Tensor Graph Decomposition for Temporal Networks

Das, B.; Isufi, E.

DOI
10.1109/ICASSP48485.2024.10447234

Publication date
2024

Document Version
Final published version

Published in
ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

Citation (APA)
Das, B., & Isufi, E. (2024). Tensor Graph Decomposition for Temporal Networks. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 9901-9905) https://doi.org/10.1109/ICASSP48485.2024.10447234

ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10447234

TENSOR GRAPH DECOMPOSITION FOR TEMPORAL NETWORKS

Bishwadeep Das and Elvin Isufi

ABSTRACT

Temporal networks arise due to certain dynamics influencing their connections or due to the change in interactions between the nodes themselves, as seen for example in social networks. Such evolution can be algebraically represented by a three-way tensor, which lends itself to using tensor decompositions to study the underpinning factors driving the network evolution. Low rank tensor decompositions have been used for temporal networks but mostly with a focus on downstream tasks and have been seldom used to study the temporal network itself. Here, we use the tensor decomposition to identify a limited number of key mode graphs that can explain the temporal network, and which linear combination can represent its evolution. For this, we put for a novel graph-based tensor decomposition approach where we impose a graph structure on the two modes of the tensor and a smoothness on the temporal dimension. We use these mode graphs to investigate the temporal network and corroborate their usability for network reconstruction and link prediction.

Index Terms— Dynamic networks, tensors, adjacency matrix, alternating optimization.

1. INTRODUCTION

Studying the driving mechanisms behind temporal networks is key to understanding complex systems where relations change over time [1]. In studying these networks, we often resort to studying their algebraic representation and identifying key properties that could explain their dynamics [1]. Three-way tensors have consistently been used to represent temporal networks, where we stack the time-varying adjacency matrices across the third dimension, usually indicating time [2–4]. This representation allows to use tensor decompositions [5–7] to analyze, compress, or complete temporal networks.

Tensor decomposition has been used for different tasks in temporal networks. For example, [8] uses the CP decomposition to infer graph topologies from correlation matrices, whereas [2] uses it for community detection. The work in [9] uses a tensor decomposition to infer dynamic network states from EEG signals. Also [10] uses a temporally-constrained CP decomposition on a stacked adjacency tensor to detect dynamic network states corresponding to communities. The block term decomposition (BTD) is used in [3] to decompose tensors comprising multi-aspect graphs and in [11] to investigate functional brain temporal networks. In other network-based disciplines, tensors have also been used for temporal knowledge base completion [12], analyzing temporal social networks [13], temporal collaborative filtering [14], or for topology inference from data [15]. Tensors have also been used to compress temporal networks in [16] or to identify communities in [17].

All these works rely on some sort of low-rank decomposition. This decomposition could identify key factors to solve the specific tasks but typically does not possess any graph structure, which may be of interest to identify key mode graphs that can represent the temporal network. Thus, we question if it is possible to obtain a few key mode graphs that can help us unravel the dynamics behind temporal networks. Targeting this question, we resort to low-rank tensor decompositions and propose a new approach to identify a few graph modes in which linear combination contributes to representing the overall dynamics. We formulate this as an optimization problem where we impose a sparse graph structure on the modes of decomposition akin to the recent conditions adopted in graph signal processing [18]. We also impose a smooth temporal variation on the temporal graph as well as a non-overlapping support condition across the mode graphs to identify different meaningful structures. This optimization problem is solved in an alternating fashion between the mode graphs and the temporal factor variables.

We compare the proposed approach with BTD alternatives that do not impose a graph structure as well as with [11] for adjacency tensor reconstruction and completion tasks. Besides achieving a substantially superior performance, our findings shed light on the features of the mode graphs, ultimately, providing a new perspective to study temporal networks and adjacency tensors in general.

2. PROBLEM FORMULATION

Consider a temporal graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$ with node set $\mathcal{V} = \{1, \ldots, N\}$ and evolving edge set $\mathcal{E}_t$ containing edges between nodes i and j at time $t = 1, \ldots, T$. We represent the temporal graph via its three-dimensional adjacency tensor $\underline{\mathbf{X}} \in \mathbb{R}^{N\times N\times T}$, where the t-th frontal slice $\underline{\mathbf{X}}_{:,:,t} \in \mathbb{R}^{N\times N}$ is the adjacency matrix at time t. Our ultimate objective is to represent the evolution of $\mathcal{G}_t$ as a linear combination of $R \ll T$ *modal* graphs with adjacency matrices $\mathbf{A}_1, \ldots, \mathbf{A}_R \in \mathbb{R}^{N\times N}$ that could unravel important hidden connectivity factors, as well as be used for subsequent downstream tasks such as temporal link prediction, graph reconstruction, or anomaly detection [13, 14].

