Beyond Rank-1: Discovering Rich Community Structure in Multi-Aspect Graphs

Ekta Gujral*
University of California, Riverside
egujr001@ucr.edu

Ravdeep Pasricha
University of California, Riverside
rpasr001@ucr.edu

Evangelos E.Papalexakis
University of California, Riverside
epapalex@cs.ucr.edu

ABSTRACT

How are communities in real multi-aspect or multi-view graphs structured? How we can effectively and concisely summarize and explore those communities in a high-dimensional, multi-aspect graph without losing important information? State-of-the-art studies focused on patterns in single graphs, identifying structures in a single snapshot of a large network or in time evolving graphs and stitch them over time.

However, to the best of our knowledge, there is no method that discovers and summarizes community structure from a multi-aspect graph, by *jointly* leveraging information from all aspects. State-of-the-art in multi-aspect/tensor community extraction is limited to discovering clique structure in the extracted communities, or even worse, imposing clique structure where it does not exist.

In this paper we bridge that gap by empowering tensor-based methods to extract rich community structure from multi-aspect graphs. In particular, we introduce cLL1, a novel constrained Block Term Tensor Decomposition, that is generally capable of extracting higher than rank-1 but still interpretable structure from a multi-aspect dataset. Subsequently, we propose RichCom, a community structure extraction and summarization algorithm that leverages cLL1 to identify rich community structure (e.g., cliques, stars, chains, etc) while leveraging higher-order correlations between the different aspects of the graph.

Our contributions are four-fold: (a) **Novel algorithm**: we develop *cLL1*, an efficient framework to extract rich and interpretable structure from general multi-aspect data; (b) **Graph summarization and exploration**: we provide *RichCom*, a summarization and encoding scheme to discover and explore structures of communities identified by cLL1; (c) **Multi-aspect graph generator**: we provide a simple and effective synthetic multi-aspect graph generator, and (d) **Real-world utility**: we present empirical results on small and large real datasets that demonstrate performance on par or superior to existing state-of-the-art.

CCS CONCEPTS

• **Computing methodologies** → **Factor analysis**; **Machine learning**; **Regularization**;

KEYWORDS

Tensors, Block Term Decomposition, Community Structures, Multi-aspect data, Graph Summarizing

ACM Reference Format:
Ekta Gujral, Ravdeep Pasricha, and Evangelos E.Papalexakis. 2020. Beyond Rank-1: Discovering Rich Community Structure in Multi-Aspect Graphs. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3366423.3380129

1 INTRODUCTION

Multi-aspect graphs emerge in various applications, as diverse as biomedical imaging [18], social networks [24], computer vision [44], recommender systems [22], and communication networks [25], and are generally shaped using high-order tensors [20]. A simple example of such multi-aspect graph could be a three-mode tensor, where each different aspect, represented by a frontal slice of the tensor, is the adjacency matrix of a social network under a different means of communication. Each such aspect of the data is an impression of the same underlying phenomenon e.g, the formation of friendship in social networks or evolution of communities over time.

Even though there has been a significant focus on graph mining and community detection for single graphs [11, 26, 43], incorporating different aspects of the graph has only recently received attention [3, 12, 13, 32]. In fact, taking into account all aspects of a graph has been shown to lead to better results compared to "two-dimensional" counterparts [12, 13, 32]. The problem we address in this paper is the following: Given a multi-aspect graph, say, an air traffic networks of airlines like European-ATN [19], how can we efficiently describe and summarize its *community structure*? The heart of this paper is finding and visualizing the structures (e.g., clique, chain, star etc.) of communities in multi-aspect/layer graphs via tensor decomposition approach, in order to get a better insight of their properties.

To the best of our knowledge, the state-of-the-art [21] focuses on a single graph and provide vocabulary (e.g clique, star) of graph primitives using the Minimum Description Length (MDL) principle. The paper [38] is further extension of [21] for dynamic graphs and stitching the temporal patterns in the time-evolving scenario. These methods [21, 38], even though they are offering very valuable insights, but they are either focus on a single graph or find community structure in single graph and then track them over time. We propose a tensor-based method. A simple but interpretable CANDECOMP/PARAFAC tensor decomposition [2, 6, 14], aka CP (see Def. 7) method is well known and studied in the literature. The fundamental *road-block* in traditional exploratory tensor decomposition (CP [2, 6, 14] and Tucker [41]) analysis, is that the

*Corresponding author.

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380129

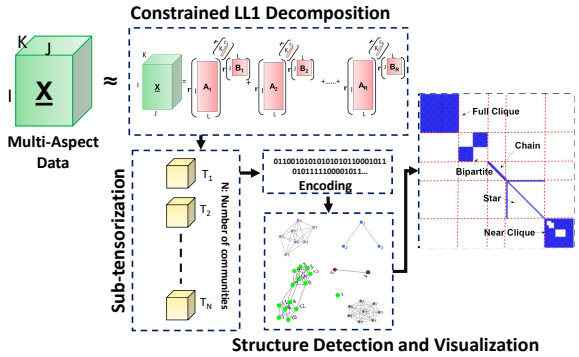

Figure 1: A toy example of RichCom: It decomposes the multi-aspect data and identifies non-overlapping as well as overlapping sets of nodes, that form sub-tensors and resultant structures like cliques, bi-bipartite, chains, stars etc. are encoded and visualized.

only form of latent structure it can discover is *rank*-1. Consider a time-evolving or a multi-aspect graph [13, 32], both of which can be expressed as a set of adjacency matrices, forming a third-order tensor $\underline{\mathbf{X}}$. These decompositions can only discover rank-1 structures in the graph, which translate into dense cliques; even if the real underlying structure in the graph is not a clique but, say, a star, CP is unable to extract it, and in the best case it will approximate it as a near-clique, or in the worst case it will fail to identify it. In the signal processing literature there exists the Block Term Decomposition (BTD) [8, 9] which shows promise; a form of BTD is the $(L, L, 1)$ decomposition[10]: $\underline{\mathbf{X}} \approx \sum_{r=1}^{R} (\mathbf{A}_r \cdot \mathbf{B}_r^T) \circ \mathbf{c}_r$ where $\mathbf{A}_r$ and $\mathbf{B}_r$ have L columns, and which essentially extracts rank-L structures in the first two dimensions. In this paper, we adapt and extend this less well-known tensor model, the Block Term Decomposition-$(L, L, 1)$ (see Def. 10), to that end. This model holds potential for "general" multi-aspect graph exploration, where structure is richer than *rank-1* but we still wish to have interpretable results. Note that paper does not focus on the community detection in data and it is out of scope of work. Here, after block term tensor decomposition, each latent component is considered as community and we focus on detecting structures of those communities.

Motivation : The motivation behind RichCom is that exploring a high-dimensional, multi-aspect graph is impossible to do manually. Thus, extraction and visualization of the main communities within such a graph is an important tool towards enabling multi-aspect graph exploration and a handful of simple structures could be easily understood, and often meaningful. Figure 2 is an illustrating example of RichCom, where the most 'important' sub-tensor that provides *EU-Air Traffic Network*'s summary is semantically interesting. Here, importance is computed based on MDL approach explained in Section (4.2). Alleviating the limitations of existing approaches, this paper presents cLL1 a novel factorization model using constrained Block Term Decomposition-rank $(L, L, 1)$ to account for the multi-aspect graph's structures. Using cLL1, factors are estimated via a novel algorithm based on the alternating optimization and alternating method of multipliers (AO-ADMM) and RichCom is used for encoding cost to discover community structures entries in the data and our contributions can be summarized as:

- **C1: Novel Problem Formulation** : We formulate the exploration and discovery of rich structures in multi-aspect graphs using a novel tensor modeling as shown in Figure 1.
- **C2: Novel and Effective Framework**: We introduce novel constrained LL1-tensor decomposition cLL1 and alternating optimization and alternating direction method of multipliers (AO-ADMM) is also developed that recovers the non-negative and sparse factors.
- **C3: Real-World Utility** : Finally, the proposed decomposition approach enables discovery of community structures via RichCom on tensor by using the recovered factors. We provide qualitative analysis of RichCom on synthetic and real, public multi-aspect datasets consisting up to thousands of edges. Experiments testify RichCom spots interesting structures like 'stars' and 'cliques' in the European Air Traffic Network (ATN) data (See Fig. 2).

Reproducibility: We make our Python and MATLAB implementation publicly available Link[1]. Furthermore, graph and tensor generator along with the small size dataset we use for evaluation are also available at the same link.

2 PRELIMINARIES

Table 1 contains the symbols used throughout the paper. Before we conceptualize the problem that our paper deals with, we define certain terms which are necessary to set up the problem.

Symbols	Definition
$\underline{\mathbf{X}}$	a tensor
$\mathbf{X}$	a matrix (uppercase, bold letter)
$\mathbf{x}$	a column vector (lowercase, bold letter)
x	a scalar (lowercase, italic letter)
$\mathbf{X}^{\dagger}$	pseudo-inverse of matrix $\mathbf{X}$
$\mathbf{X}_{(n)}$	mode-n matricization of a tensor $\underline{\mathbf{X}}$
$\|\mathbf{X}\|_F$	frobenius norm of $\mathbf{X}$
R, L	rank of tensor $\underline{\mathbf{X}}$, #components of matrix $\mathbf{X}$
$\bullet$	Tucker mode-1 product
$tenmat(\underline{\mathbf{X}}, n)$	a matrix form of a tensor $\underline{\mathbf{X}}$ in given mode n

Table 1: Table of symbols and their description

2.1 Basic definitions

Definition 1. ***Outer product*** of two vectors a and b of dimensions I and J, respectively is defined as:

$$a \circ b = ab^T \tag{1}$$

and their outer product is an $I \times J$ matrix.

