

dCMF: Learning interpretable evolving patterns from temporal multiway data

Christos Chatzis*, Carla Schenker[†], J       E. Cohen^{  } and Evrim Acar^{  }

*Simula Metropolitan Center for Digital Engineering and

Faculty of Art, Design and Technology, Oslo Metropolitan University, Oslo, Norway, christos@simula.no

^{  }Simula Metropolitan Center for Digital Engineering, Oslo, Norway, carla@simula.no

^{  }Univ Lyon, INSA-Lyon, UCBL, UJM, CNRS, Lyon, France, jeremy.cohen@cnrs.fr

^{  }Simula Metropolitan Center for Digital Engineering, Oslo, Norway, evrim@simula.no

Abstract—Multiway datasets are commonly analyzed using unsupervised matrix and tensor factorization methods to reveal underlying patterns. Frequently, such datasets include timestamps and could correspond to, for example, health-related measurements of subjects collected over time. The temporal dimension is inherently different from the other dimensions, requiring methods that account for its intrinsic properties. Linear Dynamical Systems (LDS) are specifically designed to capture sequential dependencies in the observed data. In this work, we bridge the gap between tensor factorizations and dynamical modeling by exploring the relationship between LDS, Coupled Matrix Factorizations (CMF) and the PARAFAC2 model. We propose a time-aware coupled factorization model called d(ynamical)CMF that constrains the temporal evolution of the latent factors to adhere to a specific LDS structure. Using synthetic datasets, we compare the performance of dCMF with PARAFAC2 and t(emporal)PARAFAC2 which incorporates temporal smoothness. Our results show that dCMF and PARAFAC2-based approaches perform similarly when capturing smoothly evolving patterns that adhere to the PARAFAC2 structure. However, dCMF outperforms alternatives when the patterns deviate from the PARAFAC2 structure. Furthermore, we demonstrate that the proposed dCMF method enables to capture more complex dynamics when additional prior information about the temporal evolution is incorporated.

Index Terms—tensor factorizations, PARAFAC2, time-aware methods, dynamical systems, coupled factorizations

I. INTRODUCTION

With the rapid growth in the availability of measurements in recent years, many datasets now inherently exhibit multiway structure, allowing for variations along multiple dimensions. The analysis of such data, a challenge that arises in numerous domains, has been effectively tackled using tensor factorizations (e.g. [1], [2]), i.e., extensions of matrix factorizations to multiway arrays. In their essence, these methods often enable the decomposition of the data into interpretable factors that reveal the underlying patterns.

Frequently, one of the “ways” that the data evolves across is time, implying that patterns can change over time. Consequently, the objective of such analyses often becomes the identification and tracking of these dynamics. Standard methods can be used off-the-shelf to analyze such temporal data. For example, the CANDECOMP/PARAFAC (CP) tensor model [3] can be used to extract a temporal factor that captures the evolution of pattern strength over time. However, the under-

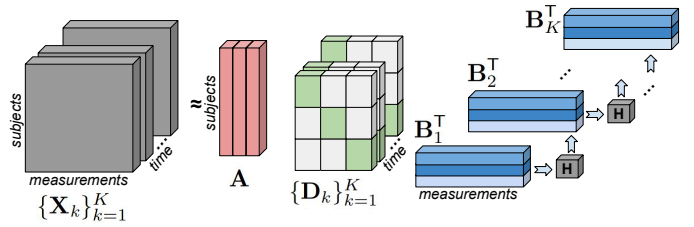


Fig. 1: The dCMF model. Evolving patterns captured by the columns of factor matrix \mathbf{B}_k are related to the previous \mathbf{B}_{k-1} through the transition matrix \mathbf{H} .

lying structure remains fixed. PARAFAC2 [4] addresses this limitation by allowing factors that capture structural changes, though specific structural constraints must hold. Coupled Matrix Factorization (CMF) methods can also be used for this task and while possessing less structural constraints, they generally lack uniqueness guarantees. Nevertheless, none of the aforementioned approaches account for the inherent sequential nature of the time dimension. Taking the temporal aspect into account offers several advantages, including improved user recommendations [5], more accurate future predictions [6], and robustness to noise [7], [8]. As a result, there are numerous approaches incorporating time in the literature. A prevalent strategy is temporal regularization. Some approaches impose such regularization on the strength of the uncovered patterns over time: Temporal Regularized Matrix Factorization (TRMF) [9] assumes an autoregressive structure and ATOM [10] assumes smoothness. Others focus on the structure of latent factors such as Sun et al. [11] integrating Kalman filter-based temporal regularization. Similarly, Appel et al. [12] assume smoothness on the evolution (of latent structures) of a coupled CMF model while t(emporal)PARAFAC2 [7] promotes smoothly changing factors. Apart from temporal regularization, other approaches include enforcing smoothness through small gradient steps on factor updates when new data is received [13] and modeling one of the factors of a CMF as a function of time [14]. There are also other approaches that fall under incremental/online methods (e.g. [13]), where a decomposition is updated as new data arrives. However, such methods often update factor matrices in all modes rather than capturing the evolving patterns in one mode; therefore, serving

a different purpose.

