n)$

7: end for

8: end while

9: $\mathcal{G} \leftarrow \mathcal{X} \times_N U_N^T \times_{N-1} U_{N-1}^T \dots \times_1 U_1^T$

3|Covid-19 Tensor Analysis Using ALS and HOOI

In this section, we explore the Covid-19 infection data tensor through reconstruction and estimation using CP decomposition and HOOI. We then compare and contrast their outputs.

3.1|Covid-19 Data Preparation

We focus on the daily Covid-19 infection data of New Jersey State from the New York Times's Covid-19 Data Depository from the period of 04/01/2020 to 12/26/2021 [25, 1]. The state of New Jersey was initially chosen since we would like to investigate the spread of the disease in the most densely populated and affected state. The raw data collected by New York Times was a daily basis cumulative data. We restructured the data table into weekly total Covid-19 infections in each county. We stacked 21 matrices representing the counties of New Jersey of size 13 weeks $\times$ 7 quarters. Thus, we construct a tensor data, $\mathcal{C} \in \mathbb{R}^{13 \times 7 \times 21}$. Each element of the tensor represents the total infection in a week of a particular quarter in a county. See Figure 3 below.

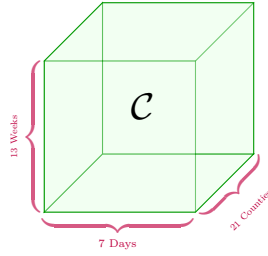


FIGURE 3. Covid-19 Tensor: New Jersey's number of weekly Covid-19 cases every quarter per county.

To increase the accuracy and efficacy of the algorithms, we normalized the data tensor converting it into relative cases tensor with respect to the population of the respective county. We constructed the population tensor of New Jersey collecting population data from US Census 2020 [1] and divided the tensor-data tensor by the population data tensor. Each element of our new normalized tensor is the following:

$$C(i, j, k) = \frac{\text{Total infections in } j^{th} \text{ week of } i^{th} \text{ quarter in the } k^{th} \text{ county}}{\text{Respective Mid-year Population}}$$

Specifically, we have

$$\begin{aligned} C(1, 1, 1) &= \frac{\text{Total covid infections of Atlantic County in 1st week of April}}{\text{mid year population of Atlantic county in 2020}} \\ &= \frac{A(1, 1, 1)}{P(1, 1, 1)} = \frac{67}{274534} = 2.44e - 04 \end{aligned}$$

The tensor $\mathcal{C}$ is rescaled by dividing by the population of respective counties of mid-year 2020 [1]. The new tensor data $\bar{\mathcal{C}}$ has greatly improved efficiency in the numerical implementations. See sections ,, and for some experimental results.

3.2|Numerical Experiments

3.2.1|CP vs HOOI

One main advantage of tensor decomposition, namely, ALS and HOOI, is that it provides analytic tools for higher-order data in several modes. We implemented the algorithms, CP and HOOI, on our tensor $\bar{\mathcal{C}}$ of size $13 \times 7 \times 6$. First, we ran the ALS algorithm to construct the CP decomposition into three-factor matrices. Then we analyze through the visualization of the three-factor matrices from the estimated constructed tensor. We are able to estimate the same evolutionary patterns of Covid-19 cases as the original data tensor; see Figures 4 and 5. Collectively, we plot the cases in Figures 6 based on the intensity levels. HOOI is faster in time, but it gives lower accuracy results than CP.

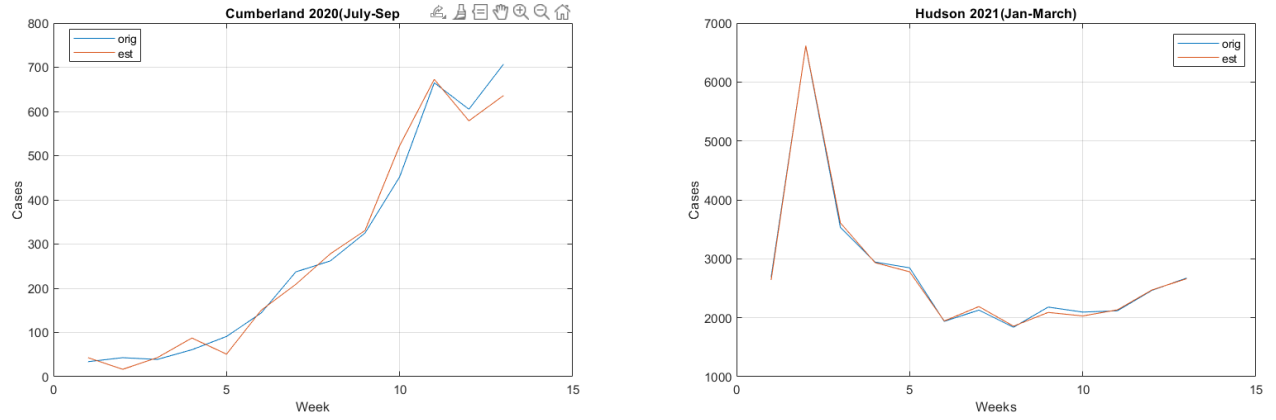


FIGURE 4. Original and Approximated using CP

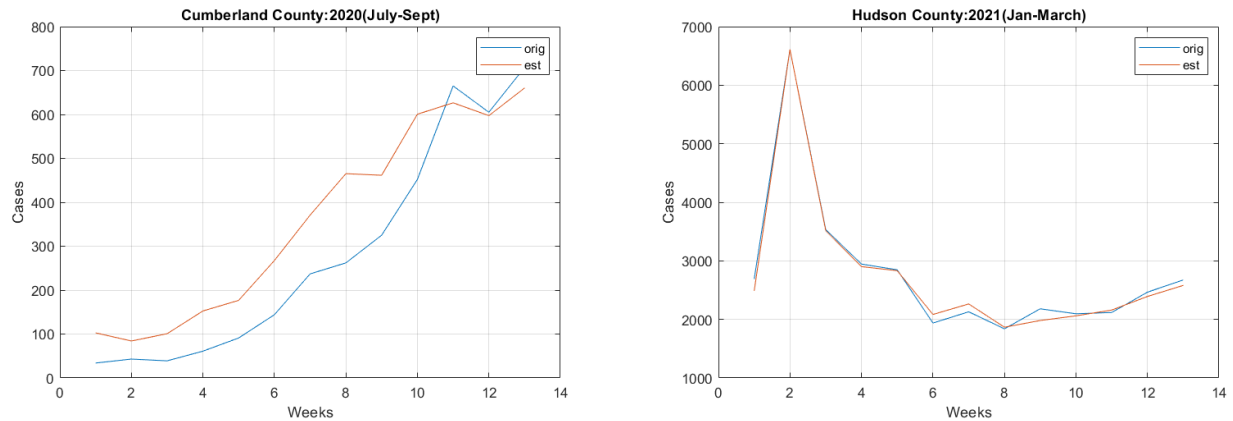


FIGURE 5. Original and Approximated using HOOI

Geo-Plot of CP approximation

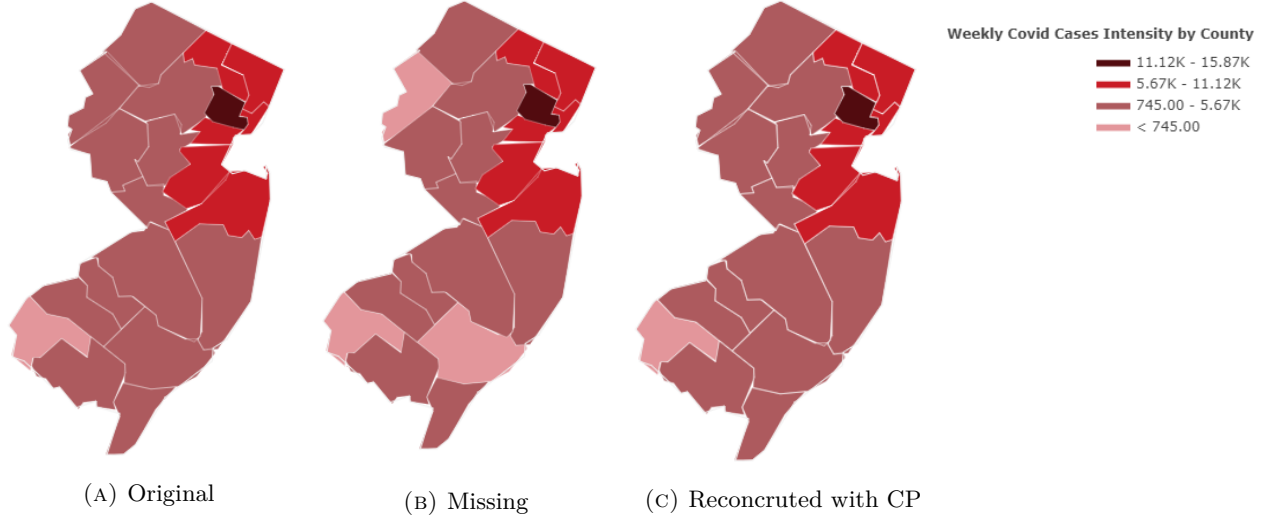


FIGURE 6. Geographical distribution plot of Covid-19 cases;

3.2.2|Extracting Patterns in Covid-19 via Factor Matrices

Extracting Patterns in Covid-19 via Factor Matrices. We further explore from our output factor matrices of the CP decomposition. We multiply different factor matrices to observe the county-wise, week-wise, and quarter-wise pattern of the Covid-19 cases. The first visualization of Figure 7 shows which quarter has the highest cases increment in 2nd week. Then the second visualization of the quarter, Sep 27-Dec-26 2020, indicates that the fourth week has the highest number of cases. Our algorithm estimates that the week, of Oct 4-10, 2020, is the week of the highest increment. Furthermore, we found that the same identical patterns on the original Covid-19 data tensor are consistent with our findings. Similarly, Figure 8 shows the identical pattern of our estimated and original Covid-19 infections in Essex county from June 27 - Sept 25, 2021.