Definition 2. ***Tensor***[20, 33]: A tensor is a higher order generalization of a matrix. An N-mode[2] tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_1 \cdots \times I_N}$ is the outer product of N vectors, as given in equation 2,

$$\underline{\mathbf{X}} = \mathbf{a}_1 \circ \mathbf{a}_2 \cdots \circ \mathbf{a}_N \tag{2}$$

essentially indexed by N variables i.e. $(\mathbf{a}_1, \mathbf{a}_2 \ldots, \mathbf{a}_N)$. The outer product 3-mode tensor $\underline{\mathbf{X}}$ of vectors $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ can be written as $x_{ijk} = a_i b_j c_k$ for all values of the indices.

[1] http://www.cs.ucr.edu/~egujr001/ucr/madlab/src/richcom.zip
[2] Notice that the literature (and thereby this paper) uses the above terms as well as "order" interchangeably.

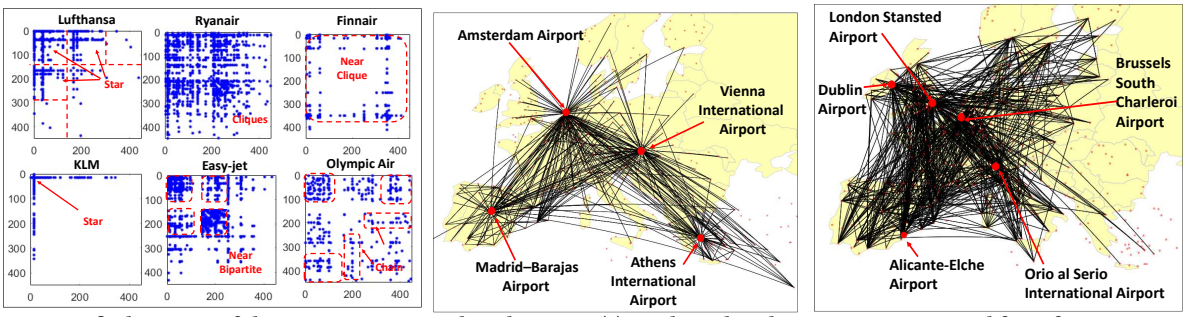

Figure 2: RichCom finds meaningful structures in EU-Airline dataset as (a) we show the adjacency matrix created from factors over multiple views showing RichCom able to find stable decompositions and detect various structures, (b) An example of ATN network of a major airline's (e.g. KLM) operating airport forming star structures and, (c) the network of a low-fare (low-cost) airline's (e.g. Ryanair) air traffic forming clique structure. In each aspect, the airports with the highest degree (hubs) are highlighted.

Definition 3. ***Rank-1*** A N-mode tensor is of rank-1 if it can be strictly decomposed into the outer product of N vectors. Therefore, we add different scaling of a sub-tensor as we introduce more modes when reconstructing the full tensor. A rank-1 3-mode tensor can be written as $\underline{\mathbf{X}} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$. A pictorial view of the rank-1 concept is shown in Figure (3).

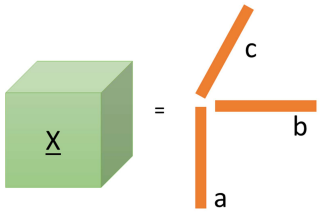

Figure 3: A rank-1 mode-3 tensor.

The rank of a tensor rank($\underline{\mathbf{X}}$) = R is defined as the minimum number of rank-1 tensors which are needed to produce $\underline{\mathbf{X}}$ as their sum.

Definition 4. ***Kronecker product***[33] is denoted by symbol $\otimes$ and the Kronecker product of two matrices $\mathbf{A} \in \mathbb{R}^{I \times L}$ and $\mathbf{B} \in \mathbb{R}^{J \times L}$ results in matrix size of $(IJ \times L^2)$ and it is defined as:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots & a_{1L}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \dots & a_{2L}\mathbf{B} \\ \vdots & \vdots & & \vdots \\ a_{I1}\mathbf{B} & a_{I2}\mathbf{B} & \dots & a_{IL}\mathbf{B} \end{bmatrix} \tag{3}$$

Definition 5. ***Column-wise Khatri-Rao product***[33] : It is denoted by symbol $\odot_c$ and the column-wise Khatri-Rao product of two matrices $\mathbf{A} \in \mathbb{R}^{I \times L}$ and $\mathbf{B} \in \mathbb{R}^{J \times L}$, $\mathbf{A} \odot_c \mathbf{B} \in \mathbb{R}^{IJ \times L}$ is defined as:

$$\mathbf{A} \odot_c \mathbf{B} = \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_2 \otimes \mathbf{b}_2 & \dots \mathbf{a}_L \otimes \mathbf{b}_L \end{bmatrix} \tag{4}$$

In case of a and b are in vector form, then the Kronecker and column-wise Khatri-Rao products are same, i.e., $\mathbf{a} \otimes \mathbf{b} = \mathbf{a} \odot_c \mathbf{b}$

Definition 6. ***Partition-wise Kronecker product*** [28] : Let $\mathbf{A} = [\mathbf{A}_1 \ \mathbf{A}_2 \dots \mathbf{A}_R] \in \mathbb{R}^{I \times LR}$ and $\mathbf{B} = [\mathbf{B}_1 \ \mathbf{B}_2 \dots \mathbf{B}_R] \in \mathbb{R}^{I \times LR}$ are two partitioned matrices. The partition-wise Kronecker product is defined by:

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} \mathbf{A}_1 \otimes \mathbf{B}_1 & \mathbf{A}_2 \otimes \mathbf{B}_2 & \dots & \mathbf{A}_R \otimes \mathbf{B}_R \end{bmatrix} \tag{5}$$

Definition 7. ***CANDECOMP/PARAFAC (CP)***[2, 6, 14]: The CP decomposition of a N-mode tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_1 \cdots \times I_N}$ with Rank R is defined as the sum of outer product rank-1 components:

$$\underline{\mathbf{X}} = \sum_{r=1}^{R} \mathbf{a}_1^{(r)} \circ \mathbf{a}_2^{(r)} \cdots \circ a_N^{(r)} \tag{6}$$

The factor matrices $(\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_N)$ are the combination of the vectors from the rank-1 components:

$$\mathbf{A}_i = [\mathbf{a}_i^{(1)} \ \mathbf{a}_i^{(2)} \dots \mathbf{a}_i^{(R)}] \tag{7}$$

Definition 8. ***Tucker Decomposition***[41] : A decomposition of a 3-mode tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$ with Rank P, Q, and R is defined as the sum of outer product rank-1 components and one small core tensor $\mathcal{G} \in \mathbb{R}^{P \times Q \times R}$:

$$\underline{\mathbf{X}} \approx \mathcal{G} \bullet_1 \mathbf{A} \bullet_2 \mathbf{B} \bullet_3 \mathbf{C} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} \mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{c}_r \tag{8}$$

Definition 9. ***BTD - (L, M, N)*** : De Lathauwer at et. [8, 9] introduce a new type of tensor decomposition that unifies the Tucker and the CP decomposition and refereed as Block Term Decomposition (BTD). The BTD of a 3-mode tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$, shown in figure 4, is a sum of rank-(L, M, N) terms is a represented as:

$$\underline{\mathbf{X}} \approx \sum_{r=1}^{R} \mathcal{G}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{B}_r \bullet_3 \mathbf{C}_r \tag{9}$$

The factor matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ is defined as $\mathbf{A} = [\mathbf{A}_1 \ \mathbf{A}_2 \dots \mathbf{A}_R] \in$

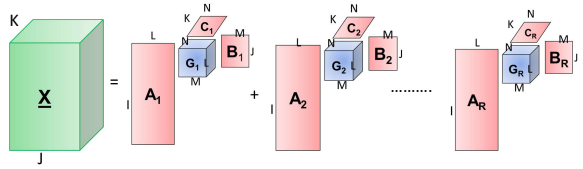

Figure 4: BTD -(L, M, N) for a third-order tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$.

$\mathbb{R}^{I \times LR}$, $\mathbf{B} = [\mathbf{B}_1 \ \mathbf{B}_2 \dots \mathbf{B}_R] \in \mathbb{R}^{J \times MR}$ and $\mathbf{C} = [\mathbf{C}_1 \ \mathbf{C}_2 \dots \mathbf{C}_R] \in \mathbb{R}^{K \times NR}$. The small core tensors $\mathcal{G}_r \in \mathbb{R}^{L \times M \times N}$ are full rank-(L, M, N). If R=1, then Block-term and Tucker decompositions are same.

Definition 10. ***BTD-(L, L, 1)***[10] : The LL1-decomposition of a 3-mode tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$ with tensor rank R is a sum of blocks with rank-$(L_r, L_r, 1)$, is a represented as:

$$\underline{\mathbf{X}} \approx \sum_{r=1}^{R} (\mathbf{A}_r \cdot \mathbf{B}_r^T) \circ \mathbf{c}_r \tag{10}$$

where factor blocks $A_r \in \mathbb{R}^{I \times L_r}$ and the matrix $B_r \in \mathbb{R}^{J \times L_r}$ are both rank-L_r, $1 \leq r \leq R$. Here, the factor matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ is of dimension $\mathbf{A} = [\mathbf{A}_1 \ \ \mathbf{A}_2 \ldots \mathbf{A}_R] \in \mathbb{R}^{I \times LR}$, $\mathbf{B} = [\mathbf{B}_1 \ \ \mathbf{B}_2 \ldots \mathbf{B}_R] \in \mathbb{R}^{J \times LR}$ and $\mathbf{C} = [\mathbf{c}_1 \ \ \mathbf{c}_2 \ldots \mathbf{c}_R] \in \mathbb{R}^{K \times R}$. The $(L \times L)$ identity matrix is represented by $I_{L \times L}$. 1_L is a column vector of all ones of length L.

