COPA: Constrained PARAFAC2 for Sparse & Large Datasets

Ardavan Afshar,
Georgia Institute of Technology

Ioakeim Perros,
Georgia Institute of Technology

Evangelos E. Papalexakis,
University of California, Riverside

Elizabeth Searles,
Children's Healthcare Of Atlanta

Joyce Ho,
Emory University

Jimeng Sun
Georgia Institute of Technology

Abstract

PARAFAC2 has demonstrated success in modeling irregular tensors, where the tensor dimensions vary across one of the modes. An example scenario is modeling treatments across a set of patients with the varying number of medical encounters over time. Despite recent improvements on unconstrained PARAFAC2, its model factors are usually dense and sensitive to noise which limits their interpretability. As a result, the following open challenges remain: a) various modeling constraints, such as temporal smoothness, sparsity and non-negativity, are needed to be imposed for interpretable temporal modeling and b) a scalable approach is required to support those constraints efficiently for large datasets.

To tackle these challenges, we propose a *CO*nstrained *PA*RAFAC2 (COPA) method, which carefully incorporates optimization constraints such as temporal smoothness, sparsity, and non-negativity in the resulting factors. To efficiently support all those constraints, COPA adopts a hybrid optimization framework using alternating optimization and alternating direction method of multiplier (AO-ADMM). As evaluated on large electronic health record (EHR) datasets with hundreds of thousands of patients, COPA achieves significant speedups (up to 36× faster) over prior PARAFAC2 approaches that only attempt to handle a subset of the constraints that COPA enables. Overall, our method outperforms all the baselines attempting to handle a subset of the constraints in terms of speed, while achieving the same level of accuracy.

Through a case study on temporal phenotyping of medically complex children, we demonstrate how the constraints imposed by COPA reveal concise phenotypes and meaningful temporal profiles of patients. The clinical interpretation of both the phenotypes and the temporal profiles was confirmed by a medical expert.

Keywords

Tensor Factorization; Unsupervised Learning; Computational Phenotyping

1 INTRODUCTION

Tensor factorization encompasses a set of powerful analytic methods that have been successfully applied in many application domains: social network analysis [2, 18], urban planning [3], and health analytics [14, 15, 21, 22, 27]. Despite the recent progression on modeling the time through regular tensor factorization approaches [3, 19], there are some cases where modeling the time mode is intrinsically difficult for the regular tensor factorization methods, due to its irregularity. A concrete example of such irregularity is electronic health record (EHR). EHR datasets consist of K patients where patient k is represented using a matrix X_k and for each patient, J medical features are recorded. Patient k can have I_k hospital visits over time, which can be of different size across patients as shown in Figure 1.

In this case, clinical visits are the irregular time points which vary across patients. In particular, the *time irregularity* lies in 1) the variable number of hospital visits, 2) the varying disease states for different patients, and 3) the varying time gaps between consecutive hospital visits. To handle such irregular tensors, the state-of-the-art tensor model is PARAFAC2 [11], which naturally handles variable size along one of its modes (e.g., time mode). Despite the wide range of PARAFAC2 applications (e.g., natural language processing [8], chemical processing [28], and social sciences [12]) its computational requirements have limited its usage for small and dense datasets [17]. Even if recently, a scalable PARAFAC2 fitting algorithm was proposed for large, sparse data [22], it cannot incorporate meaningful constraints on the model factors such as: a) sparsity, which facilitates model inspection and understanding and b) smoothness, which is meaningful to impose when temporal evolution is modeled as a mode of the input tensor.

To tackle the above challenges, we propose the COnstrained PARAFAC2 method (COPA), which introduces various useful constraints in PARAFAC2 modeling. In particular, generalized temporal smoothness constraints are integrated in order to: a) properly model temporally-evolving phenomena (e.g., evolving disease states), and b) adaptively deal with uneven spacing along the temporal dimension (e.g., when the time duration between consecutive hospital visits may range from 1 day to several years). Also, COPA introduces sparsity into the latent factors, a crucial property enhancing interpretability for sparse input data, such as the EHR.

A key property of our approach is that those constraints are introduced in a computationally efficient manner. To do so, COPA adopts a hybrid optimization framework using alternating

optimization and alternating direction method of multipliers. This enables our approach to achieve significant speedups (up to 36×) over baselines supporting only a specific constraint each, while achieving the same level of accuracy. Through both quantitative (e.g., the percentage of sparsity) and qualitative evaluations from a clinical expert, we demonstrate the meaningfulness of the constrained output factors for the task of temporal phenotyping via EHRs. In summary, we list our main contributions below:

- **Constrained PARAFAC2:** We propose COPA, a method equipping the PARAFAC2 modeling with a variety of meaningful constraints such as smoothness, sparsity, and non-negativity.

- **Scalable PARAFAC2:** While COPA incorporates a wide range of constraints, it is faster and more scalable than baselines supporting only a subset of those constraints.

- **COPA for temporal phenotyping:** We apply COPA for temporal phenotyping of a medically complex population; a medical expert confirmed the clinical meaningfulness of the extracted phenotypes and temporal patient profiles.

Table 1 summarizes the contributions in the context of existing works.

2 BACKGROUND

In this Section, we provide the necessary background for tensor operations. Then, we briefly illustrate the related work including: the classical method for PARAFAC2 and AO-ADMM framework for constrained tensor factorization. Table 2 summarizes the notations used throughout the paper.

The *mode* or *order* is the number of dimensions of a tensor. A *slice* refers to a matrix derived from the tensor where fixing all modes but two. *Matricization* converts the tensor into a matrix representation without altering its values. The mode-n matricization of $\mathcal{Y} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is denoted as $Y_{(n)} \in \mathbb{R}^{I_n \times I_1 \cdots I_{n-1} I_{n+1} \cdots I_N}$. Matricized-Tensor-Times-Khatri-Rao-Product[4] (MTTKRP) is a multiplication which a naive construction of that for large and sparse tensors needs computational cost and enormous memory and is the typical bottleneck in most tensor factorization problems. The popular CP decomposition [7] also known as PARAFAC factorizes a tensor into a sum of R rank-one tensors. CP decomposition method factorizes tensor $\mathcal{Y} \in \mathbb{R}^{K \times J \times I}$ into $\sum_{r=1}^{R} a_r \circ b_r \circ c_r$ where R is the number of target-ranks or components and $a_r \in \mathbb{R}^K$, $b_r \in \mathbb{R}^J$, and $c_r \in \mathbb{R}^I$ are column matrices and $\circ$ indicates the outer product. Here $A = [a_1, \ldots a_R]$, $B = [b_1, \ldots b_R]$, and $C = [c_1, \ldots c_R]$ are factor matrices.

Original PARAFAC2 model

As proposed in [11], the PARAFAC2 model decomposes each slice of the input $X_k \in \mathbb{R}^{I_k \times J}$ as $X_k \approx U_k S_k V^T$, where $U_k \in \mathbb{R}^{I_k \times R}$, $S_k \in \mathbb{R}^{R \times R}$ is a diagonal matrix, and $V \in \mathbb{R}^{J \times R}$. Uniqueness is an important property in factorization models which ensures that the pursued solution is not an arbitrarily rotated version of the actual latent factors. In order to enforce

uniqueness, Harshman [11] imposed the constraint $U_k^T U_k = \Phi \quad \forall k$. This is equivalent to each U_k being decomposed as $U_k = Q_K H$, where $Q_k \in \mathbb{R}^{I_k \times R}$, $Q_k^T Q_k = I \in \mathbb{R}^{R \times R}$ and $H \in \mathbb{R}^{R \times R}$. Note that Q_k has orthonormal columns and H is invariant regardless of k. Therefore, the decomposition of U_k implicitly enforces the constraint as $U_k^T U_k = H^T Q_k^T Q_k H = H^T H = \Phi$. Given the above modeling, the standard algorithm [17] to fit PARAFAC2 for dense input data tackles the following optimization problem:

$$\underset{\{U_k\}, \{S_k\}, V}{\text{minimize}}, \quad \sum_{k=1}^{K} \frac{1}{2} \left\| X_k - U_k S_k V^T \right\|_F^2 \tag{1}$$

subject to $U_k = Q_K H$, $Q_k^T Q_k = I$, and S_k is diagonal. The solution follows an Alternating Least Squares (ALS) approach to update the modes. First, orthogonal matrices $\{Q_k\}$ are solved by fixing $H, \{S_k\}, V$ and posing each Q_k as an individual Orthogonal Procrustes Problem [24]:

$$\underset{\{Q_k\}}{\text{minimize}} \quad \frac{1}{2} \left\| X_k - Q_k H S_k V^T \right\|_F^2 \tag{2}$$

Computing the singular value decomposition (SVD) of $H S_k V^T X_k^T = P_k \Sigma_k Z_k^T$ yields the optimal $Q_k = P_k Z_k^T$. With $\{Q_k\}$ fixed, the remaining factors can be solved as:

$$\underset{H, \{S_k\}, V}{\text{minimize}} \quad \frac{1}{2} \sum_{k=1}^{K} \left\| Q_k^T X_k - H S_k V^T \right\|_F^2 \tag{3}$$

The above is equivalent to the CP decomposition of tensor $\mathcal{Y} \in \mathbb{R}^{R \times J \times K}$ with slices $Y_k = Q_k^T X_k$. A single iteration of the CP-ALS provides an update for $H, \{S_k\}, V$ [17]. The algorithm iterates between the two steps (Equations 2 and 3) until convergence is reached.