The Cross Analysis of the Covid Cases with Visualization

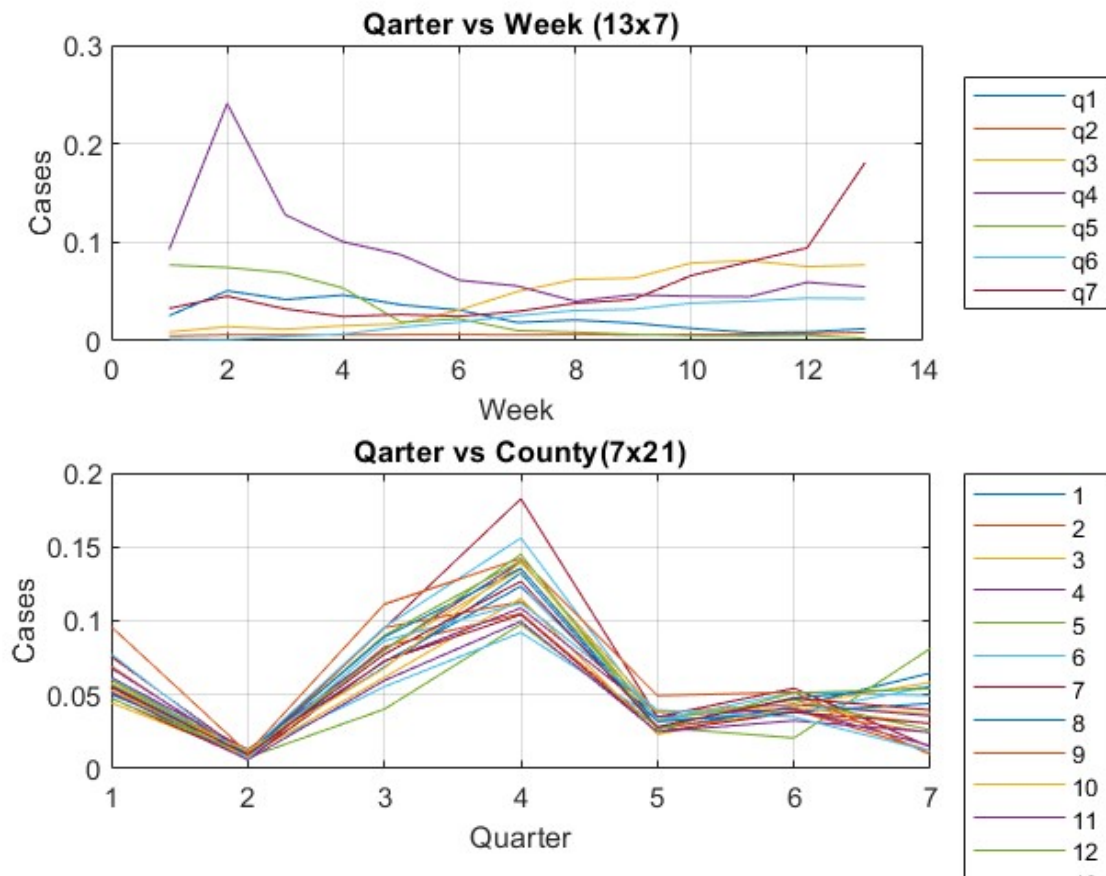


FIGURE 7. Cross-Pattern on Sep 27-Dec-26 2020

Estimated and Actual consecutive increment of relative cases

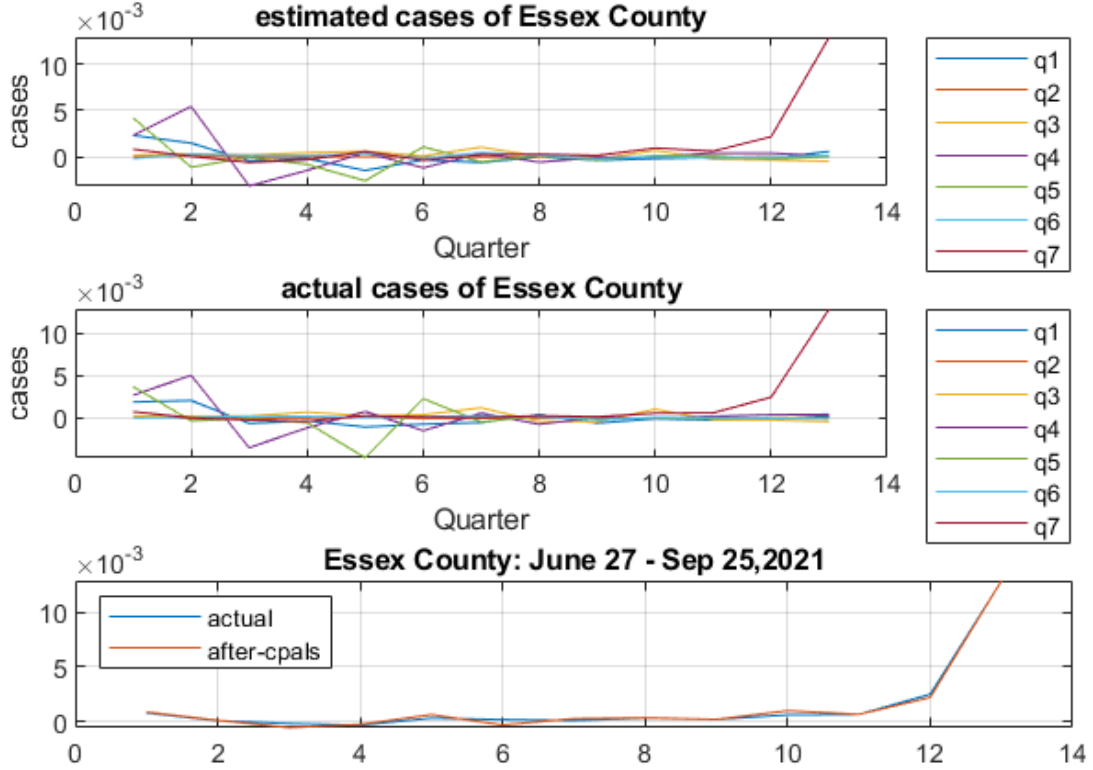


FIGURE 8. Covid-19 weekly consecutive increment (relative cases) of Essex county on 7th quarter of June 27 - Sept 25, 2021.

4|Sampling Method for Alternating Least-Squares (SMALS)

Given a tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ and a fixed positive integer R , the ALS algorithm solves three independent least squares problems alternatively. The least squares problems can be solved via various methods such as QR factorization, Cholesky factorization and etc. However, it requires of an $\mathcal{O}(R^3)$ floating point operations per second. To reduce this we let $S \subset \{1, \dots, R\}$ be the set of sample indices and A_s, B_s and C_s represent the sub-matrices obtained by choosing the columns of A, B , and C according to the index set S respectively. The partial derivatives of the objective function f with respect to the blocks A_s, B_s and C_s are

$$\frac{\partial f}{\partial A_s} = -X_{(1)}(C_s \odot B_s) + A(C \odot B)^T(C_s \odot B_s), \quad (10)$$

$$\frac{\partial f}{\partial B_s} = -X_{(2)}(C_s \odot A_s) + B(C \odot A)^T(C_s \odot A_s), \quad (11)$$

and

$$\frac{\partial f}{\partial C_s} = -X_{(3)}(B_s \odot A_s) + C(B \odot A)^T(B_s \odot A_s). \quad (12)$$

the stationary points of the above equations can be obtained by setting each gradient equal to zero. For instance $\nabla_{A_S} f = 0$ implies the following normal equation

$$A_S \left((C \odot B)^T (C_S \odot B_S) \right)^S = -A_{S^C} \left((C \odot B)^T (C_S \odot B_S) \right)^{S^C} + X_{(1)}(C_S \odot B_S) \quad (13)$$

where A^S represents the sub matrix of A obtained by sampling the rows of A corresponding to the sampling set S . Similar results can be derived by solving the equations $\nabla_{B_S} f = 0$ and $\nabla_{C_S} f = 0$. This reduces the latter computational complexity to $\mathcal{O}(\max\{|S|^3\})$. When I, J and K are relatively large, the reduction could be significant. In each iteration, the sampling set S is selected based on the performance of every block variable. For example, if updating the block A_j leads to a smaller decrease in the objective function than updating the block A_i , the index j will replace i in the subsequent iteration. More information on the differentiation of ALS can be found in [23].

Algorithm 4 Psuedo-code for SMALS

Initialize $\omega^0 = (A^0, B^0, C^0)$ where $A^0 \in \mathbb{R}^{I \times R}$, $B^0 \in \mathbb{R}^{J \times R}$ and $C^0 \in \mathbb{R}^{K \times R}$

General Step select $S \subset \{1, \dots, R\}$

For $k = 0, 1, \dots$ max iteraton, updates the corresponding blocks A_S, B_S and C_S :

$$A_S^{k+1} \in \arg \min_{A_S \in \mathbb{R}^{I \times |S|}} \frac{1}{2} \|X_{(1)} - (A_S, A_{S^C}^k)(C^k \odot B^k)^T\|_F^2$$

$$B_S^{k+1} \in \arg \min_{B_S \in \mathbb{R}^{J \times |S|}} \frac{1}{2} \|X_{(2)} - (B_S, B_{S^C}^k)(C^k \odot A^{k+1})^T\|_F^2$$

$$C_S^{k+1} \in \arg \min_{C_S \in \mathbb{R}^{K \times |S|}} \frac{1}{2} \|X_{(3)} - (C_S, C_{S^C}^k)(B^{k+1} \odot A^{k+1})^T\|_F^2$$

Update A^{k+1}, B^{k+1} and C^{k+1} by replacing the columns A_S^{k+1}, B_S^{k+1} and C_S^{k+1} .

Set $\omega^{k+1} = (A^{k+1}, B^{k+1}, C^{k+1})$.

Update the sampling set S^k according to the performance of each block variable in the previous iteration

Repeat until the stopping criteria are met.

4.1|Computational Complexity: ALS vs SMALS

In this section, we describe the computational complexity of SMALS. For a given rank R , the ALS algorithm requires computing a pseudo-inverse matrix $R \times R$ in the inner iteration, which incurs a high cost for the large tensors. In contrast, the SMALS algorithm ((4)) requires computing a pseudo-inverse of $|S| \times |S|$ matrix where $S \subset \{1, \dots, R\}$. Numerically, the computational complexity of the ALS algorithm [33] is $3RIJK + (7R^2 + R)(JK + KI + IJ) + (I + J + K)(R^2 + R) + 11R^3 \approx \mathcal{O}\{(R^2(IJ + JK + KI) + RIJK)\}$. It follows that SMALS has a computational complexity $\mathcal{O}\{|S|^2(IJ + JK + KI) + |S|IJK\}$ where $|S| \leq R$.