Another approach to studying evolving patterns are Linear Dynamical Systems (LDS). For instance, many studies in hyperspectral imaging utilize LDS and learn evolving structures from raw data [15], [16]. Closer to our work is the g(eneralized)LDS framework [6], which receives as input (a matrix formed from) multiple multivariate time series and factorizes the input into a common “emission” matrix and a factor corresponding to hidden states of an LDS. Computing the transition matrix of this system is part of the optimization procedure. While our work does not currently involve learning the transition matrix, it extends gLDS by introducing support for tensor inputs and enabling latent factors to evolve independently and under different dynamical systems.

Analyzing temporal multiway data with the goal of capturing interpretable evolving patterns requires methods with specific properties. First, **time-awareness** is essential, as the approach must explicitly account for the temporal dimension. Second, the method should allow **sufficient structural flexibility** in the latent factors. Imposing overly rigid constraints can lead to inaccurate factor estimates. Third, the **uniqueness** of the factorization is imperative for interpretability, ensuring that the extracted factors correspond to identifiable components. To the best of our knowledge, there is a lack of methods fulfilling these properties.

In this work, we propose d(ynamical)CMF, a coupled factorization approach that involves temporal regularization based on LDS (Figure 1). The framework enables incorporating prior information on the factors by choosing a-priori the transition matrix \mathbf{H} for the LDS. Our contributions are as follows:

- We introduce the dCMF model and demonstrate that incorporating the proposed LDS penalty results in a time-aware coupled factorization. We also explore the relation between dCMF and CP, PARAFAC2 models as well as temporal smoothness.
- With experiments on synthetic data, we demonstrate that dCMF is accurate and flexible since it can capture the evolving patterns that PARAFAC2-based methods would not. Further, the uniqueness of the solution is empirically discussed in different experimental settings.
- We also demonstrate that if accurate prior information is incorporated in the dCMF model, high accuracy can be expected even in settings with high noise levels.

II. BACKGROUND

Let $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ denote a third-order tensor, where the last mode (K -sized) indexes the time points as in Figure 1. Such data can be analyzed by a coupled matrix factorization approach [17] with each frontal slice $\mathbf{X}_k \in \mathbb{R}^{I \times J}$ corresponding to k -th time point modelled as:

$$\mathbf{X}_k \approx \mathbf{A}\mathbf{B}_k^\top \quad \forall k \leq K \quad (1)$$

where $\mathbf{A} \in \mathbb{R}^{I \times R}$ and $\mathbf{B}_k \in \mathbb{R}^{J \times R} \quad \forall k \leq K$ with R denoting the number of components. Interpreting the columns of the factors yields a summary of the main patterns present, which can change across k . However, there are no uniqueness guarantees

when solving for the factorization unless additional constraints (e.g. non-negativity) are imposed. The PARAFAC2 model [4], which can be considered a CMF approach, utilizes an additional factor $\mathbf{C} \in \mathbb{R}^{K \times R}$ with $\mathbf{D}_k = \text{diag}(\mathbf{C}_{k,:}) \quad \forall k \leq K$, i.e., a diagonal matrix with the k -th row of \mathbf{C} on the diagonal, and is formulated as:

$$\begin{aligned} \mathbf{X}_k &\approx \mathbf{A}\mathbf{D}_k\mathbf{B}_k^\top \\ \mathbf{B}_k &= \mathbf{P}_k\mathbf{B} \end{aligned} \quad \forall k \leq K \quad (2)$$

where $\mathbf{A} \in \mathbb{R}^{I \times R}$ and $\mathbf{B} \in \mathbb{R}^{R \times R}$. Here, $\mathbf{P}_k \in \mathbb{R}^{J \times R} \quad \forall k \leq K$ have orthonormal columns. The second line is necessary for uniqueness of the patterns (up to permutation and scaling ambiguities) [18] and referred to as the PARAFAC2 constraint. Roald et al. [19] demonstrated the model’s ability to capture evolving patterns. Nevertheless, when analyzing temporal data, this method suffers from two drawbacks: (a) it does not take into account the sequential nature of the temporal dimension, and (b) the uncovered latent factors must adhere to the PARAFAC2 constraint. To tackle (a), t(emporal)PARAFAC2 [7], [8] was proposed based on the following optimization problem:

$$\begin{aligned} \min_{\mathbf{A}, \{\mathbf{B}_k\}_{k=1}^K, \mathbf{C}} \quad & \left\{ \sum_{k=1}^K \left\| \mathbf{X}_k - \mathbf{A}\mathbf{D}_k\mathbf{B}_k^\top \right\|_F^2 + g_{\mathbf{A}}(\mathbf{A}) \right. \\ & \left. + \lambda_B \sum_{k=1}^{K-1} \left\| \mathbf{B}_{k+1} - \mathbf{B}_k \right\|_F^2 + g_{\mathbf{D}}(\{\mathbf{D}_k\}_{k=1}^K) \right\} \\ \text{subject to} \quad & \mathbf{B}_k = \mathbf{P}_k\mathbf{B} \quad \forall k \leq K \end{aligned} \quad (3)$$

where smooth changes across the evolving factors \mathbf{B}_k are assumed and $g_{\mathbf{A}}$ and $g_{\mathbf{D}}$ indicate norm-based penalties on the respective factors (necessary due to scaling ambiguity). However, issue (b) persists, potentially distorting the uncovered latent structures when data does not follow the PARAFAC2 constraint. tPARAFAC2 also struggles to capture more complex temporal dynamics beyond smooth changes.

III. PROPOSED METHOD: D(YNAMICAL)CMF

To address these points, we propose the d(ynamical)CMF model, which is a CMF-based approach possessing temporal regularization based on LDS instead of the PARAFAC2 constraint. We discuss how the proposed approach is related to CP, PARAFAC2 and tPARAFAC2 here, and compare them using numerical experiments in Section IV. In Section IV-E, we discuss the uniqueness of the dCMF solution empirically.

A. The LDS constraint

In dCMF, we regularize each component of the evolving factors to adhere to the structure of an LDS, i.e.

$$\begin{aligned} \mathbf{X}_k &\approx \mathbf{A}\mathbf{D}_k\mathbf{B}_k^\top \quad \forall k \leq K \\ \mathbf{b}_{k+1,r} &\approx \mathbf{H}^{(r)}\mathbf{b}_{k,r} \quad \forall k < K, \quad r \leq R \end{aligned} \quad (5)$$

with $\mathbf{b}_{k,r} = \mathbf{B}_k(:, r)$ and $\{\mathbf{H}^{(r)}\}_{r=1}^R$ corresponding to the component-specific transition matrices of the systems.

If $\mathbf{H}^{(1)} = \mathbf{H}^{(2)} = \dots = \mathbf{H} = \mathbf{I}$ and the relation between temporal factors holds exactly, Equation (5) corresponds to the CP model since $\mathbf{B}_1 = \mathbf{B}_2 = \dots = \mathbf{B}_K = \mathbf{B}$. If, more generally, we assume that all transition matrices are equal to

an orthogonal matrix \mathbf{H} , then this adheres to the PARAFAC2 constraint, which facilitates uniqueness. A specific instance of this would be setting $\mathbf{H} = \mathbf{I}$, where temporal smoothness similar to (3) is assumed. Furthermore, even if we relax the orthogonality requirement for \mathbf{H} , we still obtain a time-aware factorization of the input, i.e., in PARAFAC2 the factors that correspond to the k -th and l -th time-point (w.l.o.g. $k < l$) are $\mathbf{B}_l = \mathbf{P}_l \mathbf{B}$ and $\mathbf{B}_k = \mathbf{P}_k \mathbf{B}$ and related through $\mathbf{B}_l = \mathbf{P}_l \mathbf{P}_k^\top \mathbf{B}_k$ irrespective of their order in time whereas using the LDS constraint, the relation becomes $\mathbf{B}_l = \mathbf{H}^{l-k} \mathbf{B}_k$, which takes into account the time points of the observations. We restrict our analysis on LDS regularization using a single, known a-priori, potentially non-orthogonal, transition matrix as follows:

$$\begin{aligned} \mathbf{X}_k &\approx \mathbf{A} \mathbf{D}_k \mathbf{B}_k^\top \quad \forall k \leq K \\ \mathbf{b}_{k+1,r} &\approx \mathbf{H} \mathbf{b}_{k,r} \quad \forall k < K, \quad r \leq R. \end{aligned} \quad (6)$$