AO-ADMM

Recently, a hybrid algorithmic framework, AO-ADMM [16], was proposed for constrained CP factorization based on alternating optimization (AO) and the alternating direction method of multipliers (ADMM). Under this approach, each factor matrix is updated iteratively using ADMM while the other factors are fixed. A variety of constraints can be placed on the factor matrices, which can be readily accommodated by ADMM.

3 PROPOSED METHOD: CONSTRAINED PARAFAC2 FRAMEWORK

A generalized constrained PARAFAC2 approach is appealing from several perspectives including the ability to encode prior knowledge, improved interpretability, and more robust and reliable results. We propose COPA, a scalable and generalized constrained PARAFAC2 model, to impose a variety of constraints on the factors. Our algorithm leverages AO-ADMM style iterative updates for some of the factor matrices and introduces several

PARAFAC2-specific techniques to improve computational efficiency. Our framework has the following benefits:

- Multiple constraints can be introduced simultaneously.

- The ability to handle large data as solving the constraints involves the application of several element-wise operations.

- Generalized temporal smoothness constraint that effectively deals with uneven spacing along the temporal (irregular) dimension (gaps from a day to several years for two consecutive clinical visits).

In this section, we first illustrate the general framework for formulating and solving the constrained PARAFAC2 problem. We then discuss several special constraints that are useful for the application of phenotyping including sparsity on the V, smoothness on the U_k, and non-negativity on the S_k factor matrices in more detail.

3.1   General Framework for COPA

The constrained PARAFAC2 decomposition can be formulated using generalized constraints on H, S_k, and V, in the form of $c(H)$, $c(S_k)$, and $c(V)$ as:

$$\underset{\{U_k\},\,\{S_k\},\,V}{\text{minimize}} \sum_{k=1}^{K} \frac{1}{2} \left\| X_k - U_k S_k V^T \right\|_F^2 + c(H) + \sum_{k=1}^{K} c(S_k) + c(V)$$

subject to $U_k = Q_K H$, $Q_k^T Q_k = I$, and S_k is diagonal. To solve for those constraints, we introduce auxiliary variables for H, S_k, and V (denoted as $\overline{H}$, $\overline{S_k}$, and $\overline{V}$). Thus, the optimization problem has the following form:

$$\begin{aligned} \underset{\{U_k\},\,\{S_k\},\,V}{\text{minimize}} \sum_{k=1}^{K} \frac{1}{2} & \left\| X_k - U_k S_k V^T \right\|_F^2 + c(\overline{H}) + \sum_{k=1}^{K} c(\overline{S_k}) + c(\overline{V}) \\ \text{subject to} \quad & U_k = Q_k H, Q_k^T Q_k = I, S_k = \overline{S_k} \text{ for all k} = 1, \ldots, K \\ & H = \overline{H}, V = \overline{V} \end{aligned}$$

(4)

We can re-write the objective function as the minimization of $tr\left(X_k^T X_k\right) - 2tr\left(X_k^T Q_k H \, S_k V^T\right) + tr\left(V \, S_k H^T Q_k^T Q_k H \, S_k V^T\right)$ in terms of Q_k. The first term is constant and since Q_k has orthonormal columns ($Q_k^T Q_k = I$), the third term is also constant. By rearranging the terms we have $tr\left(X_k^T Q_k H \, S_k V^T\right) = tr\left(X_k V \, S_k H^T Q_k^T\right)$. Thus, the objective function regarding to Q_k is equivalent to:

$$\begin{aligned} \underset{Q_k}{\text{minimize}} \quad & \frac{1}{2} \left\| X_k V \, S_k H^T - Q_k \right\|_F^2 \\ \text{subject to} \quad & Q_k^T Q_k = I \end{aligned}$$

(5)

Thus, the optimal Q_k has the closed form solution $Q_k = B_k C_k^T$ where $B_k \in R^{I_k \times R}$ and $C_k \in R^{R \times R}$ are the right and left singular vectors of $X_k \, V \, S_k \, H^T$ [10, 24]. This promotes the

solution's uniqueness, since orthogonality is essential for uniqueness in the unconstrained case.

Given fixed $\{Q_k\}$, we next find the solution for H, $\{S_k\}$, V as follows:

$$
\begin{aligned}
&\underset{H, \{S_k\}, V}{\text{minimize}} \quad \frac{1}{2} \sum_{k=1}^{K} \left\| Q_k^T X_k - H S_k V^T \right\|_F^2 + c(\overline{H}) + \sum_{k=1}^{K} c(\overline{S_k}) + c(\overline{V}) \\
&\text{subject to} \quad S_k = \overline{S_k} \quad \text{for all } k = 1, \ldots, K \\
&\quad H = \overline{H}, V = \overline{V}
\end{aligned}
\tag{6}
$$

This is equivalent to performing a CP decomposition of tensor $\mathcal{Y} \in \mathbb{R}^{R \times J \times K}$ with slices $Y_k = Q_k^T X_k$ [17]. Thus, the objective is of the form:

$$
\begin{aligned}
&\underset{H, W, V}{\text{minimize}} \quad \frac{1}{2} \left\| \mathcal{Y} - [H; V; W] \right\|_F^2 + c(\overline{H}) + c(\overline{V}) + c(\overline{W}) \\
&H = \overline{H}, V = \overline{V}, W = \overline{W}
\end{aligned}
\tag{7}
$$

We use the AO-ADMM approach [16] to compute H, V, and W, where $S_k = diag(W(k, :))$ and $W \in \mathbb{R}^{K \times R}$. Each factor matrix update is converted to a constrained matrix factorization problem by performing the mode-n matricization of $Y_{(n)}$ in Equation 7. As the updates for H, W, and V take on similar forms, we will illustrate the steps for updating W. Thus, the equivalent objective for W using the 3rd mode matricization of $\mathcal{Y} \left(Y_{(3)} \in \mathbb{R}^{K \times RJ} \right)$ is:

$$
\begin{aligned}
&\underset{W^T}{\text{minimize}} \quad \frac{1}{2} \left\| Y_{(3)}^T - (V \odot H) W^T \right\|_F^2 + c(\overline{W}) \\
&\text{subject to} \quad W^T = \overline{W}
\end{aligned}
\tag{8}
$$

The application of ADMM yields the following update rules:

$$
W^T := \left(\left(H^T H * V^T V \right) + \rho I \right)^{-1} \left(Y_{(3)}(V \odot H) + \rho(\overline{W} + D_W^T) \right)^T
$$

$$
\overline{W} := \underset{\overline{W}}{\text{argmin}} \, c(\overline{W}) + \frac{\rho}{2} \left\| \overline{W} - W^T + D_W^T \right\|_F^2
$$

$$
D_W^T := D_W^T + \overline{W} - W^T
$$

where D_W^T is a dual variable and ρ is a step size regarding to W^T factor matrix. The auxiliary variable ($\overline{W}$) update is known as the proximity operator [20]. Parikh and Boyd show that for a wide variety of constraints, the update can be computed using several element-wise operations. In Section 3.3, we discuss the element-wise operations for three of the constraints we consider.

3.2 Implementation Optimization

In this section, we will provide several steps to accelerate the convergence of our algorithm. First, our algorithm needs to decompose $\mathcal{Y}$, therefore, MTTKRP will be a bottleneck for sparse input. Thus COPA uses the fast MTTKRP proposed in SPARTan [22]. Second, $(H^T H * V^T V) + \rho I$ is a symmetric positive definite matrix, therefore instead of calculating the expensive inverse computation, we can calculate the Cholesky decomposition of it (LL^T) where L is a lower triangular matrix and then apply the inverse on L (lines 14,16 in algorithm 1). Third, $Y_{(3)}(V \odot H)$ remains a constant and is unaffected by updates to W or $\overline{W}$. Thus, we can cache it and avoid unnecessary re-computations of this value. Fourth, based on the AO-ADMM results and our own preliminary experiments, our algorithm sets

$$\rho = \frac{\|H^T H + V^T V\|_F^2}{R} \text{ for fast convergence of } W.$$

Algorithm 1 lists the pseudocode for solving the generalized constrained PARAFAC2 model. Adapting AO-ADMM to solve H, W, and V in PARAFAC2 has two benefits: (1) a wide variety of constraints can be incorporated efficiently with iterative updates computed using element-wise operations and (2) computational savings gained by caching the MTTKRP multiplication and using the Cholesky decomposition to calculate the inverse.

3.3 Examples of useful constraints

Next, we describe several special constraints which are useful for many applications and derive the updates rules for those constraints.