4.2|Numerical Results: SMALS vs ALS

These experiments ran on a laptop with Apple M1 chip, 8-core CPU, 8GB memory, and 512GB storage. In order to demonstrate the effectiveness of the SMALS algorithm and to compare it with the standard ALS algorithm, we have conducted numerical experiments using both Covid-19 tensor data, random color image data, and randomly generated data as well. Specifically, we have focused on the comparison of lower rank approximations over the Covid-19 case tensor of three counties in New Jersey. Our results show that the SMALS estimation approach performs well in every case see figure 9.

Figure 10 illustrates the performance of the SMALS algorithm on image data and demonstrates its compelling results. Additionally, the error evolution in figure 16 indicates that SMALS performs better than ALS, as it has fewer fluctuations or "swamps". The experimental data presented in table 2 further supports this conclusion, as it shows that SMALS executes with less time cost than ALS for each tensor of different sizes and various ranks.

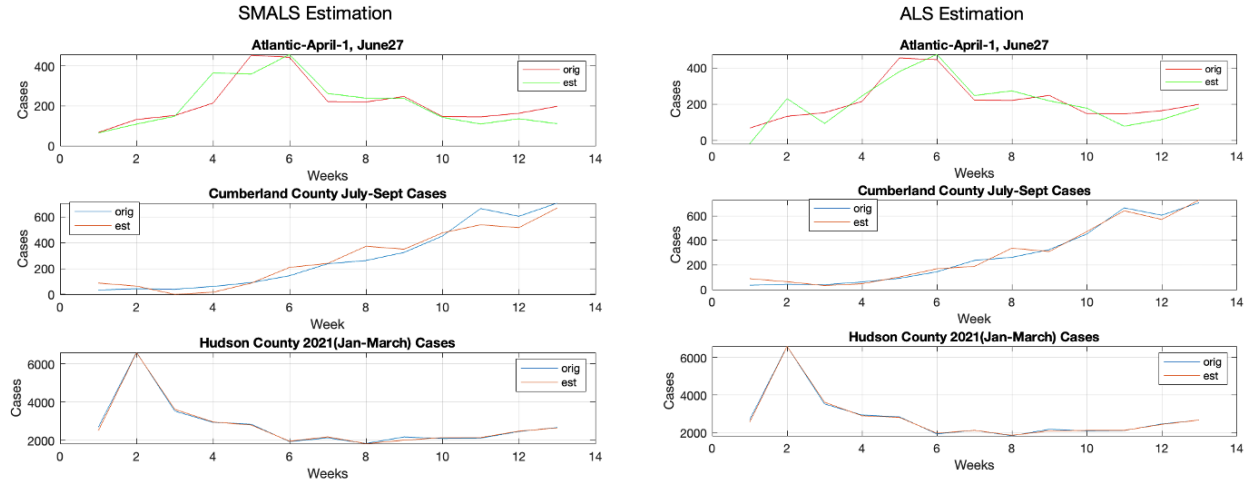


FIGURE 9. SMALS vs ALS on Covid-19 Tensor



FIGURE 10. SMALS vs ALS on Random Color Images ($R = 20$)

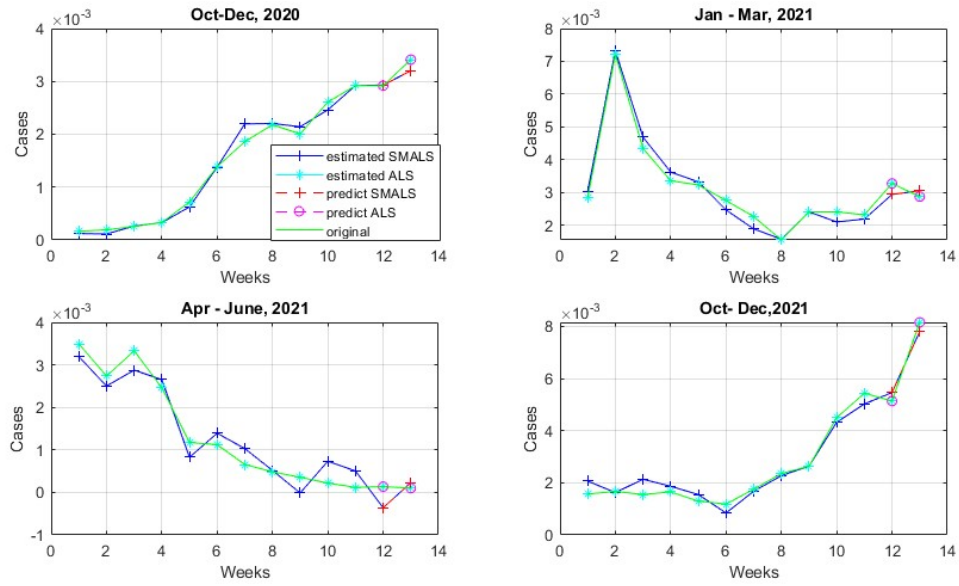


FIGURE 11. Warren County: 4th Quarter(Jan- March 2021), 7th(Oct-Dec,2021), 1st(April-June 2020) last week covid cases prediction using SMALS algorithm with rank 9×7 about 20% data missing

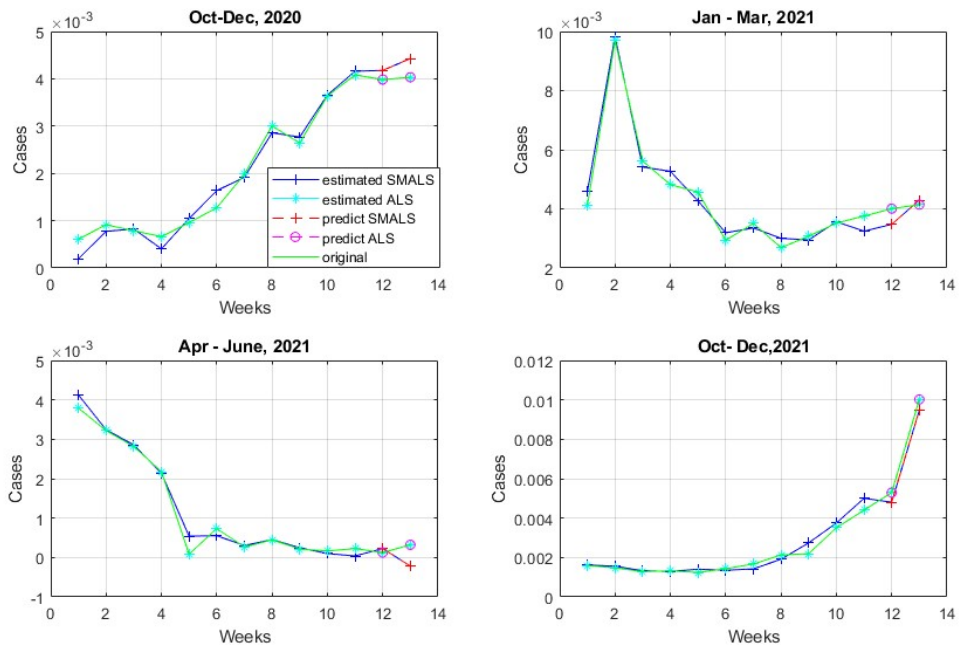


FIGURE 12. Middlesex County: 4th Quarter(Jan- March 2021), 7th(Oct-Dec,2021), 1st(April-June 2020) last week covid cases prediction using SMALS algorithm with rank 9×7 about 30% data missing

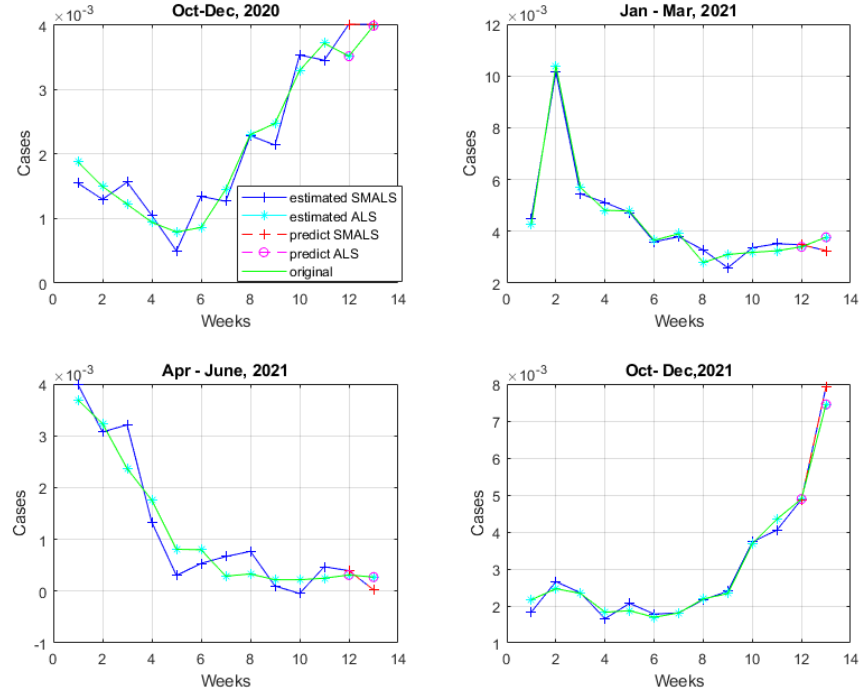


FIGURE 13. Ocean County: Fourth Quarter(Jan- March 2021), 7th(Oct-Dec,2021), 1st(April-June 2020) last week covid cases prediction using SMALS algorithm with rank 9×7 about 70% data missing

Rank	ALS		SMALS	
	Error	Time (s)	Error	Time (s)
9×7	2.98e-4	4.06	2.65e-02	2.65
7×7	5.4e-03	2.95	2.93e-02	2.1
7×8	2.1e-04	3.29	2.5e-02	2.51