3 PROBLEM FORMULATION

Consider a multi-aspect graph, where each frontal slice of a tensor is a snapshot of the adjacency matrix at a given time point or a different aspect of user relation (e.g., "calls", "text", etc). A typical application here is the extraction of communities in the graph and their evolution over time. Traditionally, one would take a CP decomposition of the tensor, where each rank-one component would map to each community; furthermore, the values on the a_r and b_r vectors (corresponding to the two dimensions of the nodes in our graph) would give the membership of each node to each community. This method has been shown to be very accurate in extracting community memberships [5, 13, 32], however, it is not able to identify the shape of the sub-graph that defines the community. As we argue in the introduction (Sec. 1) of this paper, the reconstructed adjacency matrix of the $r-th$ community will be $a_r b_r^T$ which is a rank-1 block, corresponding to a clique. Therefore, CP imposes clique structure to all communities, which need not be true, and may result in false conclusions. On the other hand, a BTD-(L, L, 1) is able to extract higher rank-1 components in the first two modes, providing flexibility in extracting richer structure in the multi-aspect graph.

The problem that we solve is the following:

Given: a multi-aspect graph given as a higher-order tensor $\underline{\mathbf{X}}$ and $\mathcal{S}$ as structure vocabulary:

- **extract** latent components of $\underline{\mathbf{X}}$ which allow for higher than rank-1 in the first two modes,
- **find** a set of possibly overlapping sub-tensors $\{\underline{\mathbf{T}}_1, \underline{\mathbf{T}}_2 \ldots \underline{\mathbf{T}}_R\}$ with minimal encoded length (average bits) i.e. $\mathcal{B}(\mathcal{S}_i) + \mathcal{B}(Err)$
- to **briefly describe** the given multi-aspect graph communities structure in a efficient and scalable fashion.

4 PROPOSED METHOD: RICHCOM

Given $\underline{\mathbf{X}}$, this section proposes the constrained LL1 decomposition in order to factorize the multi-aspect graph or tensor into its constituent community-revealing factors and provide community structure's encoding formulation. We focus on a third-order tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$ for our problem and its loss function formulation is given by:

$$\mathcal{LS}(\underline{\mathbf{X}}, \mathbf{A}, \mathbf{B}, \mathbf{C}) = \underset{\mathbf{A},\mathbf{B},\mathbf{C}}{\operatorname{argmin}} ||\underline{\mathbf{X}} - \sum_{r=1}^{R} (\mathbf{A}_r \cdot \mathbf{B}_r^T) \circ \mathbf{c}_r||_F^2 \tag{11}$$

The above Eq. (11) in its non-convex optimization form can be solved as:

$$\{\mathbf{A}, \mathbf{B}, \mathbf{C}\} \leftarrow \underset{\mathbf{A},\mathbf{B},\mathbf{C}}{\operatorname{argmin}} \mathcal{LS}(\underline{\mathbf{X}}, \mathbf{A}, \mathbf{B}, \mathbf{C}) + r(\mathbf{A}) + r(\mathbf{B}) + r(\mathbf{C}) \tag{12}$$

where **r(.)** is a penalty or constrained function. Instead of solving (12) for the all three factors at once, we can use least square method by fixing all factor matrices but solve one each at a time. Thus the problem is converted into three coupled linear least-squares sub-problems. A very well-suited optimization framework, that has

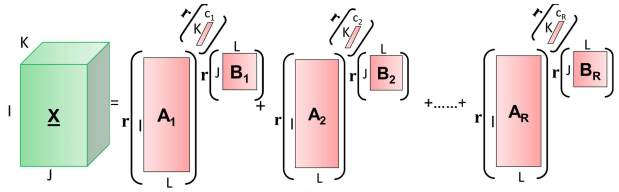

Figure 5: Proposed constrained -(L, L, 1) for a third-order tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$. Here, r(.) represents constraint or penalty function on each block.

shown promise in other, simpler tensor models [1, 16, 39], is the Alternating Method of Multipliers [4], applied in an alternating optimization fashion. In the next subsection we derive and describe our optimization method in detail.

4.1 Solving the cLL1

In the proposed framework, each step consists of fixing two factors and minimizing the sub-problem with respect to the third factor. In this section, we provide solver for tackling the problem efficiently.

4.1.1 ***Factor A update.*** Consider first the update of factor $\mathbf{A} = [\mathbf{A}_1 \ \ \mathbf{A}_2 \ldots \mathbf{A}_R] \in \mathbb{R}^{I \times LR}$ at iteration k, obtained after fixing $\mathbf{B} = \mathbf{B}^{(k-1)}$ and $\mathbf{C} = \mathbf{C}^{(k-1)}$ and solving the corresponding minimization. The arising sub-problem, after manipulation can be re-written as:

$$\mathbf{A}^{(k)} \leftarrow \underset{\mathbf{A}}{\operatorname{argmin}} [(\mathbf{B}^{(k-1)} \odot \mathbf{C}^{(k-1)})^{\dagger} \cdot \underline{\mathbf{X}}_{(1)}]^T \tag{13}$$

where $\underline{\mathbf{X}}_{(1)} = (\mathbf{B} \odot \mathbf{C})A^T$ is a metricized reshaping of the tensor $\underline{\mathbf{X}}$ in mode-1. Also $\mathbf{B}^{(k-1)} \odot \mathbf{C}^{(k-1)} := [(\mathbf{B}_1^{(k-1)} \otimes \mathbf{c}_1^{(k-1)}) \ (\mathbf{B}_2^{(k-1)} \otimes \mathbf{c}_2^{(k-1)}) \ \ldots \ (\mathbf{B}_R^{(k-1)} \otimes \mathbf{c}_R^{(k-1)})]$ is the partition-wise Kronecker product of $\mathbf{B}^{(k-1)}$ and $\mathbf{C}^{(k-1)}$, where $\mathbf{B}_r^{(k-1)}$ denotes partitioned matrix r of $\mathbf{B}^{(k-1)}$ and $\mathbf{c}_r^{(k-1)}$ denotes column r of $\mathbf{C}^{(k-1)}$, and $\otimes$ denotes the Kronecker product operator; see also Def. 6. The regularized version of Eq. 13 is given as:

$$\mathbf{A}^{(k)} \leftarrow \underset{\mathbf{A}_{reg}}{\operatorname{argmin}} [(\mathbf{B}^{(k-1)} \odot \mathbf{C}^{(k-1)})^{\dagger} \cdot \underline{\mathbf{X}}_{(1)}]^T + r(\mathbf{A}^{(k-1)}) \tag{14}$$

Constraints are implemented in such a way that when the constraints are violated then **r(.)** takes the value of infinity (∞), otherwise in normal scenario regularization (e.g. sparsity, l1 etc.) uses finite values to penalize undesirable but reasonable solutions. Following the steps in [39], the primal, an auxiliary and a dual variables as $\mathbf{H} \in \mathbb{R}^{I \times LR}$, $\widetilde{\mathbf{H}} \in \mathbb{R}^{LR \times I}$ and $\mathbf{U} \in \mathbb{R}^{I \times LR}$, respectively are introduced to account for the augmented Lagrangian of Eq. (14). Each iteration optimizes factor $\mathbf{A}$ by means of *AO-ADMM* (given in

algorithm 1) as:

$$\mathcal{L}_A^{(k)}(\mathbf{H},\widetilde{\mathbf{H}},\mathbf{U}) = \underset{\mathbf{H},\widetilde{\mathbf{H}}}{\operatorname{argmin}} ||(\mathbf{Y}_A^{(k)} \cdot \underline{\mathbf{X}}_{(1)})^T||_F^2 + Trace(\mathbf{H}\mathbf{H}^T) + r(\mathbf{H}) + (\rho/2)||\mathbf{H} - \widetilde{\mathbf{H}}^T + \mathbf{U}||_F^2 \quad \text{subject to} \quad \mathbf{H} = \widetilde{\mathbf{H}} \tag{15}$$

where $\mathbf{Y}_A^{(k)} := (\mathbf{B}^{(k-1)} \odot \mathbf{C}^{(k-1)})^\dagger$, and $r(\mathbf{H})$ is the regularizer (e.g. non-negativity, sparsity etc.) constraint on factor $\mathbf{A}$. The Lagrange multiplier ρ is set to minimum value between 10^{-3} and $(||\mathbf{Y}_A||_F^2/LR)$ to yield good performance. The AO-ADMM solver advances further by iteratively updating the variables $\mathbf{H}$, $\widetilde{\mathbf{H}}$,$\mathbf{U}$ until a convergence criterion is met, i.e. whether the prescribed error tolerance is met, i.e., $||\mathbf{A}_{curr}^{(k)} - \mathbf{A}_{prev}^{(k)}||_F / ||\mathbf{A}_{prev}^{(k)}.||_F \leq \epsilon$ or maximum number of iterations are reached.

4.1.2 ***Factor*** **B** ***update***. Update of factor $\mathbf{B} = [\mathbf{B}_1 \ \ \mathbf{B}_2 \ldots \mathbf{B}_R] \in \mathbb{R}^{J \times LR}$ can be similarly obtained by solving the sub-problem as:

$$\mathbf{B}^{(k)} \leftarrow \underset{\mathbf{B} \geq 0}{\operatorname{argmin}} [(\mathbf{C}^{(k-1)} \odot \mathbf{A}^{(k)})^\dagger \cdot \underline{\mathbf{X}}_{(2)}]^T + r(\mathbf{B}^{(k-1)}) \tag{16}$$

, and its *AO-ADMM* solver as :

$$\mathcal{L}_B^{(k)}(\mathbf{H},\widetilde{\mathbf{H}},\mathbf{U}) = \underset{\mathbf{H},\widetilde{\mathbf{H}}}{\operatorname{argmin}} ||(\mathbf{Y}_B^{(k)} \cdot \underline{\mathbf{X}}_{(2)})^T||_F^2 + Trace(\mathbf{H}\mathbf{H}^T) + r(\mathbf{H}) + (\rho/2)||\mathbf{H} - \widetilde{\mathbf{H}}^T + \mathbf{U}||_F^2 \quad \text{subject to} \quad \mathbf{H} = \widetilde{\mathbf{H}} \tag{17}$$

where $\mathbf{H} \in \mathbb{R}^{J \times LR}$, $\widetilde{\mathbf{H}} \in \mathbb{R}^{LR \times J}$ and $\mathbf{U} \in \mathbb{R}^{J \times LR}$ and $\mathbf{Y}_B^{(k)} := (\mathbf{C}^{(k-1)} \odot \mathbf{A}^{(k)})^\dagger := [(\mathbf{c}_1^{(k-1)} \otimes \mathbf{A}_1^{(k)}) \ (\mathbf{c}_2^{(k-1)} \otimes \mathbf{A}_2^{(k)}) \ \ldots \ (\mathbf{c}_R^{(k-1)} \otimes \mathbf{A}_R^{(k)})]^\dagger$, providing a similar optimization problem as in 14. Eq. 17 formulates the update rule for solving (14) and similarly method using a general framework for cLL1 to update factor $\mathbf{B}$.