3.3.1 Smoothness on U_k : For longitudinal data such as EHRs, imposing latent components that change smoothly over time may be desirable to improve interpretability and robustness (less fitting to noise). Motivated by previous work [12, 26], we incorporate temporal smoothness to the factor matrices U_k by approximating them as a linear combination of several smooth functions. In particular, we use M-spline, a non-negative spline function which can be efficiently computed through a recursive formula [23]. For each subject k, a set of M-spline basis functions $\left(M_k \in \mathbb{R}^{I_k \times l}\right)$ are created where l is the number of basis functions. Thus, U_k is an unknown linear combination of the smooth basis functions $U_k = M_k W_k$, where W_k is the unknown weight matrix.

The temporal smoothness constrained solution is equivalent to performing the PARAFAC2 algorithm on a projected $X'_k = C_k^T X_k$, where C_k is obtained from the SVD of $M_k = \left[C_k, O_k, P_k^T\right]$. We provide proof of the equivalence by analyzing the update of Q_k for the newly projected data (X'_k) :

$$\underset{Q_k}{\text{minimize}} \left\|C_k^T X_k - Q_k H S_k V^T\right\|_F^2$$

Algorithm 1

COPA

Input: $X_k \in \mathbb{R}^{I_k \times J}$ for k=1,...,K and target rank R

Output: $U_k \in \mathbb{R}^{I_k \times R}$, $S_k \in \mathbb{R}^{R \times R}$ for $k = 1,...,$ K, $V \in \mathbb{R}^{J \times R}$

1: Initialize H, V, $\{S_k\}$ for k=1,...,K

2: **while** convergence criterion is not met **do**

3:   **for** k=1,...,K **do**

4:     $\left[B_k, D_k, C_k^T \right]$ =truncated SVD of $X_k V\, S_k H^T$

5:     $Q_k = B_k C_k^T$

6:     $Y_k = Q_k^T X_k$

7:     $W(k, :) = diag(S_k)$

8:   **end for**

9:   $Z_1 = H, Z_2 = W, Z_3 = V$

10:   **for** n=1,...,3 **do**

11:     $G = *_{i \neq n} Z_i^T Z_i$

12:     $F = Y_{(n)}(\odot_{i\,n} Z_i)$ //calculated based on [22]

13:     $\rho = trace\,(G)/R$

14:     $L =$ Cholesky$(G + \rho I)$

15:     **while** convergence criterion is not met **do**

16:       $Z_n^T = \left(L^T \right)^{-1} L^{-1} \left(F + \rho \left(\overline{Z_n} + D_{Z_n} \right) \right)^T$

17:       $\overline{Z_n} := \arg\min_{\overline{Z_n}} c\left(\overline{Z_n} \right) + \frac{\rho}{2} \left\| \overline{Z_n} - Z_n^T + D_{Z_n} \right\|_F^2$

18:       $D_{Z_n} := D_{Z_n} + \overline{Z_n} - Z_n^T$

19:     **end while**

20:   **end for**

21: **end while**

22: $H = \overline{Z_1}, W = \overline{Z_2}, V = \overline{Z_3}$

23: **for** k=1,...,K **do**

24:   $U_k = Q_k H$

25:   $S_k = diag(W(k, :))$

26: **end for**

This can be re-written as the minimization of

$tr\left(X_k^T C_k C_k^T X_k \right) - 2tr\left(X_k^T C_k Q_k H\, S_k V^T \right) + tr\left(V S_k H^T Q_k^T Q_k H\, S_k V^T \right)$. Since C_k and Q_k have orthonormal columns the first and third terms are constants. Also $tr(A^T) = tr(A)$ and $tr(ABC) = tr(CAB) = tr(BCA)$, thus the update is equivalent to:

$$\max_{Q_k} \ tr\left(C_k^T X_k V \ S_k H^T Q_k^T\right) = tr\left(X_k' V \ S_k H^T Q_k^T\right)$$

$$\Leftrightarrow \min_{Q_k} \ \left\|X_k' V \ S_k^H - Q_k\right\|_F^2$$

(10)

This is similar to equation 5 (only difference is X_k', a projection of X_k) which can be solved using the constrained quadratic problem [10]. Thus, solving for H, S_k, and V remains the same. The only difference is that after convergence, U_k is constructed as $C_k Q_k H$.

In some domains, there may be uneven time gaps between the observations. For example, in our motivating application, patients may not regularly visit a healthcare provider but when they do, the visits are closely clustered together. To adaptively handle time-varying gaps, we alter the basis functions to account for uneven gaps. Under the assumption of small and equidistant gaps (Helwig's approach), the basis functions are created directly on the visits. Instead, we define a set of M-spline functions for each patient in the interval $[t_1, t_n]$, where t_1 and t_n are the first and last hospital visits. These spline functions (with day-resolution) are then transformed to their visit-resolution. Thus, given the number and position of the knots ($[\beta_1.\beta_m]$), which can be estimated using the metric introduced in Helwig's work [12], we create the ith basis function of patient k with degree d using the following recursive formula:

$$m_{ik,d}(t) = \frac{t - \beta_i}{\beta_{i+d} - \beta_i} m_{ik,d-1}(t) + \frac{\beta_{i+d+1} - t}{\beta_{i+d+1} - \beta_{i+1}} m_{i+1k,d-1}(t)$$

where t denotes the hospital visit day and β_i is the ith knot. Hence, we can reconstruct the basis functions as $m_{ik,0}(t)$ is 1 if $t \in [\beta_i, \beta_{i+1}]$ and zero otherwise. Figure 2 shows the two types of basis functions related to a patient with sickle cell anemia.

3.3.2 Sparsity on V : Sparsity constraints have wide applicability to many different domains and have been exploited for several purposes including improved interpretability, reduced model complexity, and increased robustness. For the purpose of EHR-phenotyping, we impose sparsity on factor matrix V, to obtain sparse phenotype definitions. While several sparsity inducing constraints can be introduced, we focus on the ℓ_0 and ℓ_1 norms, two popular regularization techniques. The ℓ_0 regularization norm, also known as hard thresholding, is a non-convex optimization problem that caps the number of non-zero values in a matrix. The ℓ_0 regularization norm, or the soft thresholding metric, is often used as a convex relaxation of the ℓ_0 norm. The objective function with respect to V for the sparsity (ℓ_0 norm) constrained PARAFAC2 is as follows:

$$\underset{V}{\text{minimize}} \quad \frac{1}{2}\|\mathcal{Y} - [H; V; W]\|_F^2 + \lambda\|\bar{V}\|_0, \text{ s.t. } V = \bar{V}$$

(11)

where λ is a regularization parameter which needs to be tuned. The proximity operator for the ℓ_0 regularization, $\bar{V}$, uses the hard-thresholding operator which zeros out entries below a specific value. Thus, the update rules for V are as follows:

$$V^T := \left(\left(H^T H * W^T W\right) + \rho I\right)^{-1}\left(Y_{(2)}(W \odot H) + \rho(\overline{V} + D_V{}^T)\right)^T$$

$$\overline{V} := \arg\min_{\overline{V}} \lambda\|\overline{V}\|_0 + \frac{\rho}{2}\left\|\overline{V} - V^T - D_V{}^T\right\|_F^2 = \begin{cases} 0 & \overline{V}^2 \leq \frac{2\lambda}{\rho} = \mu \\ \overline{V} & \overline{V}^2 \geq \frac{2\lambda}{\rho} = \mu \end{cases}$$

$$D_V{}^T := D_V{}^T + \overline{V} - V^T$$

The update rule corresponding to the ℓ_1 regularization, $\overline{V}$, is the soft-thresholding operator:

$$\overline{V} := \arg\min_{\overline{V}} \lambda\|\overline{V}\|_1 + \frac{\rho}{2}\left\|\overline{V} - V^T - D_V{}^T\right\|_F^2 = \max\left(0, \left|V^T + D_V{}^T\right| - (\lambda/\rho)\right)$$

Note that imposing factor sparsity boils down to using element-wise thresholding operations, as can be observed above. Thus, imposing sparsity is scalable even for large datasets.

3.3.3 Non-negativity on S_k : COPA is able to impose non-negativity constraint to factor matrices H, S_k, and V. Because the updating rules for these three factor matrices are same we just show the update rules for factor matrix S_k for simplicity ($S_k = diag(W(k, :))$):

$$W^T := \left(\left(H^T H * V^T V\right) + \rho I\right)^{-1}\left(Y_{(3)}(V \odot H) + \rho(\overline{W} + D_W{}^T)\right)^T$$

$$\overline{W} := max\left(0, W^T - D_W{}^T\right)$$

$$D_W{}^T := D_W{}^T + \overline{W} - W^T$$

Note that our update rule for $\overline{W}$ only involves zeroing out the negative values and is an element-wise operation. The alternating least squares framework proposed by [6] and employed by SPARTan [22] can also achieve non-negativity through non-negative least squares algorithms but that is a more expensive operation than our scheme.

4 EXPERIMENTAL RESULTS

In this section, we first provide the description of the real datasets. Then we give an overview of baseline methods and evaluation metrics. After that, we present the quantitative experiments. Finally, we show the success of our algorithm in discovering temporal

signature of patients and phenotypes on a subset of medically complex patients from a real data set.