TABLE 1. Efficiency Comparison: ALS vs SMALS

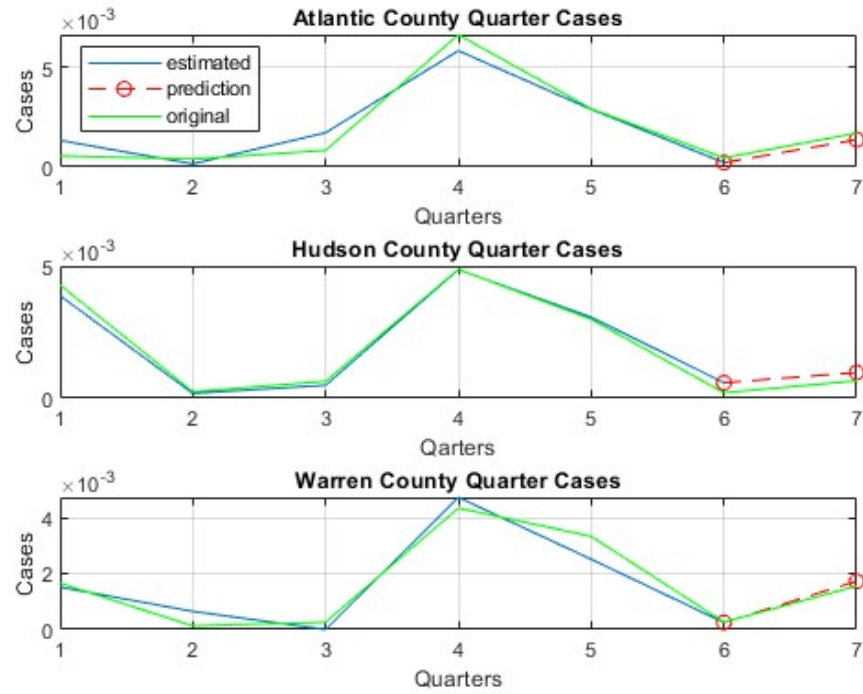


FIGURE 14. Quarter Estimation using SMALS

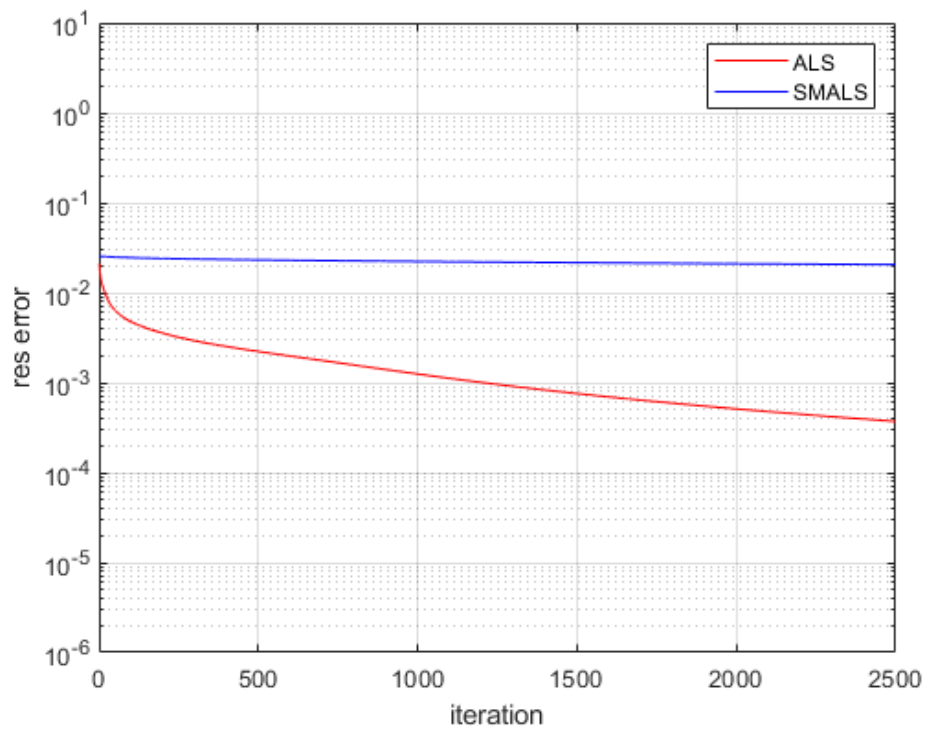


FIGURE 15. Residual Error: SMALS vs ALS with rank 63

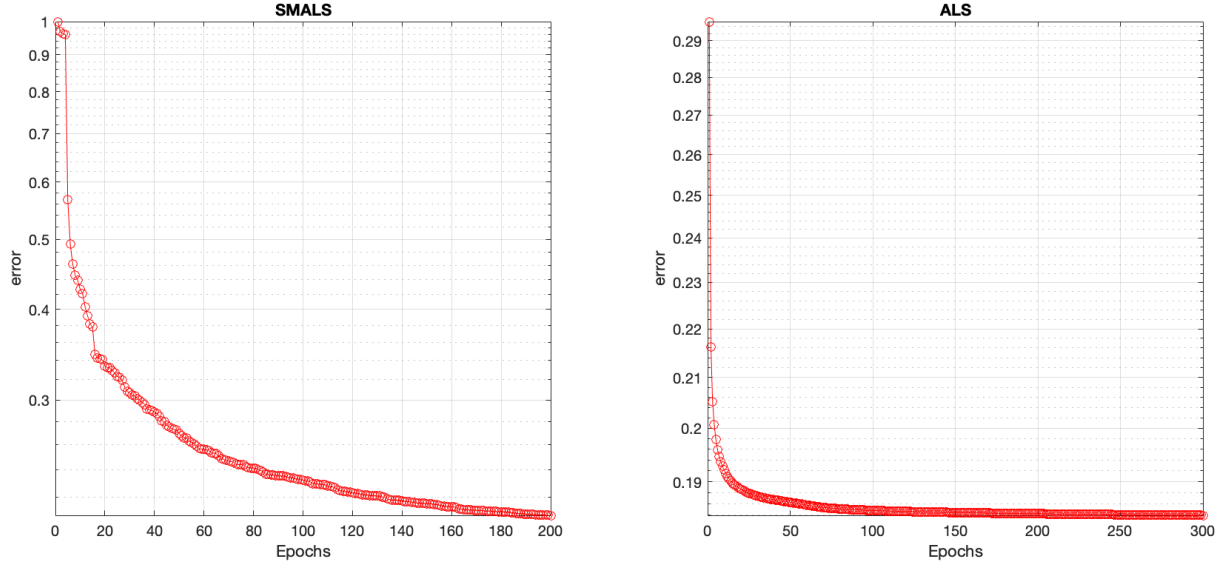


FIGURE 16. Error Evolution of ALS vs SMALS on random images

Efficacy Comparison Between ALS and SMALS

ALS		SMALS		Rank	Tensor Size
Error	time	Error	Time		
5.84e-03	0.79s	2.75e-02	0.24s	50	13X7X21
3.6e-02	0.45s	5.70e-02	0.14s	20	13X7X21
7.10e-02	0.28s	9.70e-02	0.10s	10	13X7X21
5.05e-04	1.16s	1.92e-03	0.39s	75	13X7X21
5.90e-02	1.63s	7.38e-02	1.42s	20	224X224X3
4.10e-02	2.82s	6.62e-02	2.47s	37	224X224X3
3.1e-02	3.42s	4.97e-02	2.78s	60	91X8X21
2.30e-01	60.4s	2.4e-01	54.2s	20	1433X1068X3
1.98e-01	35.1m	2.67e-01	25.1m	40	3000X4000X3
0.98	4.2m	0.98	3.9m	100	180X240X75

TABLE 2. ALS vs SMALS Comparison

We implement both ALS and SMALS algorithms on different sizes of varying tensors. We run the codes twenty times for each tensor case and average their results. Our result indicates SMALS is more efficient in terms of time cost, see table 2

5|Tensor Sparse Optimization

In [29], an iterative method based on proximal algorithms called low-rank approximation of tensors (LRAT) was proposed to solve the minimum rank optimization:

$$\min_{a_r, b_r, c_r, \alpha_r} \|\mathcal{C} - \mathcal{L}\|_F^2 + \sigma \|\alpha\|_{\ell_1} \quad (14)$$

where $\mathcal{L} = \sum_{r=1}^R \alpha_r a_r \circ b_r \circ c_r$, $\mathcal{C}$, a given data tensor. The implementation is in Algorithm 5. In a recent work, [7], a more practical choice of the parameter σ led to a more efficient and accurate algorithm. The practical regularization method is based on a flexible Golub-Kahan (fGK) process. In Figures 17 and 18, we implemented the LRAT + fGK algorithm with the convergence plot; the sparse model predicts the general phenomena of the Covid-19 cases.