Proposition 1: If the sequence generated by AO-ADMM in Algorithm 1 is bounded, then the sequence $\{\mathbf{A}(k), \mathbf{B}(k), \mathbf{C}(k)\}$ and AO-ADMM converges to a stationary point of Equ. 12.

Proof: The convergence follows from [[16], Theorem 1; [35], Theorem 2].

4.1.3 ***Factor*** **C** ***update***. Update of factor $\mathbf{C} = [\mathbf{c}_1 \ \ \mathbf{c}_2 \ldots \mathbf{c}_R] \in \mathbb{R}^{K \times R}$ is obtained by fixing $\mathbf{A}$ and $\mathbf{B}$ at their most recent values, and solving it by :

$$\mathbf{C}^{(k)} \leftarrow \underset{\mathbf{C}_{reg}}{\operatorname{argmin}} \{[(\mathbf{A}_1^{(k)} \odot_c \mathbf{B}_1^{(k)})1_{L_1} \ldots (\mathbf{A}_R^{(k)} \odot_c \mathbf{B}_R^{(k)})1_{L_R}]^\dagger \cdot \underline{\mathbf{X}}_{(3)}\}^T + r(\mathbf{C}^{(k-1)}) \tag{18}$$

where $\underline{\mathbf{X}}_{(3)} := [(\mathbf{A}_1 \odot_c \mathbf{B}_1)1_{L_1} \ (\mathbf{A}_2 \odot_c \mathbf{B}_2)1_{L_2} \ldots (\mathbf{A}_R \odot_c \mathbf{B}_R)1_{L_R}] \cdot C^T$. Utilizing an *AO-ADMM* approach, the augmented Lagrangian is represented as:

$$\mathcal{L}_C^{(k)}(\mathbf{H},\widetilde{\mathbf{H}},\mathbf{U}) = \underset{\mathbf{H},\widetilde{\mathbf{H}}}{\operatorname{argmin}} ||(\mathbf{Y}_C^{(k)} \cdot \underline{\mathbf{X}}_{(3)})^T||_F^2 + Trace(\mathbf{H}\mathbf{H}^T) + r(\mathbf{H}) + (\rho/2)||\mathbf{H} - \widetilde{\mathbf{H}}^T + \mathbf{U}||_F^2 \quad \text{subject to} \quad \mathbf{H} = \widetilde{\mathbf{H}} \tag{19}$$

where $\mathbf{H} \in \mathbb{R}^{K \times R}$, $\widetilde{\mathbf{H}} \in \mathbb{R}^{R \times K}$ and $\mathbf{U} \in \mathbb{R}^{K \times R}$ and $\mathbf{Y}_C^{(k)} := (\mathbf{A}^{(k)} \odot \mathbf{B}^{(k)})^\dagger := [(\mathbf{A}_1^{(k)} \otimes \mathbf{B}_1^{(k)})1_{L_1} \ (\mathbf{A}_2^{(k)} \otimes \mathbf{B}_2^{(k)})1_{L_2} \ \ldots \ (\mathbf{A}_R^{(k)} \otimes \mathbf{B}_R^{(k)})1_{L_R}]^\dagger$. The normalization ($c_r \leftarrow c_r/||c_r||$) is applied to each column of obtained factor matrix $\mathbf{C}^{(k)}$ to avoid any underflow or overflow.

Our proposed method readily extends to higher-order tensors, since the underlying tensor model BTD - rank (L,L,1) mathematically extends naturally as such. In this study, we focus on three-mode tensor only for simplicity.

Algorithm 1: RichCom: Discovering Rich Community Structure

Input: $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$, $L \in \mathbb{R}^R$, Max iterations I_{max}.
Output: Factor matrices **A**, **B**, **C**, Structures $\mathcal{S}$.

(**A**, **B**, **C**) ← cLL1($\underline{\mathbf{X}}$, L, I_{max})
$\mathbf{D}_r \leftarrow (\mathbf{A}_r \cdot \mathbf{B}_r^T) \quad \forall r \in R$
$\{Y_{nodes}, Y_{comm}\} \leftarrow communityDetection(\mathbf{D})$
for i = 1 : total communities **do**
  $m \leftarrow Y_{nodes}(find(Y_{comm} == i))$
  $\underline{\mathbf{T}}_i \leftarrow \underline{\mathbf{X}}(m, m, m)$
  $\mathcal{S}_i \leftarrow encode(\underline{\mathbf{T}}_i)$ ▷ **using section 4.2**
end for
Visualize $\mathcal{S}$ ▷ **using section 4.4**
Return (**A**, **B**, **C**, $\mathcal{S}$)

Function cLL1 ($\underline{\mathbf{X}}$, L, I_{max})
  Initialize **A**, **B**, **C** randomly
  $s \leftarrow sum(L)$; $R \leftarrow length(L)$
  $\underline{\mathbf{X}}_{(1)} = tenmat(\underline{\mathbf{X}}, 1)$; $\underline{\mathbf{X}}_{(2)} = tenmat(\underline{\mathbf{X}}, 2)$; $\underline{\mathbf{X}}_{(3)} = tenmat(\underline{\mathbf{X}}, 3)$
  while $k < I_{max}$ or not-convergence **do**
    $\mathbf{G} \leftarrow AA^T$; $\mathbf{Y}_A^{(k)} \leftarrow (\mathbf{B}^{(k-1)} \odot \mathbf{C}^{(k-1)})^\dagger$
    $\mathbf{F} \leftarrow (\mathbf{Y}_A^{(k)} \cdot \underline{\mathbf{X}}_{(1)})^T$; $\rho = min(10^{-3}, (||\mathbf{Y}_A^{(k)}||_F^2/s)$
    $\mathbf{A}^{(k)}, \hat{\mathbf{A}}^{(k)} \leftarrow$ **ADMM**($\mathbf{A}^{(k-1)}$, $\hat{\mathbf{A}}^{(k-1)}$, **F**, **G**, ρ)
    ▷ **Algorithm ADMM step [39]**
    $\mathbf{G} \leftarrow BB^T$; $\mathbf{Y}_B^{(k)} \leftarrow (\mathbf{c}^{(k-1)} \odot \mathbf{A}^{(k)})^\dagger$
    $\mathbf{F} \leftarrow (\mathbf{Y}_B^{(k)} \cdot \underline{\mathbf{X}}_{(2)})^T$; $\rho = min(10^{-3}, (||\mathbf{Y}_B^{(k)}||_F^2/s)$
    $\mathbf{B}^{(k)}, \hat{\mathbf{B}}^{(k)} \leftarrow$ **ADMM**($\mathbf{B}^{(k-1)}$, $\hat{\mathbf{B}}^{(k-1)}$, **F**, **G**, ρ)
    $\mathbf{G} \leftarrow CC^T$; $\mathbf{Y}_C^{(k)} \leftarrow (\mathbf{A}^{(k)} \odot \mathbf{B}^{(k)})^\dagger =$
    $[(\mathbf{A}_1^{(k)} \otimes \mathbf{B}_1^{(k)})1_{L_1} \ (\mathbf{A}_2^{(k)} \otimes \mathbf{B}_2^{(k)})1_{L_2} \ \ldots \ (\mathbf{A}_R^{(k)} \otimes \mathbf{B}_R^{(k)})1_{L_R}]^\dagger$
    $\mathbf{F} \leftarrow \mathbf{Y}_C^{(k)} \cdot \underline{\mathbf{X}}_{(3)})^T$; $\rho = min(10^{-3}, (||\mathbf{Y}_C^{(k)}||_F^2/s)$
    $\mathbf{C}^{(k)}, \hat{\mathbf{C}}^{(k)} \leftarrow$ **ADMM**($\mathbf{C}^{(k-1)}$, $\hat{\mathbf{C}}^{(k-1)}$, **F**, **G**, ρ)
  end while
  Return A, B, C
end Function

4.2 Community Structure Encoding

Once the proposed solver returns the solution of (12), next step is to find communities by extracting weakly connected components [17] from first two dimensions of tensor i.e obtained matrices $D_r = (\mathbf{A}_r \cdot \mathbf{B}_r^T)$ from factors $\mathbf{A}$ and $\mathbf{B}$. Once the communities are formed, we extract the sub-tensors $\underline{\mathbf{T}}_i \in \mathbb{R}^{b \times b \times b}$ from original multi-aspect graph or tensor using nodes falls under each community and we find structure $\mathcal{S}$ such as *FC*: Full Clique; *NC*: Near Clique; *ST*: Star; *CH*: Chain; *CB*: Complete Bipartite; *NB*: Near Bipartite, that best describes its characteristics using below encoding cost (average bits) $\mathcal{B}(\underline{\mathbf{T}})$ with help of J. Rissanen modeling $B_{\mathbb{N}}$ [36]. We observed that these structures appear very frequent, in most of real world data, (e.g in company communication networks, there is possibility of one way interaction that results in star (hub and spoke) structure and in friendship network, most of friends are connected to each other forming cliques or near cliques etc).