B. Optimization

We formulate (6) as the following optimization problem:

$$\begin{aligned} \min_{\mathbf{A}, \{\mathbf{B}_k\}_{k=1}^K, \mathbf{C}} \quad & \left\{ \sum_{k=1}^K \left\| \mathbf{X}_k - \mathbf{A} \mathbf{D}_k \mathbf{B}_k^\top \right\|_F^2 + \lambda \left\| \mathbf{A} \right\|_F^2 \right. \\ & \left. + \lambda_B \sum_{k=1}^{K-1} \left\| \mathbf{B}_{k+1} - \mathbf{H} \mathbf{B}_k \right\|_F^2 + \sum_{k=1}^K \left(\lambda \left\| \mathbf{D}_k \right\|_F^2 + \ell_{\mathbb{R}_+}(\mathbf{D}_k) \right) \right\} \end{aligned} \quad (7)$$

where the first and third terms denote the fidelity term and proposed LDS constraint, respectively. $\ell_{\mathbb{R}_+}$ denotes the indicator/characteristic function of the non-negative orthant. Due to the scaling ambiguity, it is necessary to impose ridge regularization on factors \mathbf{A} and $\{\mathbf{D}_k\}_{k=1}^K$ [20], but other types of norm-based regularization are applicable as well. The LDS constraint is included as a soft regularization penalty, which gives the method more structural freedom (see Section IV-C). We overcome the sign ambiguity by imposing non-negativity on $\{\mathbf{D}_k\}_{k=1}^K$.

We solve this optimization problem using an Alternating Optimization - Alternating Direction Method of Multipliers (AO-ADMM)-based algorithm [20], [21]. This algorithmic framework is highly flexible, allowing for the straightforward inclusion of various types of regularization in all modes to accommodate different application requirements. We form the augmented Lagrangian of (7) as follows:

$$\begin{aligned} \mathcal{L} = & \sum_{k=1}^K \left\| \mathbf{X}_k - \mathbf{A} \mathbf{D}_k \mathbf{B}_k^\top \right\|_F^2 + \lambda \left\| \mathbf{A} \right\|_F^2 + \lambda \sum_{k=1}^K \left\| \mathbf{D}_k \right\|_F^2 \\ & + \lambda_B \sum_{k=2}^K \left\| \mathbf{Z}_{\mathbf{B}_k} - \mathbf{H} \mathbf{Z}_{\mathbf{B}_{k-1}} \right\|_F^2 + \sum_{k=1}^K \frac{\rho_{\mathbf{B}_k}}{2} \left\| \mathbf{B}_k - \mathbf{Z}_{\mathbf{B}_k} + \boldsymbol{\mu}_{\mathbf{B}_k} \right\|_F^2 \\ & + \sum_{k=1}^K \ell_{\mathbb{R}_+}(\mathbf{Z}_{\mathbf{D}_k}) + \sum_{k=1}^K \frac{\rho_{\mathbf{D}_k}}{2} \left\| \mathbf{D}_k - \mathbf{Z}_{\mathbf{D}_k} + \boldsymbol{\mu}_{\mathbf{D}_k} \right\|_F^2, \end{aligned} \quad (8)$$

where the auxiliary variables $\{\mathbf{Z}_{\mathbf{B}_k}, \mathbf{Z}_{\mathbf{D}_k}\}_{k=1}^K$, dual variables $\{\boldsymbol{\mu}_{\mathbf{B}_k}, \boldsymbol{\mu}_{\mathbf{D}_k}\}_{k=1}^K$ and feasibility penalties $\{\rho_{\mathbf{B}_k}, \rho_{\mathbf{D}_k}\}_{k=1}^K$ have been introduced. We then iteratively solve the subproblems for each mode until convergence is achieved (small relative or absolute change in function value of (7)) or a pre-set maximum number of iterations is reached. Our algorithm follows [20], with the addition of the update of $\{\mathbf{Z}_{\mathbf{B}_k}\}_{k=1}^K$, for which a block diagonal system has to be solved.

IV. EXPERIMENTAL EVALUATION

A. Experimental setup

We evaluate the performance of the proposed method using synthetic datasets, each with dimensions $30 \times 40 \times 35$. To assess the accuracy of the method in terms of finding the underlying patterns under different scenarios, we generate four types of synthetic data, with details provided in the subsequent subsections. In all cases, we generate the ground truth factors $\mathbf{A}, \{\mathbf{B}_k\}_{k=1}^K, \mathbf{C}$ with $R = 3$ components, form each slice of the input data $\mathbf{X}_k = \mathbf{A} \mathbf{D}_k \mathbf{B}_k^\top \quad \forall k \leq K$ and add noise according to

$$\mathbf{X}_{noisy} = \mathbf{X} + \eta \left\| \mathbf{X} \right\|_F \frac{\boldsymbol{\mathcal{E}}}{\left\| \boldsymbol{\mathcal{E}} \right\|_F}$$

where $\boldsymbol{\mathcal{E}} \sim \mathcal{N}(0, 1)$ with $\eta = 0.25, 0.75, 1.25$ denoting the noise level. The goal is to recover the underlying factors from noisy data. We assess accuracy using the Factor Match Score (FMS), taking values between 0 and 1, with 1 indicating perfect match.