4.1   Setup

4.1.1   Data Set Description.

Children's Healthcare of Atlanta (CHOA): This dataset contains the EHRs of 247,885 pediatric patients with at least 3 hospital visits. For each patient, we utilize the International Classification of Diseases (ICD9) codes [25] and medication categories from their records, as well as the provided age of the patient (in days) at the visit time. To improve interpretability and clinical meaningfulness, ICD9 codes are mapped into broader Clinical Classification Software (CCS) [1] categories. Each patient slice X_k records the clinical observations and the medical features. The resulting tensor is 247,885 patients by 1388 features by maximum 857 observations.

Centers for Medicare and Medicaid (CMS):[1]: CMS released the Data Entrepreneurs Synthetic Public Use File (DE-SynPUF), a realistic set of claims data that also protects the Medicare beneficiaries' protected health information. The dataset is based on 5% of the Medicare beneficiaries during the period between 2008 and 2010. Similar to CHOA, we extracted ICD9 diagnosis codes and summarized them into CCS categories. The resulting number of patients are 843,162 with 284 features and the maximum number of observations for a patient are 1500.

Table 3 provides the summary statistics of real datasets.

4.1.2   Baseline Approaches.—In this section, we briefly introduce the baseline that we compare our proposed method.

- **SPARTan** [22][2] is a recently-proposed methodology for fitting PARAFAC2 on large and sparse data. The algorithm reduces the execution time and memory footprint of the bottleneck MTTKRP operation. Each step of SPARTan updates the model factors in the same way as the classic PARAFAC2 model [17], but is faster and more memory efficient for large and sparse data. In the experiments, SPARTan has non-negativity constraints on H, S_k, and V factor matrices.

- **Helwig** [12] incorporates smoothness into PARAFAC2 model by constructing a library of smooth functions for every subject and apply smoothness based on linear combination of library functions. We implemented this algorithm in MATLAB.

4.1.3   Evaluation Metrics.—We use FIT [6] to evaluate the quality of the reconstruction based on the model's factors:

---

[1]https://www.cms.gov/Research-Statistics-Data-and-Systems/Downloadable-Public-Use-Files/SynPUFs/DE_Syn_PUF.html
[2]The MATLAB code is available at https://github.com/kperros/SPARTan

$$FIT = 1 - \frac{\sum_{k=1}^{K} \left\| X_k - U_k S_k V^T \right\|^2}{\sum_{k=1}^{K} \left\| X_k \right\|^2}$$

The range of FIT is between $[-\infty, 1]$ and values near 1 indicate the method can capture the data perfectly. We also use SPARSITY metric to evaluate the factor matrix V which is as follows:

$$SPARSITY = \frac{nz(V)}{size(V)}$$

where $nz(V)$ is the number of zero elements in V and $size(V)$ is the number of elements in V. Values near 1 implies the sparsest solution.

4.1.4 Implementation details.—COPA is implemented in MATLAB and includes functionalities from the Tensor Toolbox [5]. To enable reproducibility and broaden the usage of the PARAFAC2 model, our implementation is *publicly available at:* https://github.com/aafshar/COPA. All the approaches (including the baselines) are evaluated on MatlabR2017b. We also implemented the smooth and functional PARAFAC2 model [12], as the original approach was only available in R [13]. This ensures a fair comparison with our algorithm.

4.1.5 Hardware.—The experiments were all conducted on a server running Ubuntu 14.04 with 250 GB of RAM and four Intel E5–4620 v4 CPU's with a maximum clock frequency of 2.10GHz. Each processor contains 10 cores. Each core can exploit 2 threads with hyper-threading enabled.

4.1.6 Parallelism.—We utilize the capabilities of Parallel Computing Toolbox of Matlab by activating parallel pool for all methods. For CHOA dataset, we used 20 workers whereas for CMS we used 30 workers because of more number of non-zero values.

4.2 Quantitative Assessment of Constraints

To understand how different constraints affect the reconstruction error, we perform an experiment using each of the constraints introduced in Section 3.3. We run each method for 5 different random initializations and provide the average and standard deviation of FIT as shown in Figure 3. This Figure illustrates the impact of each constraint on the FIT values across both datasets for two different target ranks (R={15,40}). In all versions of COPA, S_k factor matrix is non-negative. Also, we apply smoothness on U_k and ℓ_0 regularization norm on V separately and also simultaneously. From Figure 3, we observe that different versions of COPA can produce a comparable value of FIT even with both smoothness on U_k and sparsity on V. The number of smooth basis functions are selected based on the cross-validation metric introduced in [26] and the ℓ_0 regularization parameter (μ) is selected via grid search by finding a good trade off between FIT and SPARSITY metric. The optimal values of each parameter for the two different data sets and target ranks are reported in table 4.

We next quantitatively evaluate the effects of sparsity (average and standard deviation of the sparsity metric) by applying ℓ_0 regularization norm on the factor matrix V for COPA and compare it with SPARTan for 5 different random initializations, as provided in Table 5. For both the CHOA and CMS datasets, COPA achieves more than a 98% sparsity level. The improved sparsity of the resulting factors is especially prominent in the CMS dataset, with a 400% improvement over SPARTan. Sparsity can improve the interpretability and potentially the clinical meaningfulness of phenotypes via more succinct patient characterizations. The quantitative effectiveness is further supported by the qualitative endorsement of a clinical expert (see Section 4.4).

4.3 Scalability and FIT-TIME

First, we evaluate and compare the total running time of all versions of COPA and SPARTan on the real datasets. We run each method 5 times and report averages and standard deviations. As shown in Figure 4, the average of total running time of COPA with non-negativity constraints imposed on H, $\{S_k\}$, V is faster (up to 1.57×) than SPARTan with the same set of constraints for two data sets and different target ranks. In order to provide more precise comparison we apply paired t-tests on the two sets of running time, one from SPARTan and the other from a version of COPA under the null hypothesis that the running times are not significantly different between the two methods. We present the p-values return from the t-tests in Table 6. The p-values for COPA with non-negativity constraint and sparsity constraint are small which suggest that the version of COPA is significantly better than the SPARTan (rejecting the null hypothesis). Also we provide the speedups (running time of SPARTan divide by running time of COPA) in Table 6. Moreover, the average running times of Smooth COPA are just slightly slower than SPARTan, which does not support such smooth constraints. Next, we compare the best convergence (Time in seconds versus FIT) out of 5 different random initializations of the proposed COPA approach against SPARTan. For both methods, we add non-negativity constraints to H, $\{S_k\}$, V and compare the convergence rates on both real-world datasets for two different target ranks ($R = \{15, 40\}$). Figures 5 and 6 illustrates the results on the CHOA and CMS datasets respectively. COPA converges faster than SPARTan in all cases. While both COPA and SPARTan avoid direct construction of the sparse tensor $\mathcal{Y}$, the computational gains can be attributed to the efficiency of the non-negative proximity operator, an element-wise operation that zeros out the negative values in COPA whereas SPARTan performs expensive NN-Least Square operation. Moreover, caching the MTTKRP operation and the Cholesky decomposition of the Gram matrix help COPA to reduce the number of computations.

In addition, we assessed the scalability of incorporating temporal smoothness onto U_k and compare it with Helwig's approach [12] as SPARTan does not have the smoothness constraint. Figure 7 provides a comparison of iteration time for Smooth COPA and the approach in [12] across two different target ranks. First, we remark that our method is more scalable and faster than the baseline. For $R = 30$, COPA is 27× and 36× faster on CHOA and CMS respectively. Moreover, for $R = 40$, not only was COPA 32× faster on CHOA, but the execution failed using the approach in [12] on CMS because of the excessive amount of memory required. In contrast, COPA successfully finished each iteration with an average of 224.21 seconds.

4.4   Case Study: CHOA Phenotype Discovery

4.4.1   Model interpretation: Phenotyping is the process of extracting a set of meaningful medical features from raw and noisy EHRs. We define the following model interpretations regarding to our target case study:

- Each column of factor matrix V represents a phenotype and each row indicates a medical feature. Therefore an entry $V(i, j)$ represents the membership of medical feature i to the jth phenotype.

- The rth column of $U_k \in \mathbb{R}^{I_k \times R}$ indicates the evolution of phenotype r for all I_k clinical visits for patient k.

- The diagonal S_k provides the importance membership of R phenotypes for the patient k. By sorting the values $diag(S_k)$ we can identify the most important phenotypes for patient k.

4.4.2   Case Study Setup: For this case study, we incorporate smoothness on U_k, non-negativity on S_k, and sparsity on V simultaneously to extract phenotypes from a subset of medically complex patients from CHOA dataset. These are the patients with high utilization, multiple specialty visits and high severity. A total of 4602 patients are selected with 810 distinct medical features. For this experiment, we set the number of basis functions to 7 (as shown in figure 2), $\mu = 49$, and $R = 4$.

