Artificial Intelligence Review (2021) 54:2891–2916
https://doi.org/10.1007/s10462-020-09916-4

Tensor decomposition for analysing time-evolving social networks: an overview

Sofia Fernandes[1] · Hadi Fanaee-T[2] · João Gama[1]

Published online: 30 September 2020

Abstract
Social networks are becoming larger and more complex as new ways of collecting social interaction data arise (namely from online social networks, mobile devices sensors, ...). These networks are often large-scale and of high dimensionality. Therefore, dealing with such networks became a challenging task. An intuitive way to deal with this complexity is to resort to tensors. In this context, the application of tensor decomposition has proven its usefulness in modelling and mining these networks: it has not only been applied for exploratory analysis (thus allowing the discovery of interaction patterns), but also for more demanding and elaborated tasks such as community detection and link prediction. In this work, we provide an overview of the methods based on tensor decomposition for the purpose of analysing time-evolving social networks from various perspectives: from community detection, link prediction and anomaly/event detection to network summarization and visualization. In more detail, we discuss the ideas exploited to carry out each social network analysis task as well as its limitations in order to give a complete coverage of the topic.

Keywords Social networks · Tensor decomposition · Time evolution · Network analysis

1 Introduction

Evolving networks consist of sets of entities (such as people), interacting with each other over time. These networks are dubbed as social when the interactions exhibit social (non-random) patterns such as the small world effect (Milgram 1967), the skewed degree distribution (Barabási and Albert 1999; Faloutsos et al. 1999), the densification

✉ Sofia Fernandes
sdsf@inesctec.pt

Hadi Fanaee-T
hadi.fanaee@hh.se

João Gama
jgama@fep.up.pt

[1] LIAAD, INESC TEC, University of Porto, Porto, Portugal

[2] Center for Applied Intelligent Systems Research (CAISR), Halmstad University, Halmstad, Sweden

power law and the shrinking diameters (Leskovec et al. 2005). Briefly, the small world effect states that, on average, only five intermediates are needed to link two random individuals by following an "acquaintance of" sequential path. This is also known as six degree separation. Recent work has provided evidence that this distance is even smaller—4 intermediates (Kwak et al. 2010). Barabási and Albert (1999) and Faloutsos et al. (1999) provided empirical evidence that the node degree distribution in this type of networks was right-skewed, that is, there were several nodes with low degree and few with high degree. Finally, in Leskovec et al. (2005), the authors found that the number of edges grows superlinearly in the number of the nodes over time and, consequently, the average degree grows over time. This property is referred to as densification power law. On the other hand, the authors observed that the distance between vertexes decreased over time. In other words, the number of paths between two nodes increased over time and, as a consequence, the size of the shortest path between them decreases over time.

It is noteworthy that the dynamics level may differ according to the network type. For example, it is expected that a friendship network grows slowly. However, in phone calls or email networks, the interactions may vary in each time period considered. In other words, a friendship link at a given time t is expected to hold also at time $t + 1$, nonetheless, a phone call delivered at time t may not be repeated at time $t + 1$. Examples of social networks are presented in Table 1.

Analysing this type of networks defines the scope of social network analysis (SNA). In general terms, the SNA allows the discovery of the underlying interaction patterns such as communities of highly interacting entities or entities/timestamps in which the behaviour strongly deviates from the remaining.

In this field, researchers and companies are interested in methods to analyse the networks in real time with few storage cost. Such analysis has applications not only related to advertisement (Tang et al. 2009) but also to biology (Pavlopoulos et al. 2011), epidemics (Bauer and Lizier 2012; Keeling and Eames 2005) and other domains.

In a biological context, protein–protein interaction networks have been studied using social network analysis allowing the discovery of connection patterns associated with diseases (Goñi et al. 2008; Jordán et al. 2012).