4.2.1 ***Cliques and Near Cliques.*** These are the simplest structures, as all the nodes have a similar role. For a full and near cliques, we compute a set of fully-connected or almost fully connected nodes. Consider sub-tensor $\underline{\mathbf{T}} \in \mathbb{R}^{|b|\times|b|\times|b|}$, where $|b|$ is number of nodes fall in the community and $|nz|$ as number of non-zero elements in $\underline{\mathbf{T}}$. Also, consider $|n|$ represents number of nodes having at least two non-zero element in $\underline{\mathbf{T}}$. Now, if $\{|nz| = |b| * |b| * |b|\}$ then sub-tensor is considered as full clique, otherwise between range $\{0.75 * |b| * |b| * |b| \leq |nz| \leq |b| * |b| * |b|\}$, sub-tensor is refereed as near clique and we formalize the average bits to encode the structure as.

$$\mathcal{B}_{\underline{\mathbf{T}}}^{(FC,NC)} = B_{\mathbb{N}}(\underline{\mathbf{T}}) + \log_2({}^{|b|}C_{|n|}) - |nz|\log(|nz|) - |z|\log(|z|) + 3b^3\log(b) \tag{20}$$

where $|z|$ are number of elements not present in $\underline{\mathbf{T}}$. The intuition is that the more sparse a near-clique is, encoding will be cheaper.

4.2.2 ***Star.*** A star is very special case of the structure because of its highly sparse nature and it consists of a single node we call it *hub* connected to at least other two nodes. Consider sub-tensor $\underline{\mathbf{T}} \in \mathbb{R}^{|b|\times|b|\times|b|}$ and we formulate the average bits to encode the structure as :

$$\mathcal{B}_{\underline{\mathbf{T}}}^{(ST)} = B_{\mathbb{N}}(\underline{\mathbf{T}}) + \log_2({}^{|b|-1}C_{|n-1|}) + n\log(n) \tag{21}$$

where $|n|$ represents number of nodes having at least one non-zero element in $\underline{\mathbf{T}}$ and $|b|$ is number of nodes fall in the community.

4.2.3 ***Chain.*** In the chain structure, each node is linked to only one of its adjacent next node and forming a super-diagonal sub-tensor that means it has non-zero elements below, above and at the diagonal position only. Consider sub-tensor $\underline{\mathbf{T}} \in \mathbb{R}^{|b|\times|b|\times|b|}$ and we formulate the average bits to encode the structure as :

$$\mathcal{B}_{\underline{\mathbf{T}}}^{(CH)} = B_{\mathbb{N}}(\underline{\mathbf{T}}) + |n|\log_2(|b|) \tag{22}$$

where $|n|$ represents number of nodes having at least one non-zero diagonal element in $\underline{\mathbf{T}}$ and $|b|$ is number of nodes fall in the community.

4.2.4 ***Bipartite and Near Bipartite.*** A complete bipartite and near-bipartite structure is a sub-tensor whose nodes can be divided into two subsets C_1 and C_2 such that no edge has both endpoints in the same subset. We formulate the average bits to encode the structure as :

$$\mathcal{B}_{\underline{\mathbf{T}}}^{(CB,NB)} = B_{\mathbb{N}}(\underline{\mathbf{C_1}}) + B_{\mathbb{N}}(\underline{\mathbf{C_2}}) + \log_2({}^{|b|}C_{|n_2|}) \log_2({}^{|b|}C_{|n_1|}) + -|nz|\log(|nz|) - |z|\log(|z|) + 3b^3\log(b) \tag{23}$$

where $|nz|$ as number of non-zero elements in $\underline{\mathbf{T}}$, $|z|$ are number of elements not present in $\underline{\mathbf{T}}$, $|n_1|$ and $|n_2|$ represents number of nodes having at least two non-zero element in $\underline{\mathbf{C_1}}$ and $\underline{\mathbf{C_2}}$, respectively.

4.3 Encoding the Error

We encode the errors made by structure $\mathcal{S}$ with regard to $\underline{\mathbf{X}}$ and store the information in two separate encoding matrix $\mathbf{E}^+$ and $\mathbf{E}^-$. The former refers to the area of $\underline{\mathbf{X}}$ that structure $\mathcal{S}$ include and later refer as the area of $\underline{\mathbf{X}}$ that $\mathcal{S}$ does not include. We formulate the average bits to encode the error as:

$$\begin{aligned}\mathcal{B}(\mathbf{E}^+) &= \log_2(|\mathbf{E}^+|) - ||\mathbf{E}^+||\log(|nz|) - ||\mathbf{E}^+||'\log(|z|) + |\mathbf{E}|\log(|b|)\\ \mathcal{B}(\mathbf{E}^-) &= \log_2(|\mathbf{E}^-|) - ||\mathbf{E}^-||\log(|nz|) - ||\mathbf{E}^-||'\log(|z|) + ||\mathbf{E}||\log(|b|)\end{aligned} \tag{24}$$

where $\mathbf{E} = \mathbf{E}^+ + \mathbf{E}^-$. We first encode the number of 1*s* in $\mathbf{E}^+$ and $\mathbf{E}^-$, then followed by sending the actual 1*s* and 0*s* to its optimal prefix codes.

4.4 Visualization of Community Structure

Community structure visualization is a powerful tool to convey the content of a community and can highlight patterns, and show connections among nodes. We developed tool (link[1]) in MATLAB to visualize each community structure. Figure 6 is visualization of a few structures discovered by RichCom that provide summarization with the minimum encoding cost in American college football dataset.

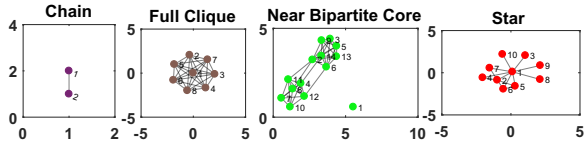

Figure 6: Visualization: a few community structures of Football[37] dataset.

Finally, by putting everything together, we obtain the general version of our RichCom and ADMM solver,as presented in Algorithm 1 and 2, respectively.

Algorithm 2: ADMM solver for Equ. (12).

Input: Residual matrices $\mathbf{R}_H, \mathbf{R}_U, \mathbf{R}_F, \mathbf{R}_G$, and ρ
Output: $\mathbf{R}_H, \mathbf{R}_U$
1: $\mathbf{L} \leftarrow$ Lower Cholesky decomposition$(\mathbf{R}_G + \rho\mathbf{I})$
2: **while** $iter < I_{ADMM}$ or $(r < \epsilon$ and $s < \epsilon)$ **do**
3: $\quad\widetilde{\mathbf{R}}_H \leftarrow (\mathbf{L}^T)^{-1}\mathbf{L}^{-1}(\mathbf{R}_F + \rho(\mathbf{R}_H + \mathbf{R}_U)$
4: $\quad\mathbf{R}_H^0 \leftarrow \mathbf{R}_H$
5: $\quad\mathbf{R}_H \leftarrow \text{argmin}_{\mathbf{R}_H}\, r(\mathbf{R}_H) + Tr(\mathbf{R}_G) + \frac{\rho}{2}||\mathbf{R}_H - \widetilde{\mathbf{R}}_H^T + \mathbf{R}_U||$
6: $\quad\mathbf{R}_U \leftarrow \mathbf{R}_U + \mathbf{R}_H - \widetilde{\mathbf{R}}_H$
7: $\quad r \leftarrow ||\mathbf{R}_H - \widetilde{\mathbf{R}}_H^T||_F^2/||\mathbf{R}_H||_F$
8: $\quad s \leftarrow ||\mathbf{R}_H - \widetilde{\mathbf{R}}_H^0||_F^2/||\mathbf{R}_U||_F$
9: **end while**
Return $(\mathbf{R}_H, \mathbf{R}_U)$

5 EMPIRICAL ANALYSIS

We evaluate the quality, scalability, and real-world utility of RichCom on both real and synthetic datasets. We used MatlabR2016b for our implementations, along with functionalities for tensors from the Tensor-Toolbox [2] and Tensorlab [42].

5.1 Experimental Setup

We provide the datasets used for evaluation in Table 3.

5.1.1 Synthetic Data. In order to fully control and evaluate the community structures in our experiments, we generate synthetic graphs and converted them into multi-aspect graphs or tensor with different cluster density. The code for both generators are available at link[1]. Consider graph or network $G = (\mathcal{V},\mathcal{E})$, represented by node or vertex $(\mathcal{V})$ and relation between entities are defined by

Method	Precision							
	FC	ST	CH	CB	FC	ST	CH	CB
	With Noise				Without Noise			
RichCom	0.93	0.71	0.35	0.71	0.98	0.85	0.75	0.79
VoG[21]	–	–	–	–	0.86	1	0.65	0.0
TimeCrunch [38]	0.58	0.73	0.0	0.0	0.74	0.73	0.0	0.23

Table 2: Result based on structures found correctly in synthetic datasets (higher is better).

weighted or unweighted edges $(\mathcal{E}, \mathcal{W}_{ij})$ s.t. $(\mathcal{V}, \mathcal{E}, \mathcal{W}) \in \mathbb{R}^{N \times N}$. Let $\mathcal{W}_{ij}$ denotes the edge weight if $(i, j) \in \mathcal{E}$, otherwise $\mathcal{W}_{ij} = 0$. Consider $\underline{\mathbf{X}}_n = (\mathcal{V}, \mathcal{E}^{(n)}, \mathcal{W}^{(n)}) \subset G$ denotes the structure constructed as the sub-tensor by subset of nodes $N \in \mathcal{V}$ and third mode as its single and two-hop neighbors. The tensor $\underline{\mathbf{X}}$ captures dependencies between structures in graph G. In a network, the proposed multi-aspect representation is indeed capable of preserving the inherent "structure" among node adjacency along the 3^{rd} mode. We add Gaussian(μ, σ) white noise where the interactions in $\underline{\mathbf{X}}$ are uniformly distributed. This makes recovering the original communities structure increasingly challenging.