To obtain the modal graphs, we propose to approximate the adjacency tensor $\mathbf{X}$ via the following decomposition

$$\hat{\underline{\mathbf{X}}} = \sum_{r=1}^{R} \mathbf{A}_r \circ \mathbf{c}_r := [[\mathbf{A}, \mathbf{C}]] \tag{1}$$

where $\mathbf{c}_r \in \mathbb{R}^T$ and the outer product $\mathbf{A}_r \circ \mathbf{c}_r \in \mathbb{R}^{N\times N\times T}$ is the rth rank-one block term. Decomposition (1) expresses $\underline{\mathbf{X}}$ as a sum of R rank one block term tensors, each of which corresponds to an adjacency matrix scaled over time. This is different from the popular low-rank decomposition, which would require also a low-rank structure on each adjacency matrix $\mathbf{A}_r$ but the latter is typically

This research is supported by the TTW-OTP project GraSPA (project number 19497) financed by the Dutch Research Council (NWO) and by the TU Delft AI programme. The authors are with the Multimedia Computing Group, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, The Netherlands. e-mails: {b.das,e.isufi-1}@tudelft.nl

unrealistic. The tth temporal slice of $\underline{\mathbf{X}}$ reads as

$$\underline{\mathbf{X}}_{:,:,t} = \hat{\underline{\mathbf{X}}}_{:,:,t} + \underline{\mathbf{E}}_{:,:,t} = \sum_{r=1}^{R} c_{rt}\mathbf{A}_r + \underline{\mathbf{E}}_{:,:,t}. \tag{2}$$

where c_{rt} is the tth element of $\mathbf{c}_r$ and $\underline{\mathbf{E}}_{:,:,t}$ is the tth temporal slice of a tensor $\underline{\mathbf{E}}$ that captures dynamics like added/subtracted edges or noise.

Given $\underline{\mathbf{X}}$, our objective transforms into estimating $\mathbf{A}_r$ and $\mathbf{c}_r$ for all $r = 1, \ldots, R$ by imposing a graph structure into each adjacency matrix $\mathbf{A}_r$. We achieve this by solving

$$\begin{aligned} &\underset{\mathbf{A},\mathbf{C}}{\text{minimize}} \quad l(\underline{\mathbf{X}}, [[\mathbf{A}, \mathbf{C}]]) + \sum_{r=1}^{R} g(\mathbf{A}_r) + h(\mathbf{A}) + i(\mathbf{C}) \\ &\text{subject to } \mathbf{A}_r \in \mathcal{S},\ \mathbf{A} = [\mathbf{A}_1, \ldots, \mathbf{A}_R],\ \mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_R] \end{aligned} \tag{3}$$

where $l(\cdot)$ is a data-fitting term that we fix to the squared Frobenius norm $l(\cdot) = ||\underline{\mathbf{X}} - \sum_{r=1}^{R} \mathbf{A}_r \circ \mathbf{c}_r||_F^2$, and $g(\cdot)$, $h(\cdot)$, $i(\cdot)$ impose desired constraints on each adjacency matrix $\mathbf{A}_r$ separately, on all of them $\mathbf{A}$, and on the temporal factor matrix $\mathbf{C}$, respectively. Finally, set $\mathcal{S}$ imposes desired properties on matrices $\mathbf{A}_r$. We recognize that problem (3) is generic and could be tailored to different setting. We specifically consider:

- The adjacency matrices to be elementwise positive and have a zero diagonal (no self-loops). This is achieved via the constraint set $\mathcal{S} = \{\mathbf{S} \in \mathbb{R}^{N\times N} : \mathbf{S} \geq \mathbf{0}, \text{tr}(\mathbf{S}) = 0\}$ [19].
- Sparsity on each adjacency matrix through the ℓ_1-norm $g(\mathbf{A}_r) = ||\mathbf{A}_r||_1$ since real-world and meaningful graphs are sparse [20]. For graphs with positive edge weights, this leads to $g(\mathbf{A}_r) = \mathbf{1}_N^\top \mathbf{A}_r \mathbf{1}_N$, where $\mathbf{1}$ is the all one vector.
- Penalizing shared support across $\mathbf{A}_r$s, i.e., that one interaction appears ideally on a single mode graph. Our rationale is to avoid redundancy in the different graphs $\mathbf{A}_r$. This could be achieved via the penalty term $h(\mathbf{A}) = \sum_{\ell,k=1}^{R} \text{vec}(\mathbf{A}_\ell)^\top \text{vec}(\mathbf{A}_k)$ with $\text{vec}(\cdot)$ the vectorization operator. Since all adjacency matrices are element-wise positive, the dot product of their vector forms is lower when they share fewer edges with a lower weight; i.e., ideally if $[\mathbf{A}_\ell]_{i,j} \neq 0$, then $[\mathbf{A}_k]_{i,j} = 0$ for $k \neq \ell$. By penalizing $h(\mathbf{A})$ via a weight $\beta/2$, we could control this shared support as sharing some edges may still be beneficial to find consistency across the modal graphs.
- A slow varying temporal graph $\mathcal{G}_t$ with no sudden changes, i.e., each $\mathbf{c}_r$ changes slowly over time. This is the case for edge-rewirings, where a small number of edges are rewired over time, as seen in biological networks [21]. We impose this by setting $i(\mathbf{C}) = ||\mathbf{D}\mathbf{C}||_F^2 = \sum_{r=1}^{R} ||\mathbf{D}\mathbf{c}_r||_2^2$, where $\mathbf{D}$ is the $(T-1) \times T$ temporal difference matrix.

With this in place, our problem reads as

$$\begin{aligned} &\min_{\mathbf{A},\mathbf{C}} \frac{1}{2}\left\|\underline{\mathbf{X}} - [[\mathbf{A}, \mathbf{C}]]\right\|_F^2 + \gamma \sum_{r=1}^{R} \mathbf{1}^\top \mathbf{A}_r \mathbf{1} + \frac{\beta}{2} \sum_{\substack{\ell,k=1 \\ \ell\neq k}}^{R} \text{vec}(\mathbf{A}_\ell)^\top \text{vec}(\mathbf{A}_k) \\ &\quad + ||\mathbf{D}\mathbf{C}||_F^2 \\ &\text{subject to } \mathbf{A}_r \in \mathcal{S},\ \mathbf{A} = [\mathbf{A}_1, \ldots, \mathbf{A}_R],\ \mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_R] \end{aligned} \tag{4}$$

where scalars $\gamma, \beta > 0$ give emphasis to the sparsity and to the non-alignment constraint, respectively.

Remark 1. *Note that* (1) *is can also be seen as a matrix factorization. We found the Tensor Formulation to be more useful in terms of representing the dynamics, but it is true that for the optimization problem, this amounts to solving a low rank matrix problem.*

3. TENSOR GRAPH DECOMPOSITION METHOD

Problem (4) is jointly non-convex but it is disjointly convex in each $\mathbf{A}_r$ and $\mathbf{C}$, and it is typically approached via alternating minimization [22, 23]. We here detail the steps to solve it by alternating between $\mathbf{A}$ and $\mathbf{C}$, and then discuss its computation cost.

3.1. Solving for the Adjacency Matrices

Given some initialization of the matrices $\mathbf{A}_r$ and vectors $\mathbf{c}_r$, we solve for each matrix $\mathbf{A}_r$ sequentially. Thus, we first write the loss function explicitly as a function of $\mathbf{A}_r$. For this, we vectorize each observed adjacency matrix over time and build matrix $\mathbf{X}_\circ = [\text{vec}(\underline{\mathbf{X}}_{:,:,1}), \text{vec}(\underline{\mathbf{X}}_{:,:,2}), \ldots, \text{vec}(\underline{\mathbf{X}}_{:,:,T})] \in \mathbb{R}^{N^2\times T}$. We also build the analogous matrix for the modal graphs $\mathbf{A}_\circ = [\text{vec}(\mathbf{A}_1), \text{vec}(\mathbf{A}_2), \ldots, \text{vec}(\mathbf{A}_R)] \in \mathbb{R}^{N^2\times R}$ and write the data-fitting loss as $l(\mathbf{X}, [[\mathbf{A}, \mathbf{C}]]) = ||\mathbf{X}_\circ - \mathbf{A}_\circ \mathbf{C}^\top||_F^2$. Then, upon defining the connectivity residue that is not represented by the tuple $(\mathbf{A}_r, \mathbf{c}_r)$,

$$\mathbf{Y}_r = \mathbf{X}_\circ - \sum_{k=1,k\neq r}^{R} \text{vec}(\mathbf{A}_k)\mathbf{c}_k^\top, \tag{5}$$

we can express the loss in terms of $\mathbf{A}_r$ as

$$l(\mathbf{X}, [[\mathbf{A}, \mathbf{C}]]) = \frac{1}{2}||\mathbf{Y}_r - \text{vec}(\mathbf{A}_r)\mathbf{c}_r^\top||_F^2. \tag{6}$$