Table 1 Examples of social networks

Network type	Example	#Nodes	#Edges
Friendship	High school friendship (Mastrandrea et al. 2015)	134	406
	Facebook friendship (Leskovec and Mcauley 2012)	4039	88,234
Online media	Following in Twitter (Boldi and Vigna 2004)	41,652,230	1,468,364,884
	Wall-posts on Facebook (Viswanath et al. 2009)	63,891	522,118
Email exchange	Enron (Priebe et al. 2005)	184	10,394
	Radoslaw (Michalski et al. 2011)	49	11,816
Co-authorship	DBLP (Ley 2002)	1,314,050	1,898,661
	Hepth (Leskovec et al. 2005)	22,908	2,673,133
Human contact	Contacts in a workplace (Génois et al. 2015)	92	9827
	Hospital ward contacts (Vanhems et al. 2013)	75	32,424
	Infectious SocioPatterns (Isella et al. 2011)	10,970	52,761
Co-purshasing	Amazon (Leskovec et al. 2007)	334,863	925,872

Moreover, in Xu and Chen (2005), social network analysis is presented as a promising tool for fighting crime. The idea consists of using crime incident data to derive the criminals interactions and their interaction patterns and, consequently, be able of take a more effective action. This research direction has been further considered to find criminal organizations in mobile phone networks (Ferrara et al. 2014) and the Web (Chen et al. 2008).

From an economic point of view, the financial market may also be regarded as a social network in which the social component arises from the investors "chain" response to changes in the market (Bak et al. 1996). In this context, the stock market may be described as a correlation network of financial asset returns and SNA can be applied for identifying and characterizing crises periods (Fernandes et al. 2019). With such information, there is an increasing chance of predicting changes in the stock market (Billio et al. 2012; Brunetti et al. 2019). Additionally, SNA is also used to identify the roles of the corporations in the economy (Heiberger 2018).

It should be noted that, since these social networks evolve over time, the patterns observed now may change in the future and, consequently, real time analysis is of great importance. Moreover, as previously stated, in practice, the size of real-world networks is large whereby computational complexity of the methods becomes a challenge. Thus, analysing a large network in a scalable and efficient way is one of the main challenges in SNA.

In time-evolving networks, entities interact with each other over time. Therefore, there is a natural *nodes* × *nodes* × *time* three-way representation of these networks. Currently, in the literature we can find some methods that explicitly model the time dimension and others that do not. From the latter, one of the existing research directions consists of considering a matrix representation of the dynamic network. In more detail, each network state is described by an adjacency or node/egonet features matrix and the sequence of matrices is further transformed into a single matrix. In this context, matrix decomposition methods have been applied on the matrix resulting from concatenating the adjacency matrices and from (weightily) collapsing the adjacency matrices. These modellings have been considered, respectively, for structural summarization (Tsalouchidou et al. 2018) and link prediction purposes (Dunlavy et al. 2011).

Another matrix perspective is to view the sequence of adjacency matrices as a time-series. The idea consists of (implicitly) incorporating the already observed network structure into the matrix decomposition model of the incoming timestamp. In this context, Mankad and Michailidis (2013) sequentially decompose the incoming network states by imposing smoothness constraints on the factor matrices (so that the factor matrices do not change abruptly from one timestamp to the following). A similar idea is exploited in Ma and Dong (2017) by combining the current network state adjacency matrix with the previous observed community structure before applying matrix decomposition.

While these matrix decomposition-based approaches are able to manage time-evolving networks, they may not be able to capture the nodes temporal dynamics (Fanaee-T and Gama 2016a), an issue that can be tackled by considering tensor decomposition techniques.

To the best of our knowledge, there is no survey or review study in literature focusing on the application of tensor decomposition to SNA. Therefore, the contribution of this work is to provide an overview on the existing literature on the topic. We demonstrate how the tensor decomposition can be used for exploratory analysis and how it can be exploited to tackle major SNA tasks.

Provided that the scope of this work is the application of tensor decomposition to the analysis of dynamic social networks, the reader may refer to Tabassum et al. (2018) for a further overview of methods for static SNA. For an extensive literature review on specific tasks, the reader can refer to Fortunato (2010) and Rossetti and Cazabet (2018) for community detection, (Wang et al. 2015) for link prediction, (Akoglu et al. 2015; Ranshous et al. 2015; Fanaee-T and Gama 2016b) for anomaly detection and (Liu et al. 2018) for graph summarization.