5.1.2 Real Data. Table 3 provides a brief description of the real-world datasets from different domains that we use in our experiments.The Football[37] dataset is network of American football games of Fall 2000; European Air Traffic Network(ATN) [19] multi-layer network composed of 37 different layers and each one corresponding to a different airline operating in Europe; Wikipedia[23] consists of users participating in the elections over 7 days and its bipartite in nature; EU-Core[46] consists of e-mail data from a large European research institution over 526 days; Autonomous Systems (AS)-level interconnection[30] consists of multi-aspect of peering information inferred from (a) Oregon route-views, (b) RIPE RIS BGP, (c) Looking glass data, and (d) Routing registry, all combined and Enron[34] consists of a million emails communication over 899 unique dates.

5.1.3 The baseline method. We compare exploration for methods: (a) VoG[21]: finds meaningful patterns in single-layer graphs. We find each structure of $\underline{\mathbf{X}}$ iteratively with it and remove any duplicate structure found for fair analysis and (b) TimeCrunch[38]: find coherent, temporal patterns in dynamic graphs, and evaluate RichCom: our proposed method to discover rich community structures in multi-aspect graphs. It is also *noted* that RichCom does not necessarily compete against [21] and [38], it is an alternative way of achieving the same goal as those method while incorporating joint information from all aspects. Furthermore, the proposed method has direct implications in advancing our ability to analyze general multi-aspect data, where the underlying latent structure is not necessarily rank-1.

Note: Traditional community detection approaches rely on kernel functions, summary graph statistics (e.g., degrees or clustering coefficients) or designed features to extract structural data from graphs for measuring local structures (cliques and lines). However, these approaches are limited because these hand-engineered features are inflexible and designing these features can be time-consuming and expensive. In this work, we only focus on directly related baselines approaches i.e. VoG[21] and TimeCrunch[38]. Thus, even though the extracted sub-structures can be used to identify members of each community, the scope of this work is to identify the structures themselves, and thus we do not compare against community detection methods since the objectives are different.

5.1.4 Estimating Rank R and L. In this work, we deal with two different kinds of Rank a) the rank 'R' of a tensor $\underline{\mathbf{X}}$ and b) each 'r' block having rank - (L, L, 1). Finding the rank R of a tensor $\underline{\mathbf{X}}$ itself is a extremely hard problem [15] and out of the scope of this paper. But we refer the interested reader to previous heuristics studies which try to estimate a low-rank estimation [29, 31] for an overview. In our case, it also requires further research on the relationship between rank R and rank L of each block and it is beyond the scope of this paper. For our work, we set rank R of tensor $\underline{\mathbf{X}}$ as 5 and vary the rank L of block between $10 - 30$, experimental analysis is provided in sub-section 5.2.3.

5.2 Experimental Results

5.2.1 ***Qualitative Analysis.*** In this section, we provide quantitative evaluation and analysis of our method for structure discovery on synthetic and real-world datasets. Table 3 reports the resulting discovered structures.

Synthetic: A network of 50 cliques, 30 stars, 20 chains and 20 bipartite structures is generated and converted to tensor using one-hop neighbor relation with added noise. Discovered structures are presented in Tbl. 3 and quality in terms of finding correct structures is given in Tbl. 2. RichCom and VoG able to detect almost all of the structures (e.g clique, star and bipartite). VoG successfully detected most of the chains, both RichCom and TimeCrunch not able to detect all chain structures because of added noise to multi-aspect format of graph. However, for RichCom, we don't observe a significant drop in discovery of any structure after discarding the noise.
EU-Core[46]: This is a temporal higher-order network dataset of emails between members of the research institution during October 2003 to May 2005 (18 months) and messages can be sent to multiple recipients of 42 departments. Agreeing with hunch, EU-Core consists of a large number of clique and near-clique structures corresponding to many instances of discussion within department. However, we also find numerous re-occurred stars (see Tbl. 3) which indicate email communication between different departments. Interestingly, figure 7 shows researchers for 4 months (corresponding to Oct, 2004 - Jan, 2005) forming continuous near clique structure, consisting of 85 researchers who communicated each month belonging to same department. Suddenly, some members disappeared during Dec'04 and again in Jan'05 communication resumed back normally. We suspect that this may indicate the days corresponding to Festival of Lights, Christmas celebration week and New Year's Eve holiday time.

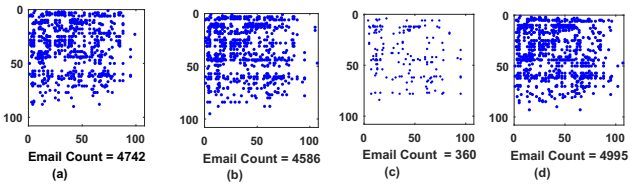

Figure 7: RichCom finds 107 research members in EU-core forming a continuous near clique (≈ 43% density) over the observed last 4 months (a-d). The member's interaction drop (see (c)) indicates the festival month (e.g December).

Dataset	Statistics		RichCom						VoG[21]						TimeCrunch [38]					
	Nodes	|nz|	*FC*	*NC*	*ST*	*CH*	*CB*	*NB*	*FC*	*NC*	*ST*	*CH*	CB	*NB*	*FC*	*NC*	*ST*	*CH*	*CB*	*NB*
Synthetic	$5k$	$0.2m$	54	–	42	7	52	5	58	–	30	13	–	–	29	–	22	–	–	–
Football	115	$8k$	11	10	35	1	–	32	8	–	2	–	6	4	35	–	66	–	–	1
European-ATN	450	$10k$	65	12	217	–	24	48	414	–	–	–	–	–	5	–	206	6	11	–
EU-Core	$1k$	$221k$	329	136	1829	360	31	530	731	–	1471	45	41	218	12	–	1996	499	36	192
Wikipedia	$7k$	$0.4m$	55	71	1373	11	30	97	1515	–	536	13	1	53	108	–	939	77	1	115
AS-level	$13k$	$0.7m$	4	26	1246	32	6	237	2	–	626	26	3	170	2442	–	–	3	3	130
Enron	$32k$	$1.9m$	11	57	1800	48	3	127	9	–	1507	30	2	119	1122	–	126	146	–	107

Table 3: Summarization of community (having > 1 node) structures found in datasets. The most frequent structures are the 'star' and 'near bipartite'. For each dataset, we provide the frequency of each structure type: *'FC'* for full cliques, *'NC'* for near-cliques, *1ST'* for star, *'CH'* for chains, *'CB'* for complete-bipartite, and *'NB'* for near bipartite.

Football[37]: Figure 8 provide visualization of Top-10 community structures discovered by RichCom and we plot these nodes using original football graph and mapped them to ground truth communities provided in literature. Football dataset is characterized by multiple cliques and near (cliques) structures. Interestingly, we found 10 conferences forming near cliques (in literature, total 12 conferences are given as ground truth) and few of the conferences teams had games with other conferences groups that result in formation of near bipartite and star relation.

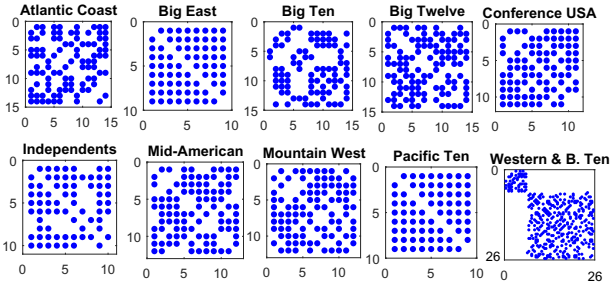

Figure 8: Top 10 structures of football teams found by RichCom.

Figure 8 provide visualization for the structures found by RichCom and we plot these nodes using original football graph and mapped them to ground truth communities provided in literature.

Wikipedia[23]: It is a bipartite graph between users (e.g voter, admins and nominators) participating in the elections. As such, it is characterized by multiple stars and (near) bipartite structures. Many of the voters cast vote to the single user (nominee), as indicated by the presence of 1389 stars and provide the vote as support/neutral/oppose for particular nominee on election week. Interestingly, it is observed that more than half (65%) of these star discovered on the very first day of election (on Sept 14, 2004), indicating the strong support for their favorable nominee. Also, it is observed that about more than half of the votes casted by existing admins and they form near bipartite relation with ordinary Wikipedia users.

Autonomous Systems (AS)[30] : The AS-level dataset is largely comprised of stars and few near clique and bipartite structures. We discover large proportion of stars which occur only at Oregon route-view instance. Further analyzing these results in Fig 9, we find that 985 of the 1246 stars (73%) are found on first instance on Oregon route-view and rest were observed in Looking glass and Routing registry instance. Interestingly, for 2 consecutive weeks, we observed set of routers form (near) clique structure, but later turned into (near) bipartite form indicate operational routers tables changes over time. When a connection between two observed routers on an earlier snapshot disappears from later snapshots, it could be caused either by actual termination or simply by a change in the route server's set of peer routers. The RichCom summary for both

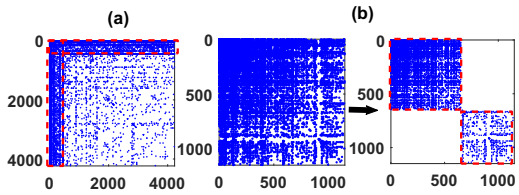

Figure 9: AS-level: Adjacency matrix of the (a) top star (4234 nodes) and (b) near bipartite (624×522 nodes) structure found by RichCom, corresponding to route-view of "traffic flows".

Enron[34] and **European ATN**[19] datasets is very interesting and provided in **Sec 5.3**.

5.2.2 ***Scalability.*** We also evaluate the scalability of the method. We present the run-time (Fig. 10) of RichCom and baseline method TimeCrunch [38] with respect to the number of non-zero elements in the input tensor. For this purpose, we use sub-tensors form of Enron dataset consists of a millions of emails and method is flexible enough to deal over all the layers. Also, we evaluate proposed method on synthetic data with increasing two modes (I and J) of tensor with third mode (K) equivalent to 2% and 20% of I. Fig. 10 shows near linear run-time for million edges. For our experiment we used Intel(R) Xeon(R), CPU E5-2680 v3 @ 2.50GHz machine with 48 CPU cores and 378GB RAM. It is worth noting that since it is a AO-ADMM optimization framework, it is possible to parallelize the implementations, which can enable its feasible adoption for analysis of even larger multi-aspect or time evolving tensors.