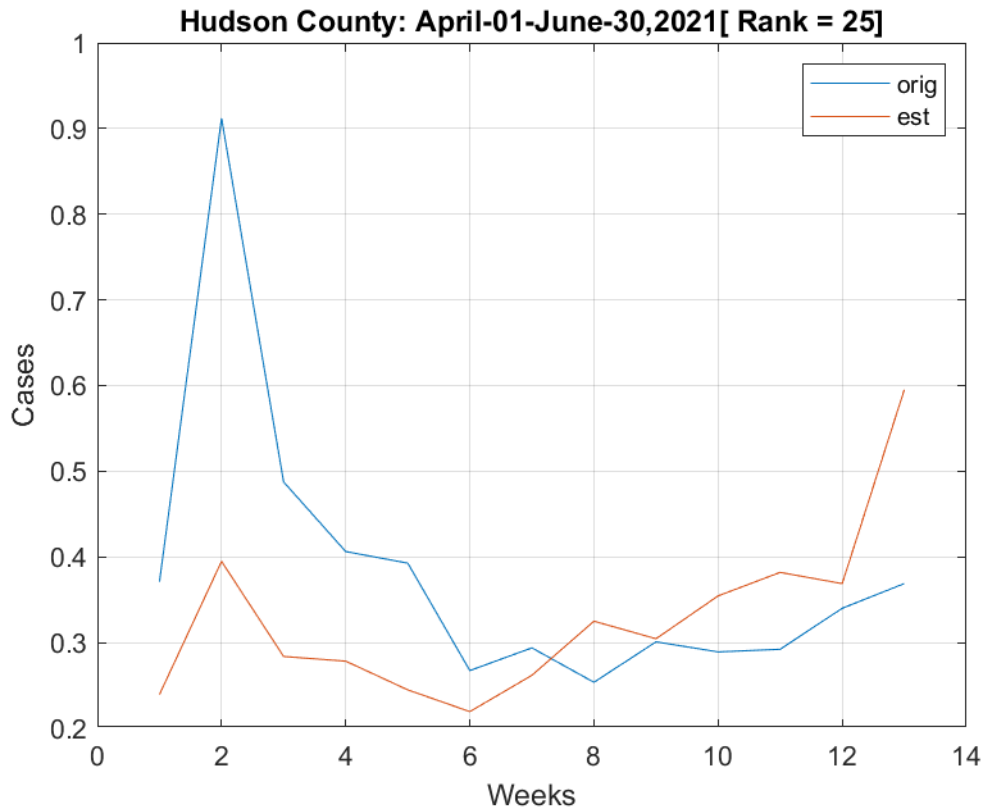


FIGURE 17. LRAT + fGK Estimation

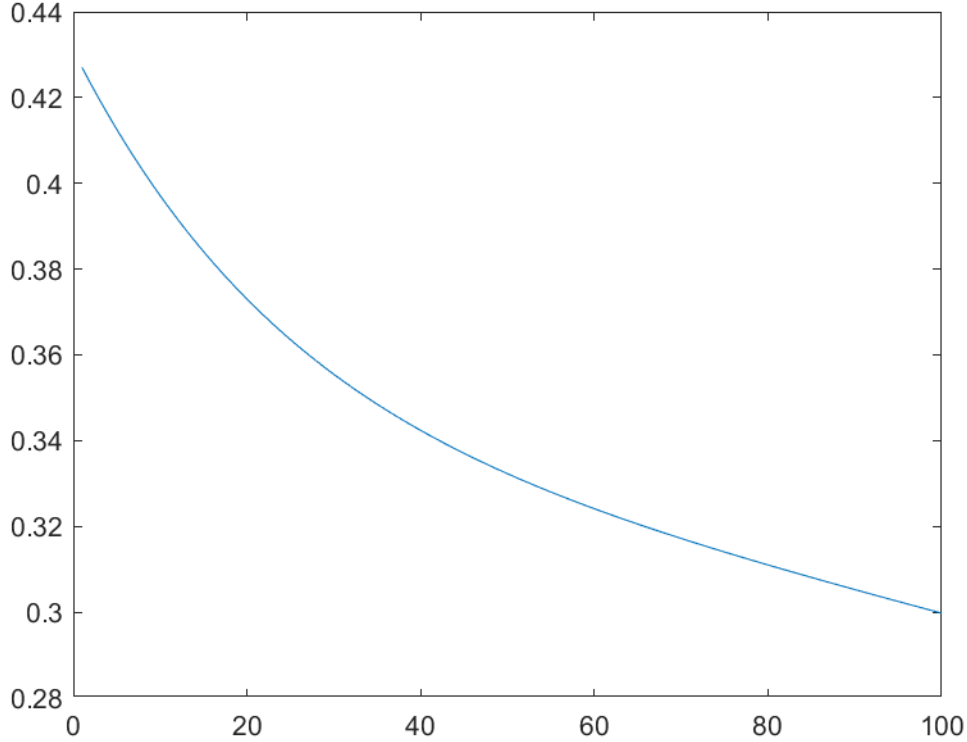


FIGURE 18. LRAT + fGK Convergence Plot

In figure 7, the error plot shows the difference between the original data and estimation. The horizontal line detects an anomaly and this spot coincides with a spike in the number of infections.

5.1|Hotspot Identification

[12, 2] Recently, “hotspots” in infectious disease epidemiology have been increasingly used, and they dictate the implementation of appropriate control measures for the specific place. Despite “Hotspots” has not a concrete definition, it is described variously as per area of elevated incidence, prevalence, higher transmission efficiency, or higher chance of disease emergence. Our research is also on Covid-19 pandemic-infected population data, so we defined “hotspots” as the geographical area where the higher intensity of disease prevalence and transmission rate as per population density and flow in the specific area. More specifically, we have defined a threshold in the specific area as per population and its activities, indicating the hotspots there. Identifying hotspots attracts the attention of authorities so that more efficient control measures may be implemented by targeting these areas to sustain further transmission.

5.1.1|Sparse Optimization for Hotspot Identification

Our goal is to detect hotspots rapidly. Our goal is to have the following decomposition: $\mathcal{Y} = \mathcal{L} + \mathcal{S}$ where $\mathcal{Y}$ is the given tensor, $\mathcal{L}$ is a low rank reconstructed tensor of $\mathcal{Y}$ and $\mathcal{S}$ is the sparse tensor. In video processing, the original video is separated into background and foreground subspaces to detect anomalous activities. The tensor $\mathcal{L}$ is the background, and $\mathcal{S}$ is the foreground. The sparse tensor $\mathcal{S}$ can provide anomalous activities. Similarly, $\mathcal{S}$ will contain hotspot occurrence. Thus, we use the sparse tensor model and implement LRAT 5.

5.1.2|Hotspot Detection with Practical Threshold

First, we convert the Covid-19 data tensor into a tensor with each entry as the rate of change in the number of infections from the prior week by computing

$$\frac{\text{Number of infections this week} - \text{Last week's total infections}}{\text{Last week's number of infections}}$$

for each entry. Then we apply the LRAT algorithm to the new tensor. We use the practical threshold $\text{mean} + 5 \times \text{standard deviation}$ on the newly converted tensor for Atlantic County during April 01-June 30, 2020 see, Figure 19 and $\text{mean} + 1.5 \times \text{std}$ for Burlington county during October 01- Dec 25, 2020, See Figure 7 The resulting plot from LRAT spikes above the threshold line in the week considered to have hotspots. The hotspots detected through our algorithm with a suitable threshold match the pattern of the original Covid-19 infection data.

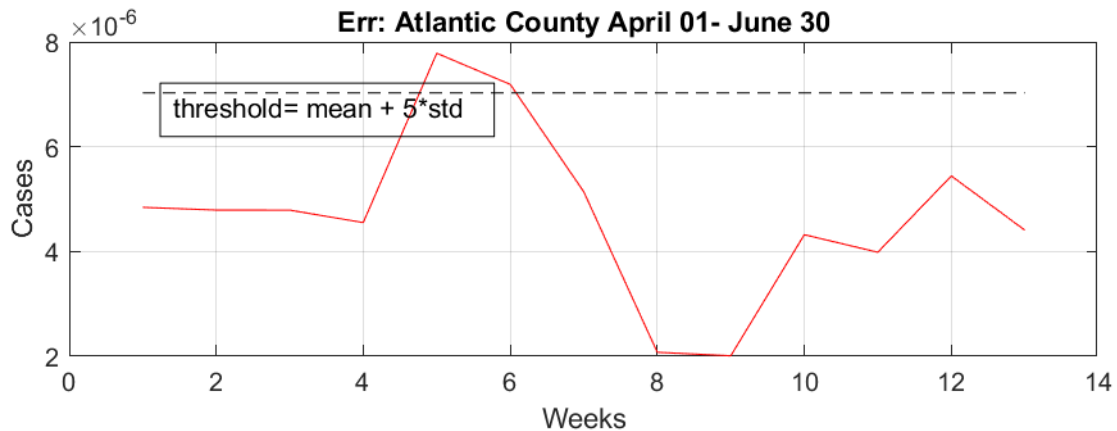


FIGURE 19. April 01 - June 30, 2020

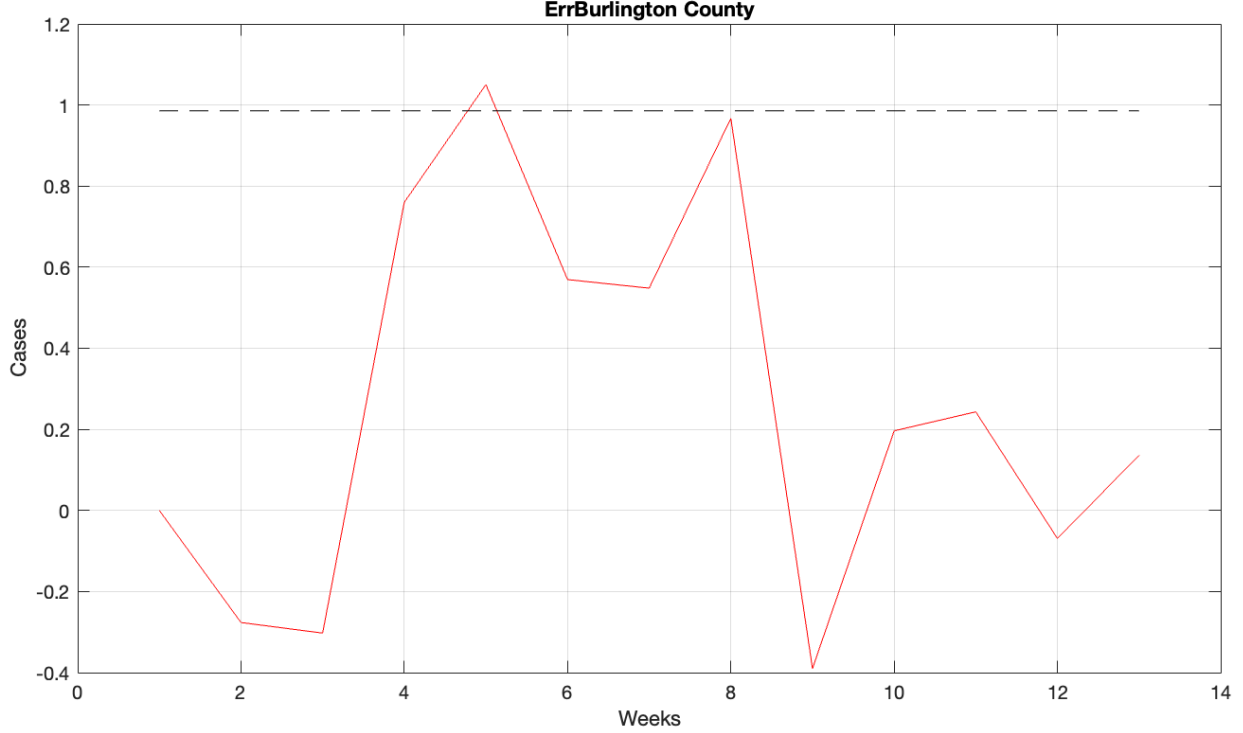


FIGURE 20. October 01- December 25,2020

5.2|Tensor Completion for Predicting Covid-19 Infection Cases

The tensor completion is the problem of completing the missing or unobserved entries of the partially observed tensor. Tensor completion algorithms have a wide range of applications in the field of big data [24], computer vision such as image completion [7, 28] which focus on filling the missing entries in a presence of noise. Other important applications of tensor completion are link prediction [15] and recommendation system [8, 5] and video completion [14]. With the given the tensor $\mathcal{L}$ of order n with missing entries for a given rank, the tensor completion optimization problem can be formulated as the following:

$$\begin{aligned} & \text{minimize}_{\mathcal{L}} \text{rank}(\mathcal{L}) \\ & \text{subject to } \mathcal{L}(\Omega) = \mathcal{C}(\Omega) \end{aligned}$$