The optimization problem for $\mathbf{A}_r$ then reads as

$$\min_{\mathbf{A}_r} \frac{1}{2}||\mathbf{Y}_r - \text{vec}(\mathbf{A}_r)\mathbf{c}_r^\top||_F^2 + \gamma\mathbf{1}^\top\mathbf{A}_r\mathbf{1} + \frac{\beta}{2}\text{vec}(\mathbf{A}_r)^\top \sum_{\substack{k=1\\k\neq r}}^{R} \text{vec}(\mathbf{A}_k)$$

subject to $\mathbf{A}_r \in \mathcal{S}$.

Dual problem. By considering the structure of set $\mathcal{S}$, we can write the Lagrangian of the above problem as

$$\begin{aligned} L(\mathbf{A}_r, \mathbf{\Lambda}, \nu) &= \frac{1}{2}||\mathbf{Y}_r - \text{vec}(\mathbf{A}_r)\mathbf{c}_r^\top||_F^2 + \gamma\mathbf{1}^\top\mathbf{A}_r\mathbf{1} \\ &+ \frac{\beta}{2}\text{vec}(\mathbf{A}_r)^\top \sum_{k=1,k\neq r}^{R} \text{vec}(\mathbf{A}_k) - \text{tr}(\mathbf{\Lambda}^\top\mathbf{A}_r) + \nu\text{tr}(\mathbf{A}_r) \end{aligned} \tag{7}$$

where $\mathbf{\Lambda} \geq \mathbf{0}$ and ν are the dual variable. Then, we consider the inverse vectorization operator that gives matrix $\mathbf{Z}_r = \text{vec}^{-1}(\mathbf{Y}_r\mathbf{c}_r)$ and rewrite the Lagrangian in terms of matrix $\mathbf{A}_r$ as

$$\begin{aligned} L(\mathbf{A}_r, \mathbf{\Lambda}, \nu) &= \frac{||\mathbf{c}_r||^2}{2}\text{tr}(\mathbf{A}_r^\top\mathbf{A}_r) - \text{tr}(\mathbf{A}_r^\top\mathbf{Z}_r) + \frac{1}{2}\text{tr}(\mathbf{Y}_r\mathbf{Y}_r^\top) \\ &+ \gamma\mathbf{1}^\top\mathbf{A}_r\mathbf{1} + \frac{\beta}{2}\text{tr}(\sum_{\substack{k=1\\k\neq r}}^{R} \mathbf{A}_r^\top\mathbf{A}_k) - \text{tr}(\mathbf{\Lambda}^\top\mathbf{A}_r) + \nu\text{tr}(\mathbf{A}_r) \end{aligned} \tag{8}$$

where the first line on the r.h.s. is due to the loss term [cf. (6)].[1] The Lagrangian is convex in $\mathbf{A}_r$ and thus we can solve the optimization problem in the dual domain. The dual function $d(\mathbf{\Lambda}, \nu) =$

[1] This is obtained by exploiting $||\mathbf{A}||_F^2 = \text{tr}(\mathbf{A}\mathbf{A}^\top)$ and $\text{vec}(\mathbf{A}^\top)\text{vec}(\mathbf{B}) = \text{tr}(\mathbf{A}^\top\mathbf{B})$.

Algorithm 1 Tensor Graph Decomposition (TGD)

```
**Input:** Tensor $\underline{\mathbf{X}} \in \mathbb{R}^{N\times N\times T}$; tensor rank $R$, $N_i$, $N_d$, $K$, $\gamma$, $\beta$.
**Output** Mode graph adjacency matrices $\{\mathbf{A}_r\}_{r=1}^{R}$, $\mathbf{C}$
**Initialization**: Initialize $\{\mathbf{A}_r\}_{r=1}^{R}$, $\mathbf{C}$
**For** $k = 1 : K$ (alternating steps between $\{\mathbf{A}_r\}_{r=1}^{R}$ and $\mathbf{C}$)
    **For** $n_i = 1 : N_i$ (update steps for each $\mathbf{A}_r$)
        **For** $r = 1 : R$ (scan over matrices $\{\mathbf{A}_r\}_{r=1:R}$)
        (Optimize for $\mathbf{A}_r$)
        Compute $\mathbf{D}_r$ as in (9)-(10)
            **For** $n_d = 1 : N_d$ (update steps of the dual)
            Perform the dual update in (11)
            **end**
        Update $\mathbf{A}_r$ as in (9)
        **end**
    **end**
Update $\mathbf{C}$ as in (12)
**end**
```