5.2.3 ***Parameter Selection*** L ***and*** R. We use synthetic dataset with 20 cliques and 20 stars consisting of 50 nodes in each structure. To evaluate the impact of R, we fixed rank of each block i.e. L. We can see that with higher values of the R, number of cliques and star structure discovered is improved as shown in Figure 11 (a). Also, it is observed that after $R \geq 5$, it become saturated. Similarly, we fixed rank $R = 5$ and vary L. We found that for $L \geq 10$ all structures discovery become stable.

5.2.4 Convergence of RichCom. Here we demonstrate the convergence of Algorithm 1 for cLL1 on three real datasets i.e **Football[37]**, **European ATN**[19] and **EU-Core**[46] network that we use for evaluation. The stabilization of fitness is observed after iteration 33, 48 and 45 for Football, EU-Core and EU-ATN, respectively. Figure 12 summarizes the convergence of the algorithm, showing the approximated fitness as a function of the number of iterations. It is clear that the algorithm converges to a very good approximation within $40 - 50$ iterations. As our method is developed based on

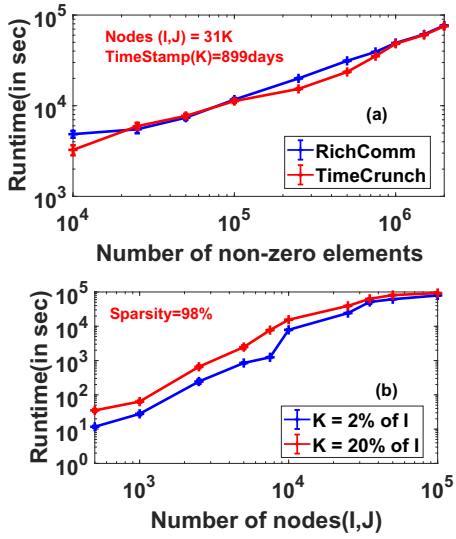

Figure 10: RichCom scales well on (a) induced aspects on third mode of Enron[27] dataset, up to 2M non-zero elements in size, and (b) on synthetic data varying two modes of tensor (I, J), up to 20M non-zero elements in size.

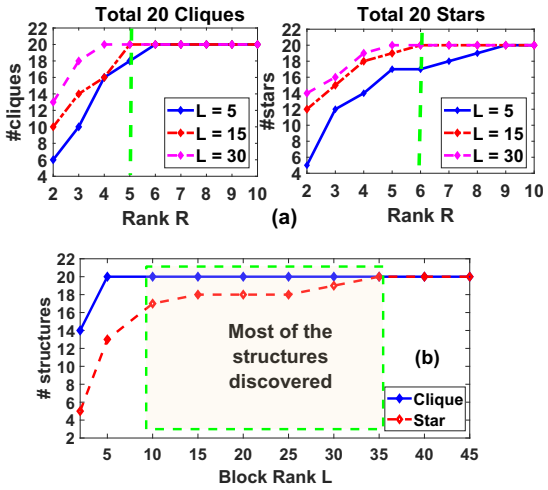

Figure 11: RichCom performance on synthetic dataset for (a) varying R and constant L, and (b) vary L and constant $R = 5$.

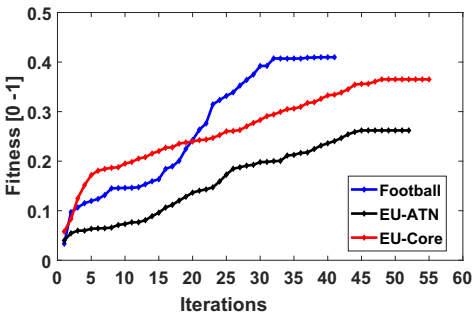

Figure 12: Fitness vs. number of iterations. For each dataset, computation cost was average 47 sec/iteration.

accelerated AO-ADMM approach [Razaviyayn et al. [35], Huang et al. 2016 [16] and Smith et al. 2017 [39]], it follows the same convergence rules and the proof follows from that work.

5.3 RichCom at Work

Beyond our qualitative analysis of the structure discovery in the six real dataset in Tbl. 3, we also consider a sample community structure analysis from the two datasets in Fig. (2 , 13), and present our findings in this section. **Case Study 1**: The European air traffic network can be represented as a graph composed of $K = 37$ different layers or aspects each representing a airline (e.g Lufthansa, KLM etc). Each layer k has the same number of nodes, $|I| = 450$, as all European airports (e.g London Heathrow, Zurichi Kloten Airport etc) are represented in each layer. RichCom extracted two structures i.e. 'star' (Fig 2(b)) and 'cliques' (Fig 2(c)) frequently from this dataset, layers comprising in particular, major national airlines (e.g Lufthansa→ 1, Finnair→ 4 and KLM→ 9), low-cost fares (e.g Ryanair → 2, Easyjets→ 3), regional (e.g Olympic Air → 37, Aegean Airlines → 30) or cargo airlines like Fed-Ex. Figure 2(a) show, instead, the single-layer ATN corresponding to a given major,regional and low-cost airlines reconstructed from decomposed factors using cLL1. These types of airlines have developed based on different commercial and structural constraints. As an example, it is well known that national airlines are designed as star network and ensuring the **hub** and **spoke** structure, to provide an almost full coverage of the airports belonging to a given country and maximize efficiency in terms of national or governmental transportation interests. On other hand low-cost airlines try to avoid this unify structures and, to be more aggressive, generally cover more than one country simultaneously, results in formation of near clique structures. One of the other reasons is that low-cost airlines stay away from busy and expensive hubs. They manage to operate from smaller airports through which ground times and delays are reduced that lead to cost reduction. The result of this study show that RichCom successfully exploit the multi aspect nature of data to discover the various useful structures.

Case Study 2: We use four years (1999-2002) of Enron email communications. In each view, the nodes represent email addresses and edges depict sent/received/cc relations. This network contains a total of $\approx 32k$ unique email addresses used only for the communication inside organization; and we analyze the data over 899 days including weekends also. The RichCom captures structures formed and changed during the major events in the company's history, such as revenue losses, CEO changes, etc. The interesting case is the constant increase in star structures between Jan and April 2001, this abnormal increase was an indicator of cover up of accounting fraud happens at that time; (see Fig. 13(a)). Jeffrey Skilling (CEO) and Kenneth Lay (Chairman) were conducting regular meeting with their top executives (e.g. Sally Beck (Chief Operating Officer), Vincent Kaminski (Quantitative Modeling Group Head), Darren Farmer (Logistics Manager), Michelle Lokay (Administrative Assistant), Louise Kitchen (Enron-online President),Williams III (Senior Analyst), Tim Belden (Trading Head) and Richard Sanders (Assistant General Counsel)), forming star structure (see Figure 13(b)), in order to find new ways to handle Enron's liability. Second observation from structure is shown in Figure 13(c) where Tim Belden's email address used to send emails to the Enron's World Trade center Office: 'center.dl-portland@enron.com' during time period of

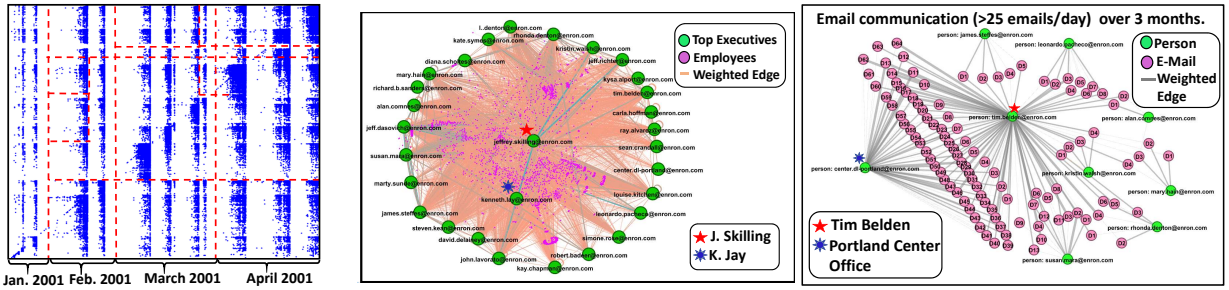

Figure 13: (a) Formation of star (include top executives only) structures over period of Jan to April 2001 (b) the most informative 'star' structure describing interactions between CEO and top executives during accounting fraud of Enron. (b) the second most informative star structure - email communication between Tim Belden and Enron's World Trade center office.

Oct - Dec 2001. But he also send a few emails to other employees inside the company as well. This would ensure that RichCom discovers most relevant people as central hub and further analysis could reveal more interesting patterns.

What sets RichCom apart: none of the state-of art method meet all the following specifications (which RichCom does): (a) consider multi-aspect graphs or tensors, (b) discover efficient communities using tensor decomposition approach and, (c) provide summarization of obtained community's structure by leveraging higher-order correlations between different aspects (either over time or over different views/aspects) to inform the extraction.

6 RELATED WORK

In this section, we provide review of the work related to our algorithm. At large, in the literature this work can be categorized into two main categories as described below:

Multi-aspect Community Detection: Real data usual exhibit different cross-domain relations and can be represented as multi-aspect graphs. In [3] the authors introduce a graph theoretic based community detection algorithm over multi-aspect graphs and relationships between nodes represented by various types of edges. In [13, 32] the authors introduce a robust algorithm for community detection on multi-view graphs based on tensor decomposition which uses a regularized CP model with sparsity penalties. In addition to different graph views, "time" is also a multi-aspect feature of a graph. Finally, most recently in [12] the authors propose a method for identifying and tracking dynamic communities in time-evolving networks. None of these works summarize in terms of local structures; our work focuses on interpretable community structures.