4.4.3   Findings: We demonstrate the effectiveness of COPA for extracting phenotypes. Also we show how COPA is able to describe the evolution of phenotypes for patients by considering the gap between every pair of clinical visits. Figure 8 displays the evolution of phenotypes (temporal pattern) relating to two patients discovered by COPA, Helwig, and SPARTan. The phenotype that is chosen has the highest weight for each patient (largest value in the diagonal S_k matrix) and the loadings on the medical features are similar across all three methods. The first row in figure 8 is from a patient who has sickle cell anemia. There is a large gap between the 19th and 20th visits (742 days or ~ 2 years) with a significant increase in the occurrence of medications/diagnosis in the patient's EHR record. COPA models this difference and yields phenotype loadings that capture this drastic change. On the other hand, the factor resulting from Helwig's approach assumes the visits are close in time and produce the same magnitude for the next visit. The second row in figure 8 reflects the temporal signature for a patient with Leukemia. In the patient's EHRs, the first visit occurred on day 121 without any sign of Leukemia. The subsequent visit (368 days later) reflects a change in the patient's status with a large number of diagnosis and medications. COPA encapsulates this phenomenon, while the Helwig factor suggests the presence of Leukemia at the first visit which is not present. Although SPARTan produces temporally-evolving phenotypes, it treats time as a categorical feature. Thus, there are sudden spikes in the temporal pattern which hinders interpretability and clinical meaningfulness.

Next, we present the phenotypes discovered by COPA in table 7. It is important to note that no additional post-processing was performed on these results. These four phenotypes have

been endorsed by a clinical expert as clinically meaningful. Moreover, the expert has provided the labels to reflect the associated medical concept. As the phenotypes discovered by SPARTan and Helwig are too dense and require significant post-processing, they are not displayed in this paper.

5 RELATED WORK

SPARTan was proposed for PARAFAC2 modeling on large and sparse data [22]. A specialized Matricized-Tensor-Times-Khatri-Rao-Product (MTTKRP) was designed to efficiently decompose the tensor $\mathscr{Y}\left(Y_k = Q_k^T X_k\right)$ both in terms of speed and memory. Experimental results demonstrate the scalability of this approach for large and sparse datasets. However, the target model and the fitting algorithm do not enable imposing constraints such as smoothness and sparsity, which would enhance the interpretability of the model results.

A small number of works have introduced constraints (other than non-negativity) for the PARAFAC2 model.Helwig [12] imposed both functional and structural constraints. Smoothness (functional constraint) was incorporated by extending the use of basis functions introduced for CP [26]. Structural information (variable loadings) were formulated using Lagrange multipliers [9] by modifying the CP-ALS algorithm. Unfortunately, Helwig's algorithm suffers the same computational and memory bottlenecks as the classical algorithm designed for dense data [17]. Moreover, the formulation does not allow for easy extensions of other types of constraints (e.g., sparsity).

Other works that tackle the problem of computational phenotyping through constrained tensor factorization (e.g., [15, 27]) cannot handle irregular tensor input (as summarized in Table 1); thus they are limited to aggregating events across time, which may lose temporal patterns providing useful insights.

6 CONCLUSION

Interpretable and meaningful tensor factorization models are desirable. One way to improve the interpretability of tensor factorization approaches is by introducing constraints such as sparsity, non-negativity, and smoothness. However, existing constrained tensor factorization methods are not well-suited for an irregular tensor. While PARAFAC2 is a suitable model for such data, there is no general and scalable framework for imposing constraints in PARAFAC2.

Therefore, in this paper we propose, COPA, a constrained PARAFAC2 framework for large and sparse data. Our framework is able to impose constraints simultaneously by applying element-wise operations. Our motivating application is extracting temporal patterns and phenotypes from noisy and raw EHRs. By incorporating smoothness and sparsity, we produce meaningful phenotypes and patient temporal signatures that are confirmed by a clinical expert.

ACKNOWLEDGMENT

REFERENCES

[1]. 2017 Clinical Classifications Software (CCS) for ICD-9-CM. https://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp (2017). Accessed: 2017-02-11.

[2]. Acar Evrim, Dunlavy Daniel M, and Kolda Tamara G. 2009 Link prediction on evolving data using matrix and tensor factorizations. In Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on IEEE, 262–269.

[3]. Afshar Ardavan, Ho Joyce C., Dilkina Bistra, Perros Ioakeim, Khalil Elias B., Xiong Li, and Sunderam Vaidy. 2017 CP-ORTHO: An Orthogonal Tensor Factorization Framework for Spatio-Temporal Data. In Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL'17) ACM, New York, NY, USA, Article 67, 4 pages. DOI: 10.1145/3139958.3140047

[4]. Bader Brett W and Kolda Tamara G. 2007 Efficient MATLAB computations with sparse and factored tensors. SIAM Journal on Scientific Computing 30, 1 (2007), 205–231.

[5]. Bader Brett W., Kolda Tamara G., and et al. 2015 MATLAB Tensor Toolbox Version 2.6 Available online. (2 2015). http://www.sandia.gov/~tgkolda/TensorToolbox/

[6]. Bro Rasmus, Andersson Claus A, and Kiers Henk AL. 1999 PARAFAC2-Part II. Modeling chromatographic data with retention time shifts. Journal of Chemometrics 13, 3–4 (1999), 295–309.

[7]. Carroll J Douglas and Chang Jih-Jie. 1970 Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. Psychometrika 35, 3 (1970), 283–319.

[8]. Chew Peter A, Bader Brett W, Kolda Tamara G, and Abdelali Ahmed. 2007 Cross-language information retrieval using PARAFAC2. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining ACM, 143–152.

[9]. Clarke Frank H. 1976 A new approach to Lagrange multipliers. Mathematics of Operations Research 1, 2 (1976), 165–174.

[10]. Golub Gene H and Van Loan Charles F. 2013 Matrix Computations. Vol. 3 JHU Press.

[11]. Harshman RA. 1972b PARAFAC2: Mathematical and technical notes. UCLA Working Papers in Phonetics 22 (1972b), 30–44.

[12]. Helwig Nathaniel E. 2017 Estimating latent trends in multivariate longitudinal data via Parafac2 with functional and structural constraints. Biometrical Journal 59, 4 (2017), 783–803. [PubMed: 28025850]

[13]. Helwig Nathaniel E and Helwig Maintainer Nathaniel E. 2017 Package â Ÿmul-tiwayâ . (2017).

[14]. Ho Joyce C, Ghosh Joydeep, Steinhubl Steve R, Stewart Walter F, Denny Joshua C, Malin Bradley A, and Sun Jimeng. 2014 Limestone: High-throughput candidate phenotype generation via tensor factorization. Journal of biomedical informatics 52 (2014), 199–211. [PubMed: 25038555]

[15]. Ho Joyce C, Ghosh Joydeep, and Sun Jimeng. 2014 Marble: high-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining ACM, 115–124.

[16]. Huang Kejun, Sidiropoulos Nicholas D, and Liavas Athanasios P. 2016 A flexible and efficient algorithmic framework for constrained matrix and tensor factorization. IEEE Transactions on Signal Processing 64, 19 (2016), 5052–5065.

[17]. Kiers Henk AL, Ten Berge Jos MF, and Bro Rasmus. 1999 PARAFAC2-Part I. A direct fitting algorithm for the PARAFAC2 model. Journal of Chemometrics 13, 3–4 (1999), 275–294.

[18]. Lin Yu-Ru, Sun Jimeng, Castro Paul C., Konuru Ravi B., Sundaram Hari, and Kelliher Aisling. 2009 MetaFac: community discovery via relational hypergraph factorization. In KDD.

[19]. Matsubara Yasuko, Sakurai Yasushi, van Panhuis Willem G, and Faloutsos Christos. 2014 FUNNEL: automatic mining of spatially coevolving epidemics. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining ACM, 105–114.

[20]. Parikh Neal, Boyd Stephen, and et al. 2014 Proximal algorithms. Foundations and Trends® in Optimization 1, 3 (2014), 127–239.

[21]. Perros Ioakeim, Evangelos E Papalexakis, Park Haesun, Vuduc Richard, Yan Xiaowei, Defilippi Christopher, Stewart Walter F, and Sun Jimeng. 2018 SUSTain: Scalable Unsupervised Scoring for Tensors and its Application to Phenotyping. arXiv preprint arXiv:1803.05473 (2018).

[22]. Perros Ioakeim, Evangelos E Papalexakis, Wang Fei, Vuduc Richard, Searles Elizabeth, Thompson Michael, and Sun Jimeng. 2017 SPARTan: Scalable PARAFAC2 for Large & Sparse Data. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17) ACM, 375–384.

[23]. Ramsay James O. 1988 Monotone regression splines in action. Statistical science (1988), 425–441.

[24]. Schönemann Peter H. 1966 A generalized solution of the orthogonal procrustes problem. Psychometrika 31, 1 (1966), 1–10.

[25]. Slee Vergil N. 1978 The International classification of diseases: ninth revision (ICD-9). Annals of internal medicine 88, 3 (1978), 424–426. [PubMed: 629506]

[26]. Timmerman Marieke E and Kiers Henk AL. 2002 Three-way component analysis with smoothness constraints. Computational statistics & data analysis 40, 3 (2002), 447–470.