$$\text{FMS} = \frac{1}{R} \sum_{i=1}^R \frac{|\hat{\mathbf{a}}_i^\top \mathbf{a}_i|}{\|\hat{\mathbf{a}}_i\| \|\mathbf{a}_i\|} \frac{|\hat{\mathbf{b}}_i^\top \mathbf{b}_i|}{\|\hat{\mathbf{b}}_i\| \|\mathbf{b}_i\|} \frac{|\hat{\mathbf{c}}_i^\top \mathbf{c}_i|}{\|\hat{\mathbf{c}}_i\| \|\mathbf{c}_i\|},$$

where $\mathbf{a}_i, \mathbf{b}_i$, and \mathbf{c}_i represent the i -th column of the true factors, and $\hat{\mathbf{a}}_i, \hat{\mathbf{b}}_i$, and $\hat{\mathbf{c}}_i$ are the factors estimated by the model (after finding the optimal permutation). $\hat{\mathbf{b}}_i$ and \mathbf{b}_i are formed by stacking the i -th column across all $\{\mathbf{B}_k\}_{k=1}^K$.

Our implementation¹ builds on [22]–[24]. AO-ADMM is used for all methods. For all experiments, the maximum number of iterations is set to 8000, the outer relative and absolute tolerance to 10^{-8} and the inner relative tolerance to 10^{-5} . The feasibility gaps (i.e., the distance between each variable and the respective auxiliary/ies) are required to be less than 10^{-5} . Twenty random initializations are used for each dataset and all runs that result in degenerate solutions [1] are discarded. Out of the remaining runs for each dataset, we designate as ‘best’ the run with the lowest function value.

B. Smooth PARAFAC2 data

In this experiment, we compare the performance of PARAFAC2, tPARAFAC2, and the proposed dCMF on smooth data exhibiting PARAFAC2 structure. For dCMF, we set $\mathbf{H} = \mathbf{I}$, since the data in this experiment is smoothly changing. We generate 30 datasets, where the ground truth factors are drawn as $\mathbf{A} \sim \mathcal{N}(0, 1)$ and $\mathbf{C} \sim \mathcal{U}(1, 15)$. For the evolving mode, we designate specific indices as ‘active’ for each pattern, setting all remaining entries to zero to ensure orthogonality between patterns. We draw the active entries of \mathbf{B}_1 from $\mathcal{N}(0, 1)$ and incrementally add values from $\mathcal{N}(0, 0.25)$ to active indices of each pattern to obtain $\mathbf{B}_2, \mathbf{B}_3, \dots, \mathbf{B}_K$. We then column-normalize each \mathbf{B}_k . With this process, we make sure the ground truth has PARAFAC2 structure since (a) the off-diagonal entries of the cross product $\mathbf{B}_k^\top \mathbf{B}_k$ are constantly zero due to the patterns not overlapping and (b) the diagonal entries are constant since the evolving factors are column-normalized. Here, we expect PARAFAC2-based methods to perform well.

The results of this experiment are shown in the first column of Figure 2, where various regularization strengths λ_B are