In the work of Wang and Navasca [29], this optimization is reformulated to the tensor sparse model. To apply the tensor sparse model with constraints in the prediction of Covid-19 infection cases, we set up our data by removing some column data from the original tensor; see Figure 21. We implement algorithm 5 to complete the Covid-19 tensor with the observed data constraints. The reconstructed tensor exhibits the same pattern as the tensor cases even though there are some dissimilarities in particular numerical data.

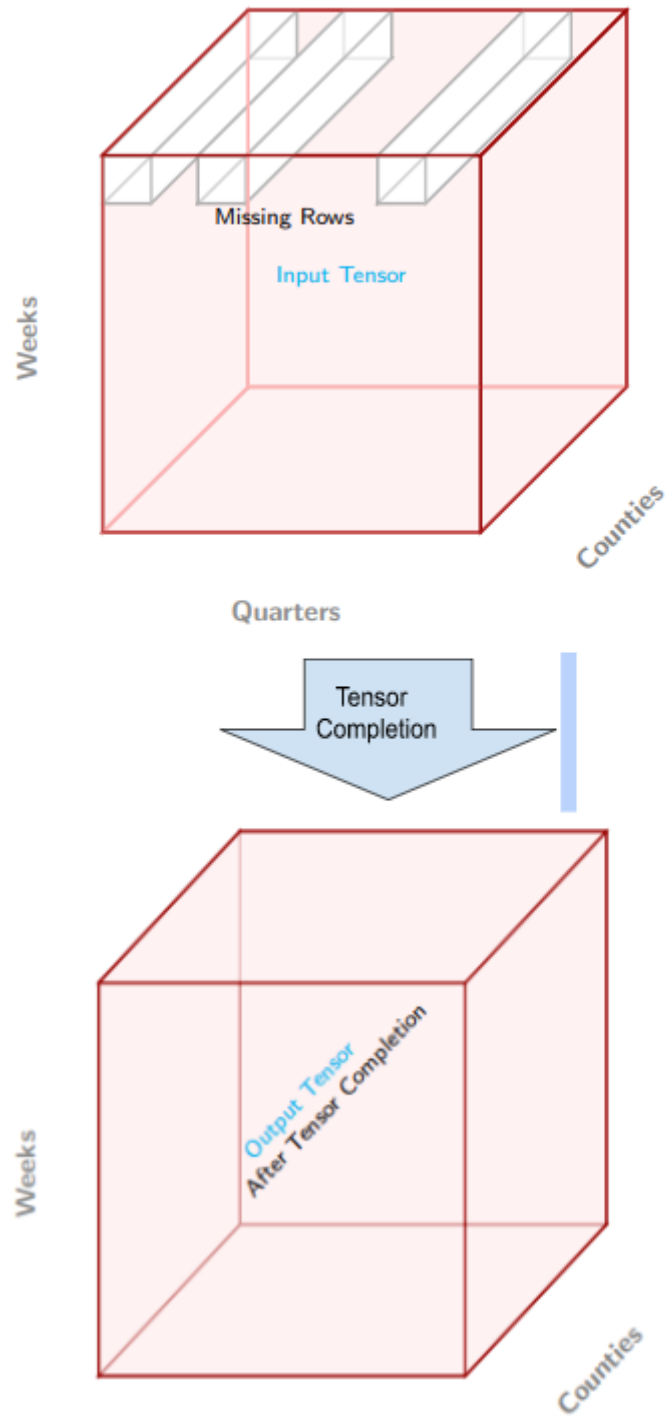


FIGURE 21. Our Tensor Completion Architecture on Covid-19 data Tensor

Algorithm 5 [29, 7, 21, 34]Tensor Completion via LRAT with ISTA or FGK

Require: Tensor $\mathcal{C} \in \mathbb{R}^{I \times J \times K}$ rank of the tensor R , a regularization parameter λ and a scale $t > 0$.

Ensure: An approximated tensor $\mathcal{X}$

1: Initialize a tensor $\mathcal{C}^0 = [\sigma^0; A^0, B^0, C^0]_T$.

2: Update steps:

I) Update factor matrices A, B, C :

Compute $U^n = D^n(C \odot B)^T$ and let $d_n = \max\{\|U^n U^{nT}\|_F, 1\}$.

Compute D^n and A^{n+1} by

$$D^n = A^n - \frac{1}{td_n} \nabla_A f(A^n, B^n, C^n, \sigma^n),$$

$$A^{n+1} = D^n \text{diag}(\|d_1^n\|, \dots, \|d_T^n\|)^{-1}$$

where d_i^n is the i -th column of D^n for $i = 1, \dots, T$ and

$\nabla_A f$ is gradient of the cost function given by 14

Compute V^n as U^n and let $e_n = \max\{\|V^n V^{nT}\|_F, 1\}$.

Compute E^n and B^{n+1} by

$$E^n = B^n - \frac{1}{te_n} \nabla_B f(A^{n+1}, B^n, C^n, \sigma^n),$$

$$B^{n+1} = E^n \text{diag}(\|e_1^n\|, \dots, \|e_T^n\|)^{-1}$$

where e_i^n is the i -th column of E^n for $i = 1, \dots, T$.

Compute W^n as U^n and let $f_n = \max\{\|W^n W^{nT}\|_F, 1\}$.

Compute F^n and C^{n+1} by

$$F^n = C^n - \frac{1}{tf_n} \nabla_C f(A^{n+1}, B^{n+1}, C^n, \sigma^n),$$

$$C^{n+1} = F^n \text{diag}(\|f_1^n\|, \dots, \|f_T^n\|)^{-1}$$

where f_i^n is the i -th column of F^n for $i = 1, \dots, R$.

II) Update the row vector σ :

Compute Q^{n+1} where $Q = (q_1^T, \dots, q_R^T)$, q_r is row vector

of rank one tensor $a_r \circ b_r \circ c_r$ constructed from columns of respective factor matrices A, B and C ,

let $\eta_n = \max\{\|Q^{n+1} Q^{n+1T}\|_F, 1\}$.

via ISTA: Compute σ^{n+1} by

$$\sigma^{n+1} = \arg\min_{\sigma} \left\{ \frac{1}{2} \|\sigma - \sigma^n + \frac{1}{s} \nabla_{\sigma} f(A^{n+1}, B^{n+1}, C^{n+1}, \sigma^n)\|^2 + \frac{1}{s\eta_n} \|\sigma\|_1 \right\}$$

via FGK:

Compute σ^{n+1} by

$$\sigma^{n+1} = \arg\min_{\sigma} \left\{ \frac{1}{2} \|t - \sigma Q\|_F^2 + \lambda \|\sigma\|_1 \right\}$$

5.2.1|Numerical Experiments on Prediction of Infected Cases

We implement a tensor completion algorithm to our tensor data. We replace the last (most current) week's data of the Atlantic and Warren counties with the mean of the remaining data. Then, the missing values on the tensor are completed via a low-rank approximation with Flexible Golub Kahan (FGK) and ISTA as well. Figure (6a) depicts the geographical distribution of Covid-19 cases by severity in the final week of December 2021 in the state of New Jersey. Figure (6b) illustrates the same as figure (6a), but with the omission of cases from two counties, while Figure (6c) shows the final representation after the application of the tensor completion algorithm LRAT with FGK. Figure 22 and 23 show the prediction of last week's covid infections in Atlantic and Warren County of a particular quarter using LRAT+ FGK and LRAT + ISTA, respectively, whereas Figure 24 and 25 show the prediction of pattern and last quarter total covid infections. Figure 26 compares the error evolution of two algorithms.