[27]. Wang Yichen, Chen Robert, Ghosh Joydeep, Denny Joshua C, Kho Abel, Chen You, Malin Bradley A, and Sun Jimeng. 2015 Rubik: Knowledge guided tensor factorization and completion for health data analytics. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining ACM, 1265–1274.

[28]. Wise Barry M, Gallagher Neal B, and Martin Elaine B. 2001 Application of PARAFAC2 to fault detection and diagnosis in semiconductor etch. Journal of chemometrics 15, 4 (2001), 285–298.

CCS CONCEPTS
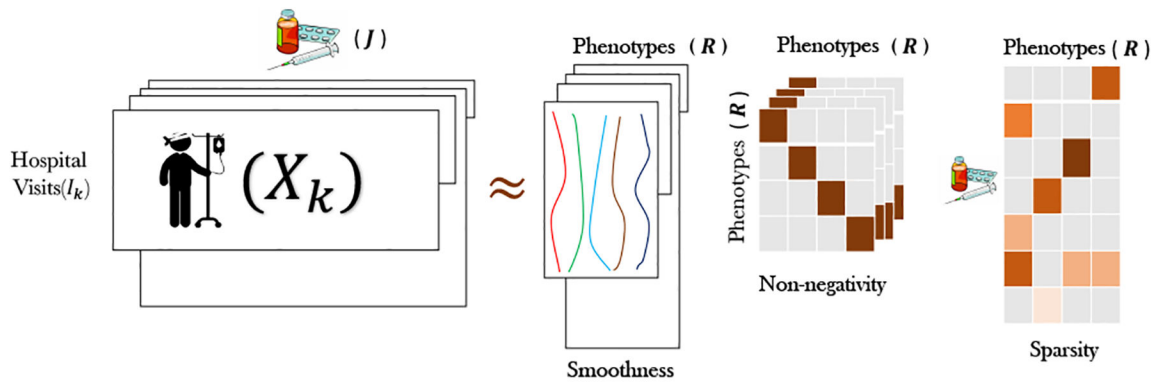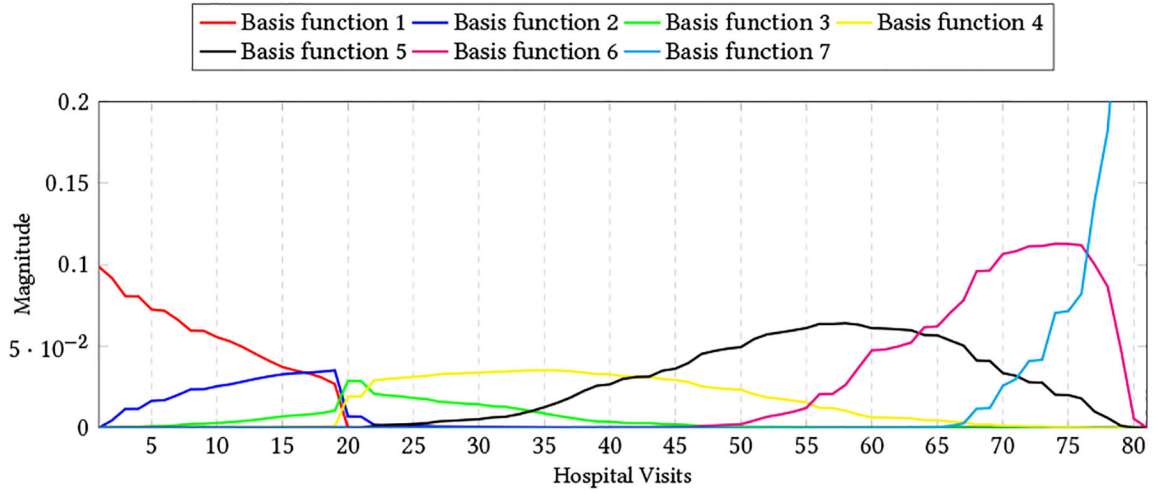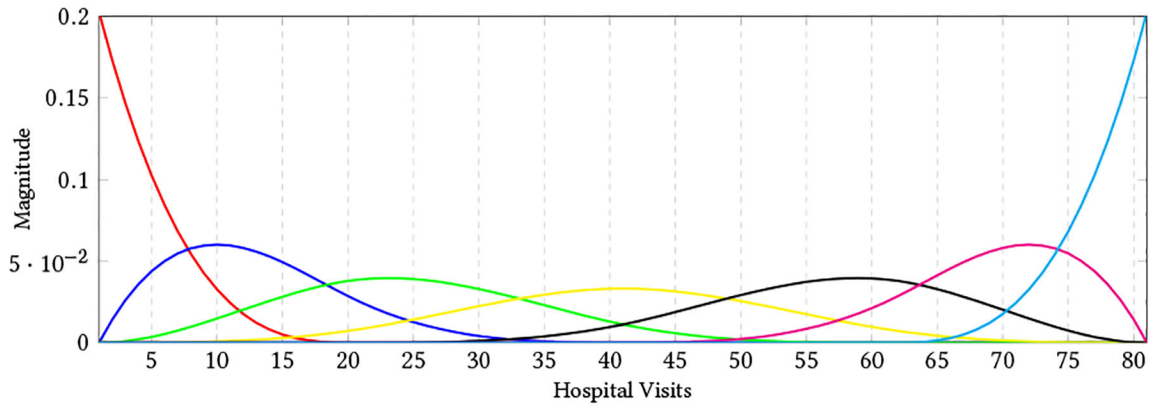
- **Information systems** → Data mining;

Figure 1:
An illustration of the constraints imposed by COPA on PARAFAC2 model factors, targeting temporal phenotyping via EHR data.

(a) *Basis functions used by* COPA.

(b) *Basis functions used by* Helwig.