$\min_{\mathbf{A}_r} L(\mathbf{A}_r, \mathbf{\Lambda}, \nu)$ is the Lagrangian obtained at $\mathbf{A}_r$ corresponding to $\nabla_{\mathbf{A}_r} L(\mathbf{A}_r, \mathbf{\Lambda}, \nu) = \mathbf{0}$ which is obtained at

$$\mathbf{A}_r = \frac{1}{||\mathbf{c}_r||^2}(\mathbf{\Lambda} + \mathbf{D}_r - \nu\mathbf{I}) \tag{9}$$

and where . The dual function has then the form

$$d(\mathbf{\Lambda}, \nu) = \frac{1}{||\mathbf{c}_r||^2}\Bigg(\text{tr}((\mathbf{\Lambda} + \mathbf{D}_r - \nu\mathbf{I})^\top(\mathbf{\Lambda} + \mathbf{D}_r - \nu\mathbf{I})) - \text{tr}((\mathbf{\Lambda} + \mathbf{D}_r - \nu\mathbf{I})^\top\mathbf{Z}_r) + \frac{1}{2}\text{tr}(\mathbf{Y}_r\mathbf{Y}_r^\top) + \gamma\mathbf{1}^\top(\mathbf{\Lambda} + \mathbf{D}_r - \nu\mathbf{I})\mathbf{1} + \frac{\beta}{2}\text{tr}(\mathbf{\Lambda}^\top(\mathbf{\Lambda} + \mathbf{D}_r - \nu\mathbf{I})) + \nu\text{tr}(\mathbf{\Lambda} + \mathbf{D}_r - \nu\mathbf{I})\Bigg). \tag{10}$$

In turn, this leads to solving $\max_{\mathbf{\Lambda},\nu} \quad d(\mathbf{\Lambda}, \nu)$ subject to $\mathbf{\Lambda} \geq \mathbf{0}$ which is concave in both variables. We can solve for its maximum by alternating N_d times between the iterations updates

$$\mathbf{\Lambda} = \max\{0; \nu\mathbf{I} - \mathbf{D}_r\} \quad \text{and} \quad \nu = \frac{1}{\mathrm{N}}\mathrm{tr}(\mathbf{\Lambda} + \mathbf{D}_\mathrm{r}) \tag{11}$$

until convergence and along with it for $\mathbf{A}_r$ [cf. (9)]. The above procedure is repeated for N_i steps across all $\mathbf{A}_r$s or until the cost function reaches stabilizes before proceeding with the update on $\mathbf{C}$.

3.2. Solving for the Temporal Factor

Given the update of $\mathbf{A}$, we now update $\mathbf{C}$, which consists of minimizing the convex cost $||\mathbf{X}_\circ - \mathbf{A}_\circ\mathbf{C}^\top||_F^2 + ||\mathbf{D}\mathbf{C}||_F^2$. Taking the derivative and using the $\text{vec}(\cdot)$ and $\text{vec}^{-1}(\cdot)$ operators, we get the closed form solution

$$\mathbf{C}^\star = \text{vec}^{-1}\big((\mathbf{A}_\circ^\top\mathbf{A}_\circ \otimes \mathbf{I}_\mathrm{T} + \mathbf{I}_\mathrm{R} \otimes \mathbf{D}^\top\mathbf{D})^{-1}\text{vec}(\mathbf{X}_\circ^\top\mathbf{A}_\circ)\big). \tag{12}$$

Algorithm 1 summarizes the tensor graph decomposition procedure.

Hyperparameters and computational complexity. The proposed approach has the following hyperparameters that affect the solution: i) the sparsity regularizer weight γ in (4); ii) the shared support penalization parameter β in (4); iii) the number of iterations K to alternate between the solutions of $\mathbf{A}$ and $\mathbf{C}$; iv) the number of alternating iterations N_i over the $\mathbf{A}_r$s; v) the alternating iterations N_d in (11). The parameters to be estimated is $(N^2 - N)R + TR$: $(N^2 - N)R$ are the parameters of the R matrices $\mathbf{A}_r$ with zero diagonals and TR are the parameters of $\mathbf{C}$.