Static/Dynamic Graph Summarization: Graph based method like [40] and [45] uses structural equivalence and minimum DFS code, respectively, to simplify single graph representation to obtain a compressed smaller graph. VoG [21] uses minimum descriptive length to label sub-graphs in terms of a vocabulary including cliques, stars, chains and bipartite cores on static graphs. Further, TimeCrunch [38] extends [21] for dynamic graphs and it uses MDL to label and stitch the dynamic sub-graphs. Here, compression is not our aim. Our work proposes AO-ADMM based constrained Block Term-(L, L, 1) decomposition for multi-aspect graphs to label community structures and provides an effective and stable algorithm for discovering them.

Tensor Decomposition and Optimization: Besides well known CP and Tucker decomposition, the Block Term Decomposition aka BTD was presented in the 3-segment papers [8–10]. BTD provides a tensor decomposition in a sum of Tucker terms. The paper [7] use BTD in modeling process for better exploitation of the spatial dimension of fMRI images. In literature, well-suited optimization framework namely Alternating Method of Multipliers, that has shown promise in other, simpler tensor models [1, 16, 39] was introduced to speed up tensor decomposition process.

7 CONCLUSIONS

We proposed RichCom and cLL1, a novel constrained (L, L, 1) based framework to learn and encode the community structures. The performance of the proposed method is assessed via experiments on synthetic as well as six real-world networks.

Take-home points:

- cLL1 is novel constrained LL1-tensor decomposition method, and alternating optimization with alternating direction method of multipliers (AO-ADMM) is developed to recover the non-negative and sparse factors. The utility of cLL1 extends beyond multi-aspect graphs to general multi-aspect data mining, where the underlying latent structure is richer than rank-1.
- Through experimental evaluation on multiple datasets, we show that cLL1 provides stable decompositions and offering high quality structure via RichCom within reasonable run time, in the presence of overlapping and non-overlapping communities.
- Utility: we provide a simple and effective multi-aspect graph generator.

There are several items that can be considered for future work. First, as a natural extension, one can generalize this to account for higher order (> 3 modes) data. Second, AO-ADMM admits parallel extensions, which can enable the exploration of billion-scale multi-aspect graphs .

8 ACKNOWLEDGEMENTS

Research was supported by an Adobe Data Science Research Faculty Award, the Department of the Navy, Naval Engineering Education Consortium under award no. N00174-17-1-0005, and the National Science Foundation Grant no. 1901379. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding parties.

REFERENCES

[1] Ardavan Afshar, Ioakeim Perros, Evangelos E Papalexakis, Elizabeth Searles, Joyce Ho, and Jimeng Sun. 2018. COPA: Constrained PARAFAC2 for Sparse & Large Datasets. *arXiv:1803.04572* (2018).

[2] Brett W Bader, Tamara G Kolda, et al. 2015. MATLAB Tensor Toolbox Version 2.6, Available online, February 2015.

[3] Michele Berlingerio, Michele Coscia, and Fosca Giannotti. 2011. Finding redundant and complementary communities in multidimensional networks. In *Proc.of the 20th ACM Int. Conf. on Information and knowledge management*. ACM, 2181–2184.

[4] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning* (2011), 1–122.

[5] Bokai Cao, Lifang He, Xiangnan Kong, S Yu Philip, Zhifeng Hao, and Ann B Ragin. 2014. Tensor-based multi-view feature selection with applications to brain diseases. In *Data Mining (ICDM), 2014 IEEE Int. Conf. on*. IEEE, 40–49.

[6] J Douglas Carroll and Jih-Jie Chang. 1970. Analysis of individual differences in multidimensional scaling via an N-way generalization of âĂIJEckart-YoungâĂİ decomposition. *Psychometrika* (1970), 283–319.

[7] Christos Chatzichristos, Eleftherios Kofidis, Yiannis Kopsinis, Manuel Morante Moreno, and Sergios Theodoridis. 2017. Higher-order block term decomposition for spatially folded fMRI data. In *Int. Conf. on Latent Variable Analysis and Signal Separation*. Springer.

[8] Lieven De Lathauwer. 2008. Decompositions of a higher-order tensor in block terms–Part I: Lemmas for.... In *SIAM J. Matrix Anal. Appl.* Citeseer.

[9] Lieven De Lathauwer. 2008. Decompositions of a higher-order tensor in block termsâĂŤPart II: Definitions and uniqueness. *SIAM J. Matrix Anal. Appl.* (2008), 1033–1066.

[10] Lieven De Lathauwer and Dimitri Nion. 2008. Decompositions of a higher-order tensor in block termsâĂŤPart III: Alternating least squares algorithms. *SIAM journal on Matrix Analysis and Applications* (2008), 1067–1083.

[11] Santo Fortunato. 2010. Community detection in graphs. *Physics reports* (2010).

[12] Alexander Gorovits, Ekta Gujral, Evangelos E Papalexakis, and Petko Bogdanov. 2018. LARC: Learning Activity-Regularized Overlapping Communities Across Time. In *Proceedings of the 24th ACM SIGKDD Int. Conf. on KDD*. ACM, 1465–1474.

[13] Ekta Gujral and Evangelos E Papalexakis. 2018. SMACD: Semi-supervised Multi-Aspect Community Detection. In *Proceedings of the 2018 SIAM Int. Conf. on SDM*. SIAM, 702–710.

[14] R.A. Harshman. 1970. Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multimodal factor analysis. (1970).

[15] Johan Håstad. 1990. Tensor rank is NP-complete. *Journal of Algorithms* (1990), 644–654.

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

[17] MatLab Inc. [n. d.]. Matlab https://www.mathworks.com. https://www.mathworks.com Version MATLAB R2016b.

[18] Guanbo Jia, Zixing Cai, Mirco Musolesi, Yong Wang, Dan A Tennant, Ralf JM Weber, John K Heath, and Shan He. 2012. Community detection in social and biological networks using differential evolution. In *Learning and Intelligent Optimization*. Springer.

[19] Jungeun Kim and Jae-Gil Lee. 2015. Community detection in multi-layer graphs: A survey. *ACM SIGMOD Record* (2015), 37–48.

[20] T.G. Kolda and B.W. Bader. 2009. Tensor decompositions and applications. *SIAM review* (2009).

[21] Danai Koutra, U Kang, Jilles Vreeken, and Christos Faloutsos. 2015. Summarizing and understanding large graphs. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 8, 3 (2015), 183–202.

[22] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. 2007. The dynamics of viral marketing. (2007).

[23] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*. ACM, 641–650.

[24] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[25] Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. 2009. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics* (2009), 29–123.

[26] Jure Leskovec, Kevin J Lang, and Michael Mahoney. 2010. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th Int. Conf. on WWW*. ACM, 631–640.

[27] Jure Leskovec and Julian J Mcauley. 2012. Learning to discover social circles in ego networks. In *Advances in neural information processing systems*. 539–547.

[28] Na Li. 2013. *Variants of ALS on Tensor Decompositions and Applications*. Ph.D. Dissertation. Clarkson University.

[29] Morten Mørup and Lars Kai Hansen. 2009. Automatic relevance determination for multi-way models. *A Journal of the Chemometrics Society* 23 (2009).

[30] University of UMICH. [n. d.]. http://topology.eecs.umich.edu/data.html. http://topology.eecs.umich.edu/data.html

[31] Evangelos E Papalexakis. 2016. Automatic unsupervised tensor mining with quality assessment. In *Proceedings of the 2016 SIAM SDM*. SIAM, 711–719.

[32] Evangelos E Papalexakis, Leman Akoglu, and Dino Ienco. 2013. Do more views of a graph help? Community detection and clustering in multi-graphs.. In *FUSION*. Citeseer, 899–905.

[33] Evangelos E Papalexakis, Christos Faloutsos, and Nicholas D Sidiropoulos. 2016. Tensors for data mining and data fusion: Models, applications, and scalable algorithms. *ACM Transactions on Intelligent Systems and Technology (TIST)* (2016).

[34] Shebuti Rayana and Leman Akoglu. 2016. Less is more: Building selective anomaly ensembles. *ACM Transactions on Knowledge Discovery from Data* (2016).

[35] Meisam Razaviyayn, Mingyi Hong, and Zhi-Quan Luo. 2013. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization* 23, 2 (2013), 1126–1153.

[36] Jorma Rissanen. 1983. A universal prior for integers and estimation by minimum description length. *The Annals of statistics* (1983), 416–431.

[37] Polina Rozenshtein, Nikolaj Tatti, and Aristides Gionis. 2014. Discovering dynamic communities in interaction networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 678–693.

[38] Neil Shah, Danai Koutra, Tianmin Zou, Brian Gallagher, and Christos Faloutsos. 2015. Timecrunch: Interpretable dynamic graph summarization. In *Proceedings of the 21th ACM SIGKDD Int. Conf. on KDD*. ACM, 1055–1064.

[39] Shaden Smith, Alec Beri, and George Karypis. 2017. Constrained tensor factorization with accelerated AO-ADMM. In *2017 46th International Conference on Parallel Processing (ICPP)*. IEEE, 111–120.

[40] Hannu Toivonen, Fang Zhou, Aleksi Hartikainen, and Atte Hinkka. 2011. Compression of weighted graphs. In *Proceedings of the 17th ACM SIGKDD Int. Conf. on KDD*. ACM, 965–973.

[41] L.R. Tucker. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika* (1966), 279–311.

[42] N Vervliet, O Debals, L Sorber, M Van Barel, and L De Lathauwer. 2016. Tensorlab 3.0, available online, Mar. 2016. *URL: http://www. tensorlab. net* (2016).

[43] Ulrike Von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and computing* (2007).

[44] Xiaogang Wang, Xiaoxu Ma, and W Eric L Grimson. 2009. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *IEEE Transactions on pattern analysis and machine intelligence* (2009), 539–555.

[45] Xifeng Yan and Jiawei Han. 2002. gspan: Graph-based substructure pattern mining. In *ICDM 2003. Proceedings. 2002 IEEE Int. Conf. on*. IEEE, 721–724.

[46] Hao Yin, Austin R Benson, Jure Leskovec, and David F Gleich. 2017. Local higher-order graph clustering. In *Proceedings of the 23rd ACM SIGKDD Int. Conf. on KDD*. ACM, 555–564.