Figure 2:
7 Basis functions for a patient with sickle cell anemia. Figure 2a shows the basis functions that COPA used for incorporating the smoothness that considers the gap between two visits while figure 2b related to basis functions for Helwig which divide the range [0,80] based on a equal distance.
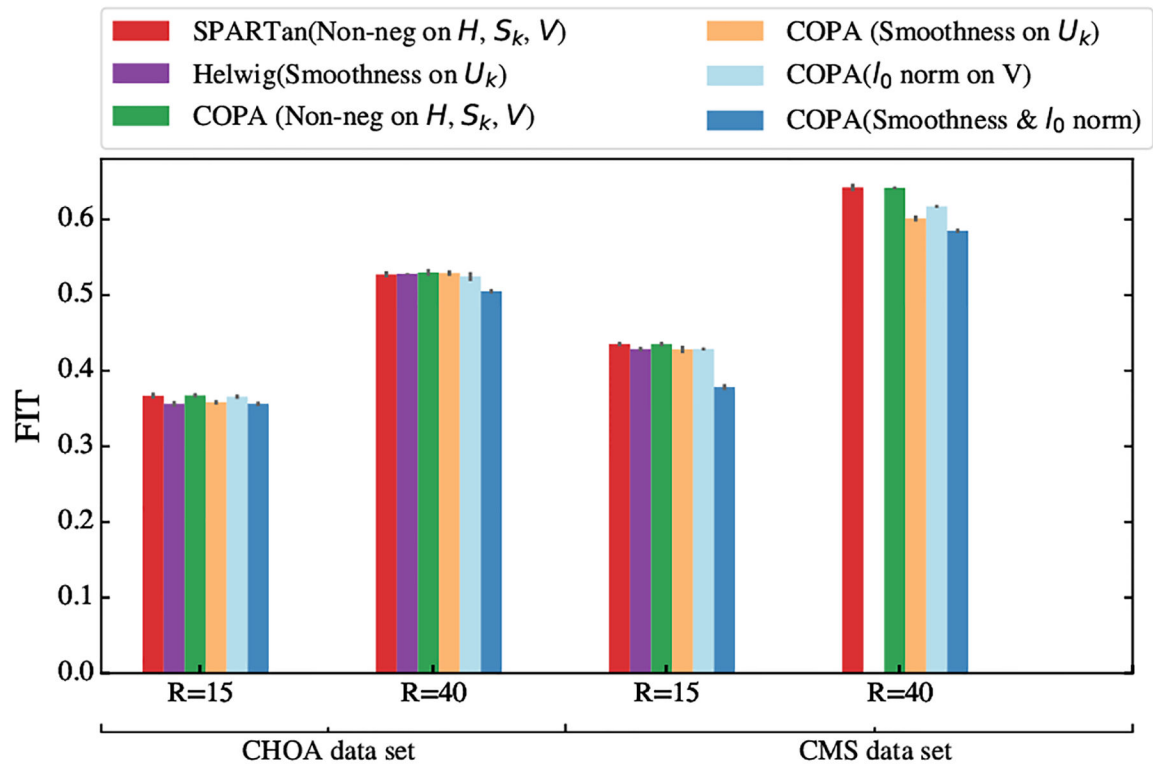
Figure 3:
Comparison of FIT for different approaches with various constraints on two target ranks R = 15 and R = 40 on real world datasets. Overall, COPA achieves comparable fit to SPARTan while supporting more constraints. The missing purple bar in the forth column is out of memory failure for Helwig method.
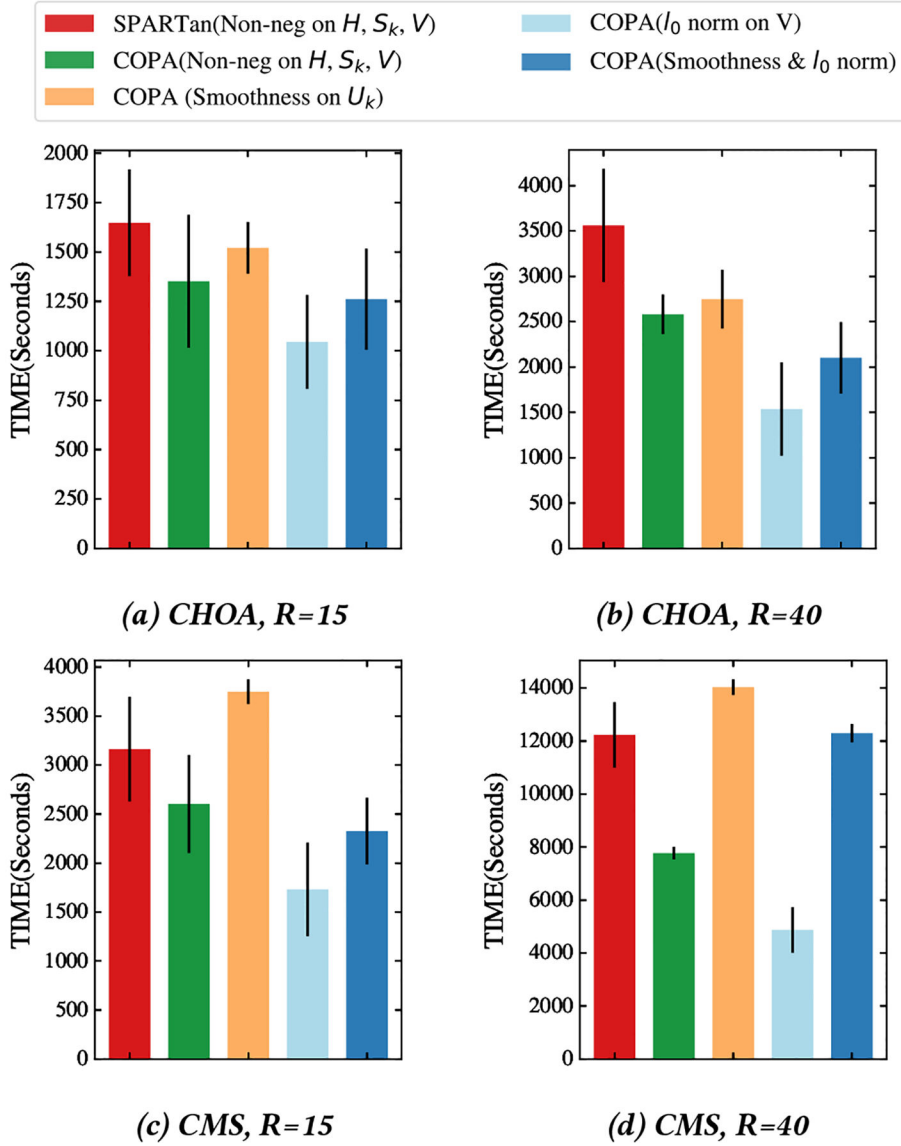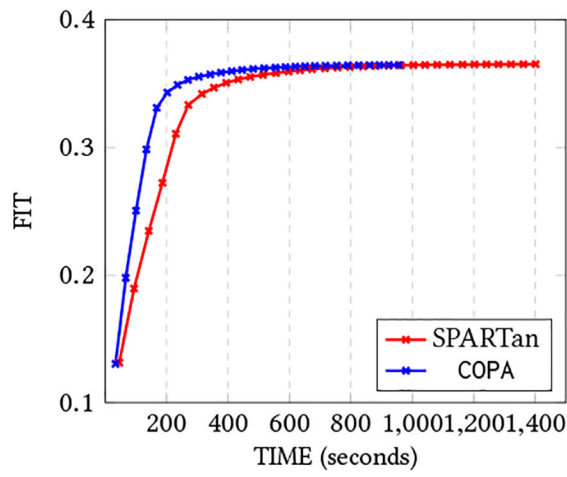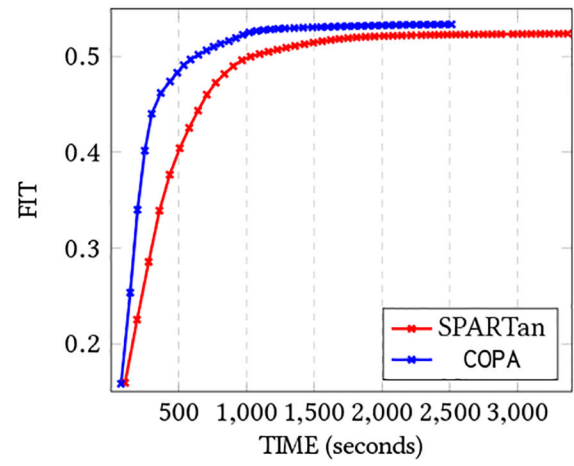
Figure 4:
The Total Running Time comparison (average and standard deviation) in seconds for different versions of COPA and SPARTan for 5 different random initializations. Note that even with smooth constraint COPA performs just slightly slower than SPARTan, which does not support such smooth constraints.
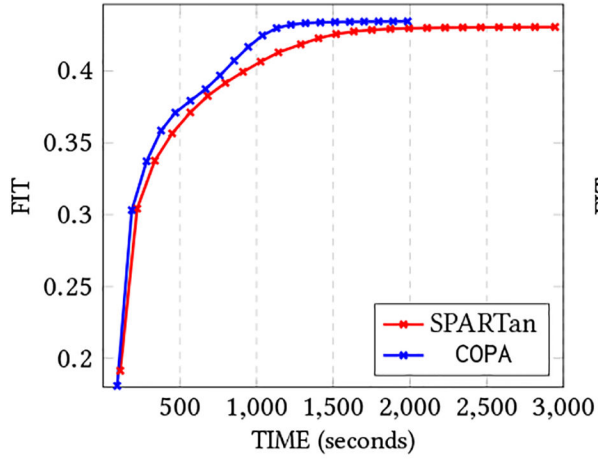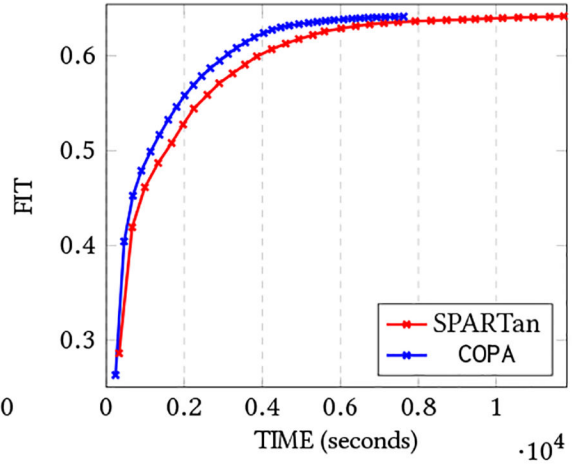
(a) 15 Components

(b) 40 Components

Figure 5:
The best Convergence of COPA and SPARTan out of 5 different random initializations with non-negativity constraint on H, { S_k }, and V on CHOA data set for different target ranks (two cases considered: R={15,40}).
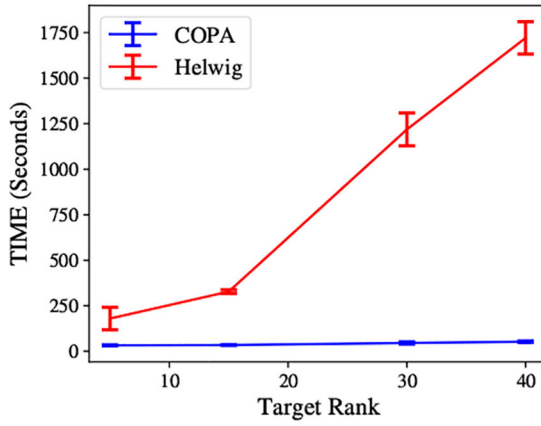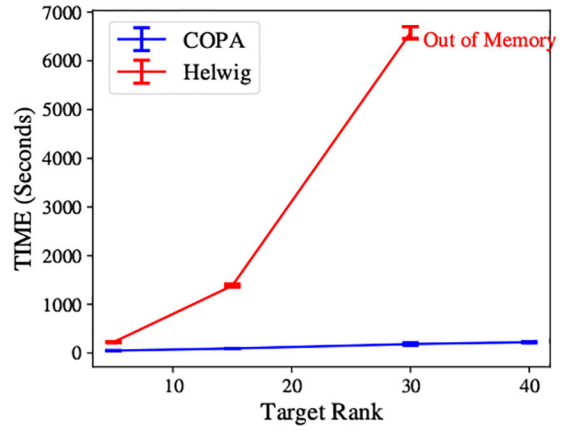
(a) 15 Components.   *(b) 40 Components.*

Figure 6:
The best convergence of COPA and SPARTan out of 5 different random initializations with non-negativity constraint on H, $\{S_k\}$, and V on CMS data with K=843,162, J=284 and maximum number of observations are 1500. Algorithms tested on different target ranks (two cases considered: R={15,40}).