The computational complexity is of order $\mathcal{O}(KRN_iTN^2 + KRN_iN_dN^2 + KR^3T^3)$. The computations for estimating $\mathbf{A}$ are governed by equations (9) and (4). In both of them, we need to compute matrix $\mathbf{D}_r$ that has a cost of order $\mathcal{O}(TN^2)$ that is governed by $\mathbf{Z}_r$. In (9), we update the R matrices $\mathbf{A}_r$ for N_i steps leading to a cost $\mathcal{O}(RN_iTN^2)$. Then, in (11) we alternate over N_d iterations and have N^2 element-wise projections, leading to a cost of order $\mathcal{O}(RN_iN_dN^2)$. For the $\mathbf{C}$ update, we ignore the computation of $\mathbf{D}^\top\mathbf{D}$ because of its high sparsity and the dominant terms are $\mathbf{A}_\circ^\top\mathbf{A}_\circ$ and the inverse of cost $\mathcal{O}(R^3T^3)$. All these have to be repeated K times leading to the overall cost.

Remark 2. *Our focus is in recovering the components given R. A discussion on selecting R is omitted because it is out of the scope of this paper. Interested readers can refer to the relevant literature.*

4. EXPERIMENTAL RESULTS

We corroborate the proposed approach on two datasets for the tasks of graph reconstruction and spatiotemporal link completion.

4.1. Experimental Setup

We consider a synthetic and a real data experimental setting. The synthetic experiments are run over a stochastic block model (SBM) graph with 100 nodes and two communities. We use a standard edge rewiring to impose graph dynamics, where at each time instant, we re-wire 5 edges at random for $T = 50$ time instants. This is done on 5 realizations of the SBM. For the real data experiments, we consider the Enron employee email interaction dataset [24]. It contains 151 users as nodes and an edge indicates an email exchange at a particular time stamp. We split the time intervals into $T = 50$ blocks and obtain the adjacency matrix for each block based on the edge data at that time. We compare the following alternatives:

1. **Uncon:** this comprises solving (4) without the constraints. It ignores the graph structure on $\mathbf{A}_r$s.
2. **BTD:** this is the classical low-rank block term decomposition [7], which ignores any graph structure all over the problem.[2].
3. **SBTD:** this is the BTD approach proposed in [11] that forces symmetry and low-rankness on each $\mathbf{A}_r$ while also ignoring a graph structure in it.
4. **TGD (proposed):** Tensor Graph Decomposition is the approach in Algorithm 1. The hyper-parameters γ and β are grid searched respectively from $[0.01, 10]$ and $[0.01, 16]$. We also set $K = 10$, $N_i = 10$, and $N_d = 5$.

We set the number of components to $R = 3$ and the internal rank for the BTS approaches to five. First, we consider the task of tensor reconstruction, where we aim to obtain the original tensor from its low rank approximations. We measure the performance via the the normalized squared error $\text{NMSE} = ||\underline{\mathbf{X}} - \hat{\underline{\mathbf{X}}}||_F^2/||\underline{\mathbf{X}}||_F^2$. Then, we consider the task of link prediction, where we mask a section of the observed tensor and use the reconstructed $\hat{\underline{\mathbf{X}}}$ to predict links over it. We have two types of masking (i) *uniformly at random*, where we mask randomly 20% of the edges; (ii) *unobserved nodes*, where we

[2] `http://dimitri.nion.free.fr/Codes/Tensor_Decompositions.html`

Table 1. Average NMSE and F1 score for all methods for the stochastic block model and the Enron dataset. (top row block) NMSE for reconstruction; (middle) F1 for uniformly at random spatiotemporal link prediction (bottom) F1 for node-specific link prediction. All standard deviations are of order 10^{-2} or lower.

Dataset	**TGD** $\gamma = 0.01$ $\beta = 0.01$	**TGD** $\gamma = 9$ $\beta = 16$	**Uncon**	**BTD**	**SBTD**
SBM	0.15	0.24	0.15	0.37	0.44
Enron	0.74	0.82	0.71	0.76	0.98
SBM	0.92	0.93	0.58	0.48	0.48
Enron	0.18	0.40	0.07	0.03	0.03
SBM	0.90	0.90	0.52	0.43	0.43
Enron	0.17	0.19	0.06	0.03	0.0

Table 2. Graph entropy (no. of connected components) for the recovered components for different γ and β on the SBM graph.