¹<https://github.com/cchatzis/temp-dcmf-repo>

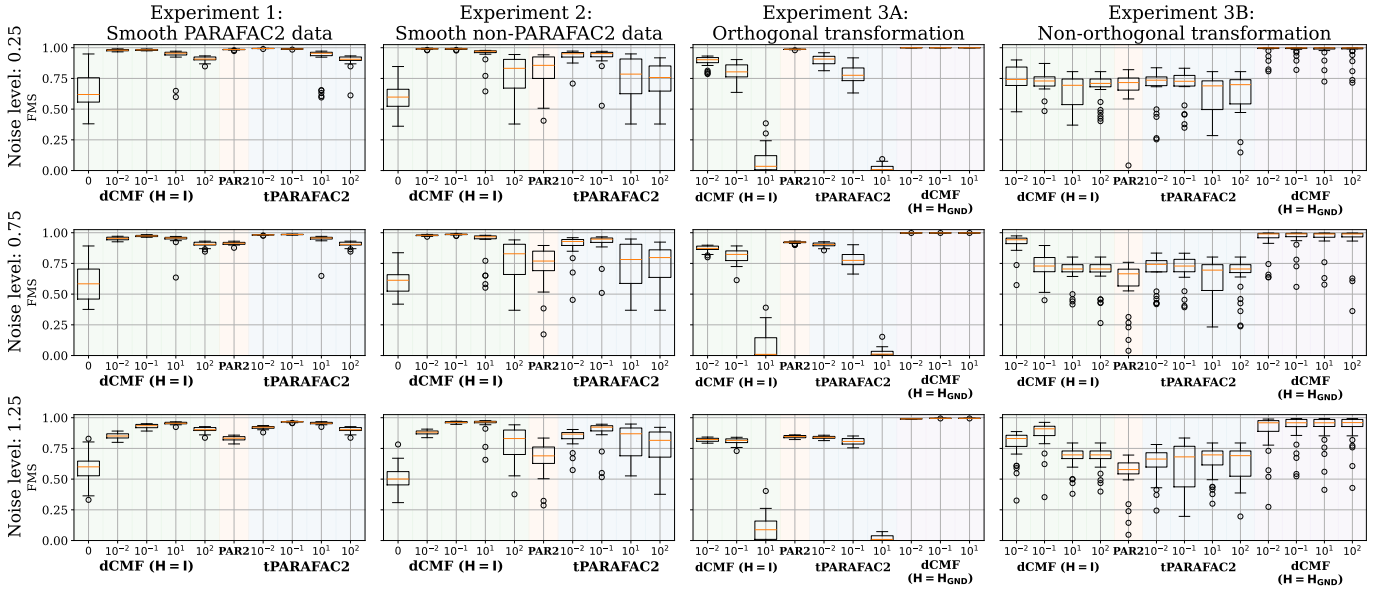


Fig. 2: Accuracy of methods in terms of FMS in each experiment. Each boxplot contains 30 points, one for the best run of each method for each dataset. Multiple regularization strengths are considered for temporal smoothness (tPARAFAC2) and the proposed LDS constraint (dCMF). PAR2 is short for PARAFAC2. \circ symbols are used to indicate outliers of the boxplots.

considered. Setting $\lambda_B = 0$ for dCMF results in an unconstrained CMF model (not considering ridge penalties), yielding non-unique output. As the assumption that the data is smooth holds, increasing λ_B results in more accurate recovery for both dCMF and tPARAFAC2, especially for higher noise levels, with tPARAFAC2 having slightly higher accuracy.

C. Non-PARAFAC2 smooth data

This setting investigates the performance of PARAFAC2, tPARAFAC2, and dCMF (with $\mathbf{H} = \mathbf{I}$) on synthetic data that significantly violates the PARAFAC2 constraint. 30 datasets are randomly generated with $\mathbf{A} \sim \mathcal{N}(0, 1)$ and $\mathbf{C} \sim \mathcal{U}(1, 15)$. According to the PARAFAC2 constraint, $\mathbf{B}_k^T \mathbf{B}_k$ has to remain constant $\forall k \leq K$. For a single time point k , off-diagonal entries of the cross product matrix capture the cosine similarity of evolving patterns (i.e., $\mathbf{B}_k(:, i)$ and $\mathbf{B}_k(:, j)$) after column normalization which has to, according to the PARAFAC2 constraint, remain constant. To violate this constraint, we create datasets by changing the similarity between the components across time by introducing structured steps from one pattern towards (or away from) another. To instantiate, component 1 will be taking steps towards component 2 for each $k \in \mathcal{R}_1 = [m, n] \cap \mathbb{Z}, 1 \leq m \leq n \leq K$ and steps “away” for the rest of the time points:

$$\mathbf{B}_k \leftarrow \underbrace{\mathbf{B}_{k-1} + \mathcal{N}(0, 0.1)}_{\text{smooth changes}} + \Delta_k,$$

$$\Delta_k(:, 1) = \begin{cases} \text{scale}_{1,k} \cdot (\mathbf{B}_k(:, 2) - \mathbf{B}_k(:, 1)) & \text{if } k \in \mathcal{R}_1, \\ -1 \cdot \text{scale}_{1,k} \cdot (\mathbf{B}_k(:, 2) - \mathbf{B}_k(:, 1)) & \text{otherwise,} \end{cases}$$

where $\text{scale}_{1,k} \sim \mathcal{U}(0.04, 0.08) \forall k \leq K$ is the step size. A similar scheme is used for the rest of the components. Steps are small and the data remains smooth. See the GitHub repo for more details on data generation.

The second column of Figure 2 presents the results. Since the data is smooth, increasing regularization helps mitigate noise. However, PARAFAC2-based methods struggle more to recover the ground truth without strong regularization due to enforced constraint. In contrast, dCMF shows better performance due to its greater structural flexibility.

D. Non-identity transition matrix

Here, the goal is to investigate whether incorporating prior knowledge via \mathbf{H} in dCMF increases accuracy. We consider two settings, i.e., one with an orthogonal and another with a non-orthogonal transition matrix \mathbf{H} .