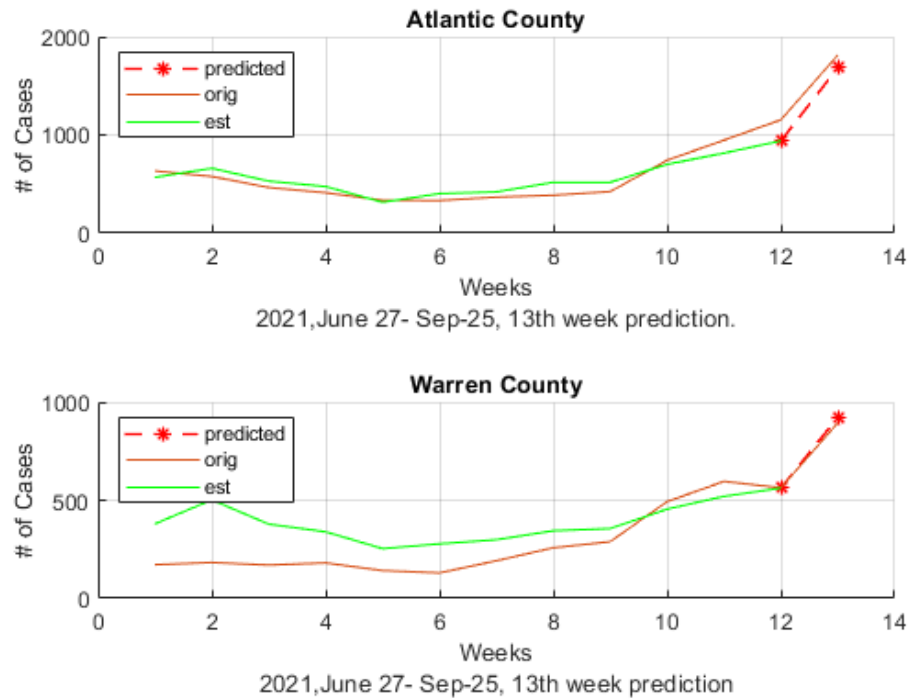


FIGURE 22. LRAT + FGK

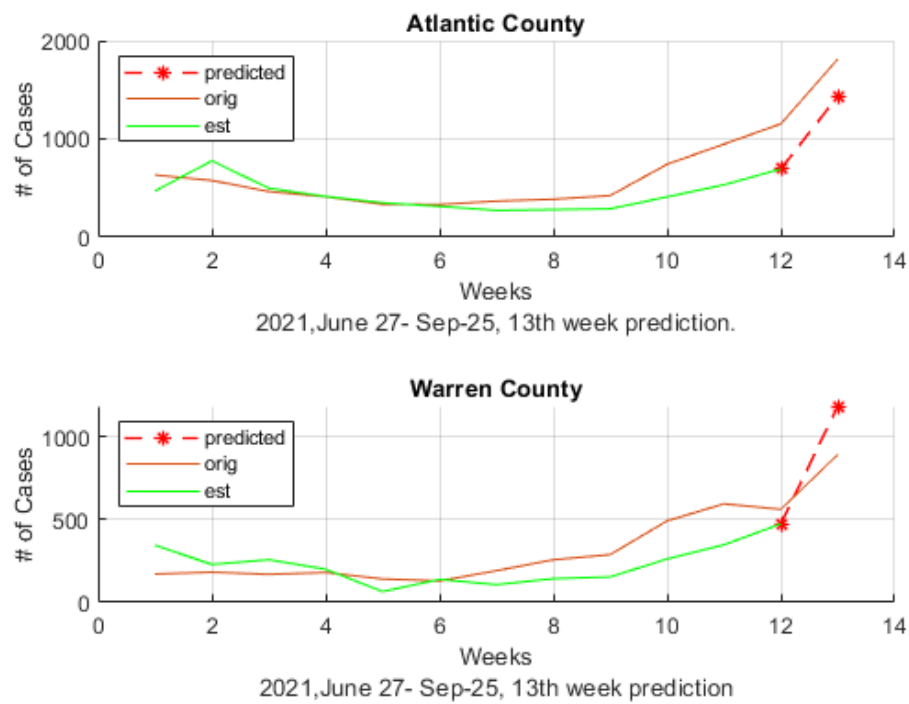


FIGURE 23. LRAT + ISTA

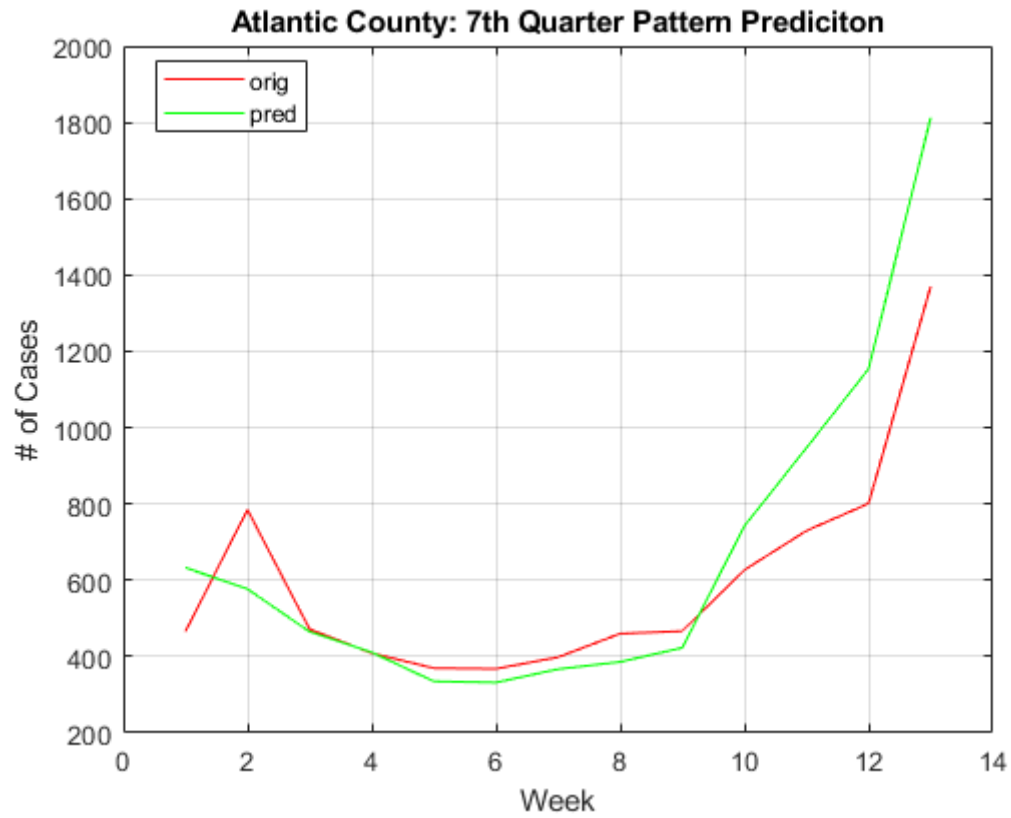


FIGURE 24. Prediction of Covid-19 Pattern of 7th quarter of Atlantic county via LRAT with FGK.

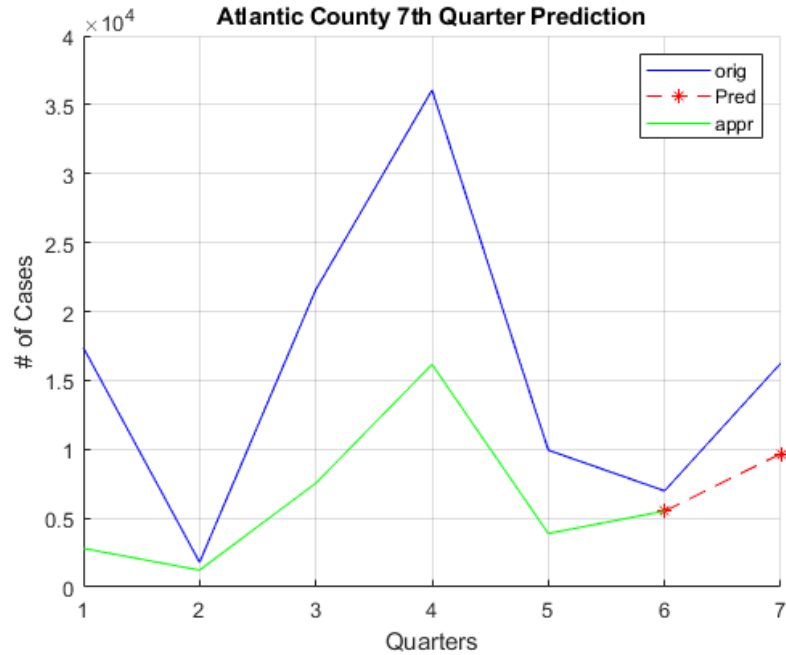


FIGURE 25. 7th Qtr Prediction of Covid Infections: this is the total number of cases prediction of Atlantic county for 7th Quarter (June27-Sep25)

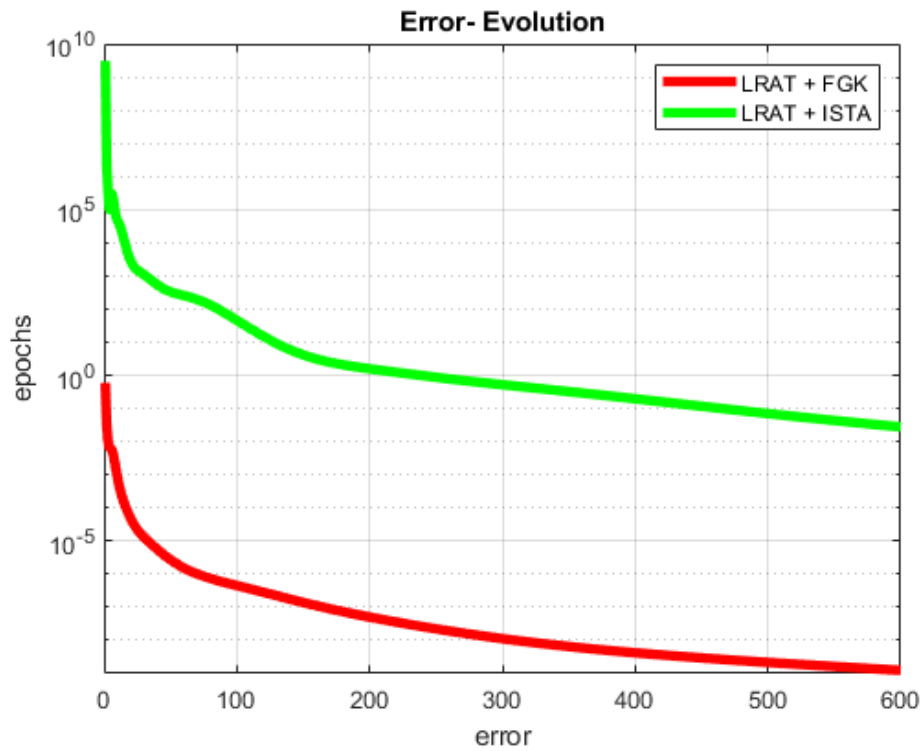


FIGURE 26. Comparison over: FGK and ISTA

5|Conclusion

In this work, we apply various tensor models and tensor algorithms to analyze Covid-19 data. The standard tensor models, CP and HOOI, with their off-the-shelves algorithms, ALS and HOOI, are tested against a new sampling method for ALS (SMALS). The numerical results are auspicious as it cuts the downtime while keeping the Frobenius norm errors relatively consistent with ALS. Here tensor sparse model [29, 7] is used as the model for predicting future Covid-19 infection cases as a tensor completion problem. The numerical results are impressive as the tensor completion algorithm can predict infection a week and quarter ahead. Moreover, the sparse tensor model can locate which counties exhibit hotspots. The sparse tensor model is based on proximal algorithms and the flexible hybrid method by Golub-Kahan for efficient practical implementation.