	$\beta = 0.01$	$\beta = 6$	$\beta = 16$
$\gamma = 0.01$	4.35(1)	4.29(1)	4.18(1.2)
$\gamma = 5$	4.2(1)	4.25(1.2)	4.12(1.26)
$\gamma = 9$	4.2(1.3)	4.19(1.4)	3.94(1.66)

remove interactions between a random set of 20 nodes after $t = 25$. We measure the performance via the F1 score.

In addition to the task-driven goodness of fit metrics, we also evaluate the graph entropy, which is the entropy of the degree distribution [25], and the number of connected components of the recovered graphs for further insight. A higher degree entropy indicates a more irregular and richer structure, while a lower entropy indicates a more regular degree distribution. The strongly connected components can quantify the global structure of the matrices estimated.

4.2. Numerical Results

Table 1 shows that the proposed approach matches the performance of the the unconstrained approach for SBM, while being lower for the Enron data-set. However, the reconstruction F1 scores for both the proposed alternatives are better across all data, showcasing that a graph structure on $\mathbf{A}_r$ is a strong inductive bias to represent adjacency tensors. This enhancement is more apparent as we move towards higher values of γ and β, as in for the SBM data. This comes however, with a trade-off. The proposed method also outperforms the other alternatives, with the exception of BTD in the Enron dataset. The results suggest overall that putting more emphasis on the sparsity of the factors and penalizing shared edges across them does not affect the solution space, ultimately, leading to more insightful factors.

Table 2 shows the entropy and the number of connected components averaged over the $R = 3$ factors for the synthetic data. For comparison, the starting graph has an entropy of 4.05 and one strongly connected component. An increase in γ and β in general results in a lower average entropy and a higher average number of connected components. The entropy is higher than that of the starting graph, suggesting that the obtained mode graphs have more structural diversity, but combined they lead to an adjacency matrix with lesser diversity. The increase in the number of components can be attributed to the increase in the graph sparsity, which could be used for example to infer key edges in the mode graphs.

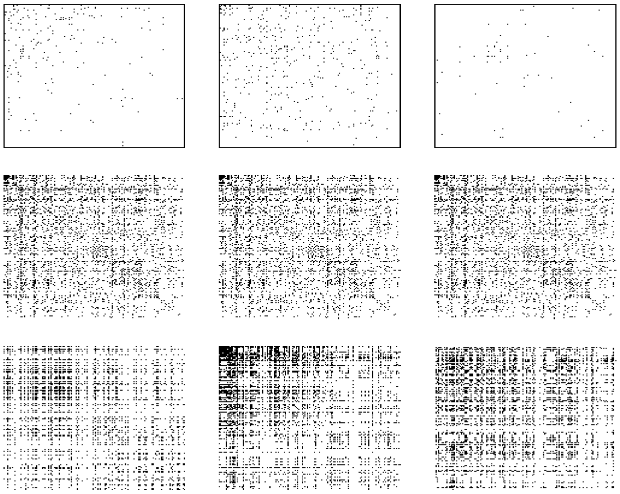

Fig. 1. (Top row) Recovered adjacency matrices with $\beta = 16$ and $\gamma = 9$ and $R = 3$. The respective graph entropies are (from left to right), 2.8, 1.2, and 2.6 and the number of connected components is 75, 134, and 78. (Middle row) Recovered factors for the unconstrained formulation. Each factor has an entropy of 7.19 and one connected component. (End row) The factors recovered from the SBTD. The factors recovered do not resemble adjacency matrices.

Figure 1 illustrates the support of the mode graphs along with their respective entropies and the number of connected components for the Enron dataset. The graphs show different scales of connectivity. We also see that the graph with more edges is structurally more diverse. For the unconstrained formulation, we see that all factors look very similar suggesting that it fits one representative adjacency matrix for depicting all the dynamics. The entropies for the factors are 7.23, 7.19, and 7.23, respectively. This suggests only one template of a richer structure is captured, which may not always give insights about the temporal evolution of the graph.

5. CONCLUSION

We propose a method to represent temporal graphs as a linear combination of a few modal graphs. We represent the temporal graph via its adjacency tensor and propose a novel tensor-based decomposition that incorporates graph-based structures along two factors representing the node-to-node pairs. We formulate a constrained optimization problem, which allows us to put graph-prior into the recovered factors as well as to impose graph sparsity and slower temporal evolution. Following conventional techniques, we solve this via alternating optimization. Numerical results on synthetic and real data corroborate our method and show there's a trade-off between the reconstruction accuracy and the structure of the recovered graphs while outperforming typical tensor low-rank decomposition that ignores the graph structure. Future work will investigate the convergence guarantees, uniqueness of the recovered matrices, as well as incorporating graph signals into the problem.