(a) CHOA data set

(b) CMS data set.

Figure 7:
Time in seconds for one iteration (as an average of 5 different random initializations) for different values of R. The left figure is the comparison on CHOA and the right figure shows the comparison on CMS. For R=40 COPA achieves 32× over the Helwig approach on CHOA while for CMS dataset, execution in Helwig failed due to the excessive amount of memory request and COPA finished an iteration with the average of 224.21 seconds.
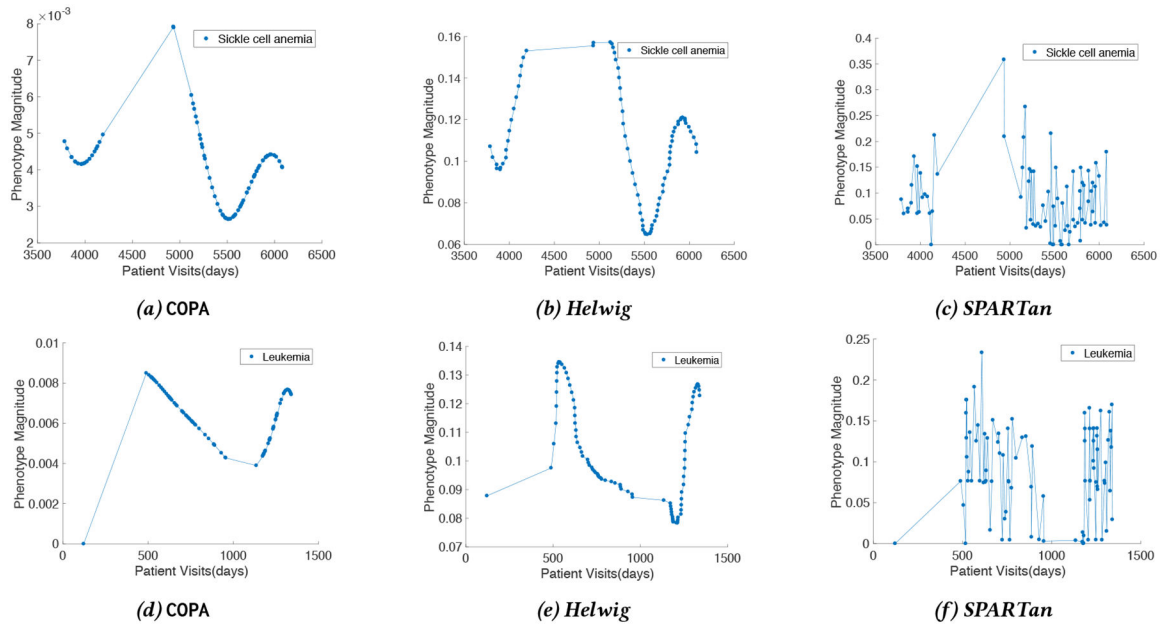
Figure 8:
The temporal patterns extracted for two patients by COPA, Helwig, and SPARTan. The first row is associated with a patient who has sickle cell anemia while the second row is for a patient with Leukemia.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Table 1:

Comparison of PARAFAC2 models and constrained tensor factorization applied to phenotyping

Property	Marble [15]	Rubik [27]	PARAFAC2 [6]	SPARTan [22]	Helwig [12]	COPA
Smoothness	-	-	-	-	✓	✓
Sparsity	✓	✓	-	-	-	✓
Scalability	-	-	-	✓	-	✓
Handle irregular tensors	-	-	✓	✓	✓	✓

Table 2:

Symbols and notations used throughout the paper.

Symbol	Definition
$*$	Element-wise Multiplication
$\odot$	Khatri Rao Product
$\circ$	Outer Product
$c(Y)$	A constraint on factor matrix Y
$\bar{Y}$	Auxiliary variable for factor matrix Y
$\mathcal{Y}, Y, y$	Tensor, matrix, vector
$Y_{(n)}$	Mode-n Matricization of $\mathcal{Y}$
$Y(i, :)$	Spans the entire i-th row of Y
X_k	k^{th} frontal slice of tensor $\mathcal{X}$
$diag(y)$	Diagonal matrix with vector y on diagonal
$diag(Y)$	Extract the diagonal of matrix Y

Table 3:

Summary statistics of real datasets that we used in the experiments. K denotes the number of patients, J is the number of medical features and I_k denotes the number of clinical visits for k^{th} patient.

Dataset	K	J	$max(I_k)$	#non-zero elements
CHOA	247.885	1388	857	11 Million
CMS	843,162	284	1500	84 Million

Table 4:

Values of parameters (l, μ) for different data sets and various target ranks for COPA.

Algorithm	CHOA		CMS	
	R=15	R=40	R=15	R=40
# basis functions (*l*)	33	81	106	253
μ	23	25	8	9

Table 5:

The average and standard deviation of sparsity metric (fraction of zero elements divided by the matrix size) comparison for the factor matrix V on CHOA and CMS using two different target ranks for 5 different random initializations.

	CHOA		CMS	
Algorithm	**R=15**	**R=40**	**R=15**	**R=40**
COPA	**0.9886±0.0035**	**0.9897±0.0027**	**0.9950±0.0001**	**0.9963±0.0002**
SPARTan [22]	0.7127 ±0.0161	0.8127±0.0029	0.1028±0.0032	0.2164±0.0236

Table 6:

Speedups (running time of SPARTan divided by running time of COPA for various constraint configurations) and corresponding p-values. COPA is faster (up to 2.5×) on the majority of constraint configurations as compared to the baseline SPARTan approach which can only handle non-negativity constraints.

		Non-neg COPA	Smooth COPA	Sparse COPA	Smooth & Sparse COPA
CHOA, R=15	Speed up	**1.21**	**1.08**	**1.57**	**1.31**
	p-value	**0.163**	**0.371**	**0.005**	**0.048**
CHOA, R=40	Speed up	**1.38**	**1.29**	**2.31**	**1.69**
	p-value	**0.01**	**0.032**	**0.0005**	**0.002**
CMS, R=15	Speed up	**1.21**	0.84	**1.82**	**1.36**
	p-value	**0.125**	1.956	**0.002**	**0.018**
CMS, R=40	Speed up	**1.57**	0.87	**2.51**	0.99
	p-value	**0.00005**	1.986	**0.000004**	1.08

Table 7:

Phenotypes discovered by COPA. The red color corresponds to diagnosis and blue color corresponds to medication. The meaningfulness of phenotypes endorsed by a medical expert. No additional post-processing was performed on these results.

Leukemias
Leukemias
Immunity disorders
Deficiency and other anemia
HEPARIN AND RELATED PREPARATIONS
Maintenance chemotherapy; radiotherapy
ANTIEMETIC/ANTIVERTIGO AGENTS
SODIUM/SALINE PREPARATIONS
TOPICAL LOCAL ANESTHETICS
GENERAL ANESTHETICS INJECTABLE
ANTINEOPLASTIC - ANTIMETABOLITES
ANTIHISTAMINES - 1ST GENERATION
ANALGESIC/ANTIPYRETICS NON-SALICYLATE
ANALGESICS NARCOTIC ANESTHETIC ADJUNCT AGENTS
ABSORBABLE SULFONAMIDE ANTIBACTERIAL AGENTS
GLUCOCORTICOIDS
Neurological Disorders
Other nervous system disorders
Epilepsy; convulsions
Paralysis
Other connective tissue disease
Developmental disorders
Rehabilitation care; and adjustment of devices
ANTICONVULSANTS
Congenital anomalies
Other perinatal conditions
Cardiac and circulatory congenital anomalies
Short gestation; low birth weight
Other congenital anomalies
Fluid and electrolyte disorders
LOOP DIURETICS
IV FAT EMULSIONS
Sickle Cell Anemia
Sickle cell anemia
Other gastrointestinal disorders
Other nutritional; endocrine; and metabolic disorders
Other lower respiratory disease
Asthma

Allergic reactions

Esophageal disorders

Respiratory failure; insufficiency; arrest (adult)

Other upper respiratory disease

BETA-ADRENERGIC AGENTS

ANALGESICS NARCOTICS

NSAIDS, CYCLOOXYGENASE INHIBITOR - TYPE

ANALGESIC/ANTIPYRETICS NON-SALICYLATE

POTASSIUM REPLACEMENT

SODIUM/SALINE PREPARATIONS

GENERAL INHALATION AGENTS

LAXATIVES AND CATHARTICS

IV SOLUTIONS: DEXTROSE-SALINE

ANTIEMETIC/ANTIVERTIGO AGENTS

SEDATIVE-HYPNOTICS NON-BARBITURATE

GLUCOCORTICOIDS, ORALLY INHALED

FOLIC ACID PREPARATIONS

ANALGESICS NARCOTIC ANESTHETIC ADJUNCT AGENTS