In our future work, we would like a more mathematical and methodical technique for locating and detecting hotspots. We have used l_1 minimization in the tensor completion; we will work on the efficacy of l_0 minimization of low-rank approximation of CP decomposition and tensor completion algorithm in the proximal framework.

Acknowledgments

The authors extend their heartfelt appreciation to the editors and anonymous reviewers for their invaluable feedback and constructive criticism, which played a vital role in significantly enhancing the quality of this paper.

Author Contribution

All authors have equal contributions to this project. All authors have read and agreed to the published version of the manuscript.

Funding

This material is based upon work supported by the National Science Foundation under Grant No. DMS-1439786 while the author, C. Navasca, was in residence at the Institute for Computational and Experimental Research in Mathematics in Providence, RI, during the Model and Dimension Reduction in Uncertain and Dynamic Systems Program. C. Navasca is also in part supported by National Science Foundation No. MCB-2126374.

Conflicts of Interest

The authors declare that they have no conflict of interest regarding the research findings presented. The funders did not participate in the study's design, data collection, analysis, interpretation, manuscript writing, or decision to publish the results.

References

- [1] United States Census Bureau: New Jersey Population (2020) GitHub repository. <https://www.census.gov/en.html>
- [2] Bejon, P. and Williams, T. N. and Liljander, A. and Noor, A. M. and Wambua, J. and Ogada, E. and Olotu, A. and Osier, F. H. and Hay, S. I. and Färnert, A. and Marsh, K. (2010) PLoS Med. Stable and unstable malaria hotspots in longitudinal cohort studies in Kenya <https://www.census.gov/en.htm>
- [3] Rasmus Bro, PARAFAC. Tutorial and applications Chemometrics and Intelligent Laboratory Systems, 1997 [https://doi.org/10.1016/S0169-7439\(97\)00032-4](https://doi.org/10.1016/S0169-7439(97)00032-4)
- [4] Fengyu Cong and Qiu-Hua Lin and Li-Dan Kuang and Xiao-Feng Gong and Piia Astikainen and Tapani Ristaniemi, Tensor decomposition of EEG signals: A brief review, Journal of Neuroscience Methods, 2015, <https://doi.org/10.1016/j.jneumeth.2015.03.018>
- [5] Ge, Hancheng and Caverlee, James and Lu, Haokai, TAPER: A Contextual Tensor-Based Approach for Personalized Expert Recommendation, Association for Computing Machinery, New York, NY, USA, Proceedings of the 10th ACM Conference on Recommender Systems, 2016,
- [6] Hillar, Christopher J. and Lim, Lek-Heng, Most Tensor Problems Are NP-Hard, Association for Computing Machinery, New York, NY, USA, J. ACM, 2013, <https://doi.org/10.1145/2512329>
- [7] Jiang, Jiahua and Sanogo, Fatoumata and Navasca, Carmeliza, Low-CP-rank tensor completion via practical regularization, Journal of Scientific Computing, 2022, <https://doi.org/10.1007/s10915-022-01789-9>

- [8] Karatzoglou, Alexandros and Amatriain, Xavier and Baltrunas, Linas and Oliver, Nuria, Multiverse Recommendation: N-Dimensional Tensor Factorization for Context-Aware Collaborative Filtering, Association for Computing Machinery, Proceedings of the Fourth ACM Conference on Recommender Systems, 2010
- [9] Karim, Ramin Goudarzi and Guo, Guimu and Yan, Da and Navasca, Carmeliza, Accurate tensor decomposition with simultaneous rank approximation for surveillance videos, 2020 54th Asilomar Conference on Signals, Systems, and Computers, 2020
- [10] Tamara G. Kolda and Brett W. Bader, Tensor Decompositions and Applications, SIAM Review, 2009 10.1137/07070111X
- [11] J.B. Kruskal, Three-way arrays: rank and uniqueness of trilinear decompositions, with applications to arithmetic complexity and statistics, SLinear Algebra Appl, 1977
- [12] Lessler, J. and Azman, A. S. and McKay, H. S. and Moore, S. M., What is a Hotspot Anyway?, Am J Trop Med Hygl, 2017
- [13] Li, Na and Kindermann, Stefan and Navasca, Carmelizar, Some Convergence Results on the Regularized Alternating Least-Squares Method for Tensor Decomposition, arXiv, 2011, <https://arxiv.org/abs/1109.3831>
- [14] Ji Liu and Musialski, Przemyslaw and Wonka, Peter and Jieping Ye, Tensor completion for estimating missing values in visual data, 2009 IEEE 12th International Conference on Computer Vision, 2009, 10.1109/ICCV.2009.5459463
- [15] Yuanyuan Liu and Fanhua Shang and Hong Cheng and James Cheng and Hanghang Tong, Factor Matrix Trace Norm Minimization for Low-Rank Tensor Completion, SDM, 2014,
- [16] Liu , Fen and Chen , Jianfeng and Tan , Weijie and Cai , Chang, A Multi-Modal Fusion Method Based on Higher-Order Orthogonal Iteration Decomposition, Entropy, 2021, <https://www.mdpi.com/1099-4300/23/10/1349>
- [17] Hanbaek Lyu and Christopher Strohmeier and Georg Menz and Deanna Needell, COVID-19 Time-series Prediction by Joint Dictionary Learning and Online NMF, CoRR, 2020, <https://arxiv.org/abs/2004.09112>
- [18] Mao, Xianpeng and Yuan, Gonglin and Yang, Yuning, A self-adaptive regularized alternating least squares method for tensor decomposition problems, Analysis and Applications, 2020, <https://doi.org/10.1142/S0219530519410057>
- [19] Mathew, Jimson and Behera, Ranjan Kumar and Panthakkalakath, Zenin Easa and others, A self-adaptive regularized alternating least squares method for tensor decomposition problems, 2020, arXivpreprintarXiv:2010.00382
- [20] Evangelos E. Papalexakis and Christos Faloutsos and Nicholas D. Sidiropoulos, Tensors for Data Mining and Data Fusion: Models, Applications, and Scalable Algorithms, ACM Trans. Intell. Syst. Technol., 2017, <https://doi.org/10.1145/2915921>
- [21] F. Sanogo and C. Navasca, Tensor Completion via the CP Decomposition, 2018 52nd Asilomar Conference on Signals, Systems, and Computers, 10.1109/ACSSC.2018.8645405
- [22] Bernard N. Sheehan and Yousef Saad, Higher Order Orthogonal Iteration of Tensors (HOOI) and its Relation to PCA and GLRAM, SDM, 2007
- [23] Sidiropoulos, Nicholas D. and De Lathauwer, Lieven and Fu, Xiao and Huang, Kejun and Papalexakis, Evangelos E. and Faloutsos, Christos, Tensor Decomposition for Signal Processing and Machine Learning, IEEE Transactions on Signal Processing, 2017 10.1109/TSP.2017.2690524
- [24] Song, Qingquan and Ge, Hancheng and Caverlee, James and Hu, Xia, Tensor Completion Algorithms in Big Data Analytics, Association for Computing Machinery, 2019 <https://doi.org/10.1145/3278607>
- [25] NY Times, New Jersey Counties Covid Cases, GitHub repository, 2020 <https://github.com/nytimes/covid-19-data/blob/master/us-counties.csv>
- [26] Uschmajew, André, Local Convergence of the Alternating Least Squares Algorithm For Canonical Tensor Approximation, SIAM Journal on Matrix Analysis and Applications, 2012 10.1137/110843587
- [27] Haiyan Wang and Nao Yamamoto, Using a partial differential equation with Google Mobility data to predict COVID-19 in Arizona, Mathematical Biosciences and Engineering, 2020, <https://www.aimspress.com/article/doi/10.3934/mbe.2020266>
- [28] Wang, X. and Navasca, C, Adaptive Low Rank Approximation of Tensors, Proceedings of the IEEE International Conference on Computer Vision Workshop (ICCVW), Santiago, Chile, 2015,
- [29] Wang, Xiaofei and Navasca, Carmeliza, Low-rank approximation of tensors via sparse optimization, Numerical Linear Algebra with Applications, 2018, <https://doi.org/10.1002/nla.2136>
- [30] Zifeng Yang and Zhiqi Zeng and Ke Wang and Sook-San Wong and Wenhua Liang and Mark Zanin and Peng Liu and Xudong Cao and Zhongqiang Gao and Zhitong Mai and Jingyi Liang and Xiaoqing Liu and Shiyue Li and Yimin Li and Feng Ye and Weijie Guan and Yifan Yang and Fei Li and Shengmei Luo and Yuqi Xie and Bin Liu and Zhoulang Wang and Shaobo Zhang and Yaonan Wang and Nanshan Zhong and Jianxing He, Modified SEIR and AI prediction of the epidemics trend of COVID-19 in China under public health interventions, Journal of Thoracic Disease, 2020, <https://jtd.amegroups.com/article/view/36385>
- [31] Zhao, Yujie and Yan, Hao and Holte, Sarah E and Kerani, Roxanne P and Mei, Yajun, Rapid detection of hot-spot by tensor decomposition with application to weekly gonorrhea data, Springer, International Workshop on Intelligent Statistical Quality Control, 2019, <https://doi.org/10.1002/nla.2136>
- [32] De Lathauwer, Lieven and De Moor, Bart and Vandewalle, Joos, A Multilinear Singular Value Decomposition, SIAM Journal on Matrix Analysis and Applications, 2000, <https://doi.org/10.1137/S0895479896305696>
- [33] Comon, Pierre and Luciani, Xavier and de Almeida, André, Tensor Decompositions, Alternating Least Squares and other Tales, Journal of Chemometrics, 2009, 10.1002/cem.1236
- [34] Chung, Julianne and Gazzola, Silvia, Flexible Krylov Methods for lp Regularization, SIAM Journal on Scientific Computings, 2019, 10.1137/18M1194456