6. REFERENCES

[1] P. Holme and J. Saramäki, "Temporal networks," *Physics reports*, vol. 519, no. 3, pp. 97–125, 2012.

[2] F. Sheikholeslami and G. B. Giannakis, “Identification of overlapping communities via constrained egonet tensor decomposition,” *IEEE Transactions on Signal Processing*, vol. 66, no. 21, pp. 5730–5745, 2018.

[3] E. Gujral, R. Pasricha, and E. Papalexakis, “Beyond rank-1: Discovering rich community structure in multi-aspect graphs,” in *Proceedings of The Web Conference 2020*, 2020, pp. 452–462.

[4] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, “Tensor decomposition for signal processing and machine learning,” *IEEE Transactions on signal processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[5] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[6] L. R. Tucker, “Some mathematical notes on three-mode factor analysis,” *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[7] L. De Lathauwer and D. Nion, “Decompositions of a higher-order tensor in block terms—part iii: Alternating least squares algorithms,” *SIAM journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1067–1083, 2008.

[8] Y. Shen, B. Baingana, and G. B. Giannakis, “Tensor decompositions for identifying directed graph topologies and tracking dynamic networks,” *IEEE Transactions on Signal Processing*, vol. 65, no. 14, pp. 3675–3687, 2017.

[9] A. G. Mahyari, D. M. Zoltowski, E. M. Bernat, and S. Aviyente, “A tensor decomposition-based approach for detecting dynamic network states from eeg,” *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 1, pp. 225–237, 2016.

[10] M. McNeil, C. Mattsson, F. W. Takes, and P. Bogdanov, “Cadence: Community-aware detection of dynamic network states,” in *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. SIAM, 2023, pp. 1–9.

[11] Y. Zhu, J. Liu, and F. Cong, “Dynamic community detection for brain functional networks during music listening with block component analysis,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2023.

[12] T. Lacroix, G. Obozinski, and N. Usunier, “Tensor decompositions for temporal knowledge base completion,” *arXiv preprint arXiv:2004.04926*, 2020.

[13] A. Sapienza, A. Panisson, J. Wu, L. Gauvin, and C. Cattuto, “Anomaly detection in temporal graph data: An iterative tensor decomposition and masking approach,” in *International Workshop on Advanced Analytics and Learning on Temporal Data, AALTD 2015*, 2015.

[14] D. M. Dunlavy, T. G. Kolda, and E. Acar, “Temporal link prediction using matrix and tensor factorizations,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 5, no. 2, pp. 1–27, 2011.

[15] Y. Shen, X. Fu, G. B. Giannakis, and N. D. Sidiropoulos, “Topology identification of directed graphs via joint diagonalization of correlation matrices,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 271–283, 2020.

[16] W. Liu, A. Kan, J. Chan, J. Bailey, C. Leckie, J. Pei, and R. Kotagiri, “On compressing weighted time-evolving graphs,” in *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012, pp. 2319–2322.

[17] E. Al-Sharoa, M. Al-khassaweneh, and S. Aviyente, “A tensor based framework for community detection in dynamic networks,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2312–2316.

[18] X. Dong, D. Thanou, L. Toni, M. Bronstein, and P. Frossard, “Graph signal processing for machine learning: A review and new perspectives,” *IEEE Signal processing magazine*, vol. 37, no. 6, pp. 117–127, 2020.

[19] V. Kalofolias, “How to learn a graph from smooth signals,” in *Artificial intelligence and statistics*. PMLR, 2016, pp. 920–929.

[20] S. P. Chepuri, S. Liu, G. Leus, and A. O. Hero, “Learning sparse graphs under smoothness prior,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 6508–6512.

[21] C. Shou, N. Bhardwaj, H. Y. Lam, K.-K. Yan, P. M. Kim, M. Snyder, and M. B. Gerstein, “Measuring the evolutionary rewiring of biological networks,” *PLoS computational biology*, vol. 7, no. 1, p. e1001050, 2011.

[22] T. Hastie, R. Mazumder, J. D. Lee, and R. Zadeh, “Matrix completion and low-rank svd via fast alternating least squares,” *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 3367–3402, 2015.

[23] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[24] R. A. Rossi and N. K. Ahmed, “The network data repository with interactive graph analytics and visualization,” 2015. [Online]. Available: https://networkrepository.com

[25] M. Dehmer and A. Mowshowitz, “A history of graph entropy measures,” *Information Sciences*, vol. 181, no. 1, pp. 57–78, 2011.