In the first setting (orthogonal \mathbf{H}), thirty datasets are created, with $\mathbf{A} \sim \mathcal{N}(0, 1)$ and $\mathbf{C} \sim \mathcal{U}(1, 15)$. For each evolving pattern, certain consecutive indices are chosen to be active and we generate the active indices of (the orthogonal) $\mathbf{B}_1 \sim \mathcal{N}(0, 1)$. Three random orthogonal transition matrices $\mathbf{H}^{(1)}, \mathbf{H}^{(2)}, \mathbf{H}^{(3)}$ are generated, each describing the evolution of the active entries of each pattern. These matrices are combined to form the block diagonal matrix $\mathbf{H} = \text{diag}(\mathbf{H}^{(1)}, \mathbf{H}^{(2)}, \mathbf{H}^{(3)})$. Then, we compute $\mathbf{B}_{k+1} = \mathbf{H}\mathbf{B}_k \forall k < K$, with each $\mathbf{H}^{(r)}$ only applied to the active indices of the r -th column of \mathbf{B}_k . Here, \mathbf{H} is used as prior information ($\mathbf{H} = \mathbf{H}_{\text{GND}}$) when fitting the dCMF model.

In the second setting (non-orthogonal \mathbf{H}), thirty more datasets are generated but instead, a shared non-orthogonal transition matrix is used for all components. After sampling $\mathbf{B}_1 \sim \mathcal{N}(0, 1)$, we generate $\mathbf{H} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$ with $\mathbf{V} \sim \mathcal{N}(0, 1)$ and $\text{diag}(\mathbf{\Lambda}) \sim \mathcal{N}(0.95, 0.1)$ and construct $\mathbf{B}_{k+1} = \mathbf{H}\mathbf{B}_k \forall k < K$. \mathbf{A} and \mathbf{C} are generated similarly to the previous setting. Since the transition matrix is non-orthogonal, the ground truth does not satisfy the PARAFAC2 constraint. Nevertheless, the

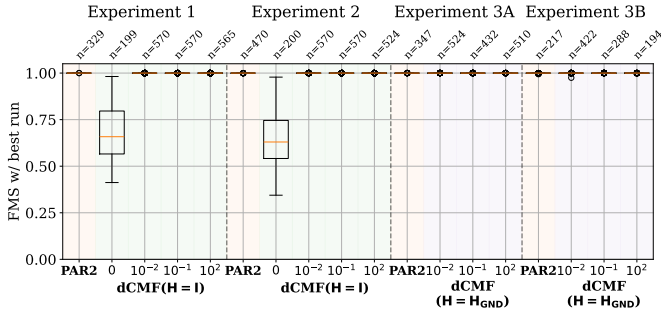


Fig. 3: Boxplots of FMS (noise $\eta = 0.75$), comparing all runs to the best run per dataset. Each experiment includes up to 570 runs (30 datasets \times 20 initializations, excluding 30 best runs). n indicates the number of valid (non-degenerate, feasible) runs.

goal remains to investigate if incorporating prior knowledge through the transition matrix increases recovery accuracy.

We compare the accuracy of the recovered factors from PARAFAC2, tPARAFAC2 and dCMF with $H = I$ and the ground truth H_{GND} of each dataset. The results are shown in the third and fourth columns of Figure 2. For the orthogonal transition matrix case, we notice that since the data is not smooth, involving smoothness either with the LDS constraint or tPARAFAC2 does not result in higher accuracy. Instead, we see PARAFAC2 performing better than tPARAFAC2 and dCMF. However, if the appropriate prior information is used, dCMF always perfectly recovers the ground truth ($\text{FMS} > 0.99$) even in the most noisy setting. For the non-orthogonal transition matrix case, the data is smooth (H_{GND} is close to identity by definition), and therefore temporal smoothness improves recovery. No dataset possesses PARAFAC2 structure, hence the decreased performance of PARAFAC2-based methods while the best results are achieved by dCMF ($H = H_{\text{GND}}$).

E. Uniqueness

Uniqueness is critical for the interpretability of the patterns extracted using factorization methods. We empirically investigate the uniqueness of dCMF by comparing all (non-failed) runs with the best run for each dataset. We consider a run as ‘failed’ if the solution was infeasible, exited due to numerical issues, was degenerate or had a function value with less than 6 decimal digits shared with the best run (to discard local minima). In Figure 3, where we also include PARAFAC2 for reference, we can observe that introducing the proposed LDS constraint (with appropriate strength) helps to achieve a unique solution. The results are similar for different noise levels.

V. CONCLUSION

In this paper, we have introduced a soft constraint for coupled matrix factorizations that regularizes evolving factors to follow the structure of a linear dynamical system. Setting the transition matrix to identity, in particular, induces temporal smoothness. Experimental results indicate that (a) dCMF can capture dynamics that do not conform to PARAFAC2 better than PARAFAC2-based methods and (b) if the transition matrix of the evolving factors is known a-priori, incorporating

it into the dCMF framework enables highly accurate recovery of the underlying patterns. As future work, we plan to focus on computational efficiency and learning the transition matrix.

REFERENCES

- [1] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.
- [2] E. Acar and B. Yener, “Unsupervised multiway data analysis: A literature survey,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 1, pp. 6–20, 2009.
- [3] R. A. Harshman, “Foundations of the PARAFAC procedure: Models and conditions for an ‘explanatory’ multi-modal factor analysis,” *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.
- [4] R. A. Harshman, “PARAFAC2: Mathematical and technical notes,” *UCLA Working Papers in Phonetics*, no. 22, pp. 30–47, 1972.
- [5] Y. Koren, “Collaborative filtering with temporal dynamics,” in *KDD’09*, pp. 447–456, 2009.
- [6] Z. Liu and M. Hauskrecht, “Learning linear dynamical systems from multivariate time series: A matrix factorization based framework,” in *SDM’16: Proc. SIAM Int. Conf. Data Mining*, pp. 810–818, 2016.
- [7] C. Chatzis, M. Pfeffer, P. Lind, and E. Acar, “A time-aware tensor decomposition for tracking evolving patterns,” in *MLSP’23: Proc. IEEE Int. Workshop Machine Learning for Signal Processing*, pp. 1–6, 2023.
- [8] C. Chatzis, C. Schenker, M. Pfeffer, and E. Acar, “tPARAFAC2: Tracking evolving patterns in (incomplete) temporal data,” *arXiv*, 2024.
- [9] H.-F. Yu, N. Rao, and I. S. Dhillon, “Temporal regularized matrix factorization for high-dimensional time series prediction,” in *Advances in Neural Information Processing Systems*, vol. 29, pp. 847–855, 2016.
- [10] J.-G. Jang, J. Lee, J. Park, and U. Kang, “Accurate PARAFAC2 decomposition for temporal irregular tensors with missing values,” in *IEEE Int. Conf. Big Data*, pp. 982–991, 2022.
- [11] Y. Sun, N. J. Yuan, Y. Wang, X. Xie, K. McDonald, and R. Zhang, “Contextual intent tracking for personal assistants,” in *KDD’16*, p. 273–282, 2016.
- [12] A. P. Appel, R. L. Cunha, C. C. Aggarwal, and M. M. Terakado, “Temporally evolving community detection and prediction in content-centric networks,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 3–18, 2018.
- [13] K. Kawabata, S. Bhatia, R. Liu, M. Wadhwa, and B. Hooi, “SSMF: shifting seasonal matrix factorization,” in *NeurIPS*, 2021.
- [14] W. Yu, C. C. Aggarwal, and W. Wang, “Temporally factorized network modeling for evolutionary network analysis,” in *WSDM’17: Proc. Tenth ACM Int. Conf. Web Search and Data Mining*, pp. 455–464, 2017.
- [15] S. Henrot, J. Chanussot, and C. Jutten, “Dynamical spectral unmixing of multitemporal hyperspectral images,” *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3219–3232, 2016.
- [16] R. A. Borsoi, T. Imbiriba, and P. Closas, “Dynamical hyperspectral unmixing with variational recurrent neural networks,” *IEEE Transactions on Image Processing*, vol. 32, pp. 2279–2294, 2023.
- [17] A. P. Singh and G. J. Gordon, “Relational learning via collective matrix factorization,” in *KDD’08*, pp. 650–658, 2008.
- [18] H. A. L. Kiers, J. M. F. ten Berge, and R. Bro, “PARAFAC2—Part I. a direct fitting algorithm for the parafac2 model,” *Journal of Chemometrics*, vol. 13, no. 3–4, pp. 275–294, 1999.
- [19] M. Roald, S. Bhinge, C. Jia, V. Calhoun, T. Adali, and E. Acar, “Tracing network evolution using the PARAFAC2 model,” in *ICASSP’20*, pp. 1100–1104, 2020.
- [20] M. Roald, C. Schenker, V. D. Calhoun, T. Adali, R. Bro, J. E. Cohen, and E. Acar, “An AO-ADMM approach to constraining PARAFAC2 on all modes,” *SIAM Journal on Mathematics of Data Science*, vol. 4, no. 3, pp. 1191–1222, 2022.
- [21] K. Huang, N. D. Sidiropoulos, and A. P. Liavas, “A flexible and efficient algorithmic framework for constrained matrix and tensor factorization,” *IEEE Transactions on Signal Processing*, vol. 64, no. 19, pp. 5052–5065, 2016.
- [22] J. Kossaifi, Y. Panagakis, A. Anandkumar, and M. Pantic, “Tensorly: Tensor learning in python,” *Journal of Machine Learning Research*, vol. 20, no. 26, 2019.
- [23] M. Roald, “Matcouply: Learning coupled matrix factorizations with python,” *SoftwareX*, vol. 21, p. 101292, 2023.
- [24] M. Roald and Y. M. Moe, “Tlviz: Visualising and analysing tensor decomposition models with python,” *Journal of Open Source Software*, vol. 7, no. 79, p. 4754, 2022.