The article is organized as follows: in Sect. 2, we introduce the tensor background required to understand the methodologies exposed. In Sect. 3, we overview the SNA methods which resort to tensor decomposition techniques. At a first stage, we approach tensor decomposition as a general analysis framework, then we focus on the advances attained for each particular SNA task. Finally, we conclude by stating the current limitations/gaps and possible future research directions.

2 Tensors

Tensors are often referred to as high order generalizations of matrices, since they can model more than the two dimensions captured in matrices. In this context, they are also regarded as multi-dimensional arrays: $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times \cdots \times N_M}$ where $N_1 \times N_2 \times \cdots \times N_M$ is the size of the tensor and M is the number of dimensions, also referred to as order or number of modes (Sun et al. 2006).

From a practical point of view, rearranging the tensor as a matrix may facilitate its manipulation and processing. This is the idea of matricization, also called unfolding. In more detail, the mode-d matricization of a tensor $\mathcal{X}$ is the matrix rearrangement of the tensor whose rows are obtained by fixing each index of mode-d and varying the other modes indexes. The result is denoted as $\mathbf{X}_{(d)} \in \mathbb{R}^{N_d \times (\prod_{i \neq d} N_i)}$. The inverse operator is referred to as folding.

One of the applications of matricization and its inverse operation occurs when multiplying a tensor by a matrix. Formally such operation is called mode-d product and involves a tensor $\mathcal{X}$ and a matrix $\mathbf{U}$ that shares dimension d with the tensor $\mathcal{X}$. Thus, assuming $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times \cdots \times N_M}$, then the mode-$d$ product is defined for matrices $\mathbf{U} \in \mathbb{R}^{N_d \times R}$ (Kolda and Bader 2009). It is denoted as $\mathcal{X} \times_d \mathbf{U}$. As it will be discussed in the following sections, this operator has a key role in the decomposition of tensors.

The Frobenius norm of a tensor is defined as

$$||\mathcal{X}|| = \sqrt{\sum_{i_1=1}^{N_1} \sum_{i_2=1}^{N_2} \cdots \sum_{i_M=1}^{N_M} \left[\mathcal{X}(i_1, i_2, \ldots, i_M)\right]^2}.$$

A summary of the tensor notation is presented in Table 2. For the sake of simplicity, in the following sections we assume the tensors have only three dimensions. However, the methods are also applicable for higher order tensors.

2.1 Tucker decomposition

Introduced in 1927 by Hitchcock (1927) and later named after Tucker's work (Tucker 1966), Tucker decomposition is one of the most popular approaches. Briefly, it consists of decomposing a tensor in the (mode) product of a core tensor and a projection/factor matrix for each mode.

Table 2 Tensors notation summary

Symbol	Description
$\circ$	Vector outer product
$\mathbf{x}$	Vector (bold lower case)
$\mathbf{x}(i)$	ith entry of vector $\mathbf{x}$
$\mathbf{X}$	Matrix (bold upper case)
$\mathbf{X}(i,j)$	Entry (i,j) of matrix $\mathbf{X}$
$\mathbf{X}(i,:)$	ith row of matrix $\mathbf{X}$
$\mathbf{X}(:,j)$	jth column of matrix $\mathbf{X}$
$\mathcal{X}$	Tensor
$\mathcal{X}(i_1,\dots,i_M)$	Entry $(i_1,\dots,i_M)$ of tensor $\mathcal{X}$
$\mathbf{X}_{(d)}$	Mode-d matricization of tensor $\mathcal{X}$
$\lvert\lvert\mathcal{X}\rvert\rvert$	Frobenius norm of tensor $\mathcal{X}$

Formally, given a tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ and the core tensor sizes $\{R_1, R_2, R_3\}$, the problem consists of finding a core tensor $\mathcal{Y} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ and factor matrices $\mathbf{A} \in \mathbb{R}^{I \times R_1}, \mathbf{B} \in \mathbb{R}^{J \times R_2}, \mathbf{C} \in \mathbb{R}^{K \times R_3}$ such that:

$$||\mathcal{X} - \mathcal{Y} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}|| \tag{1}$$

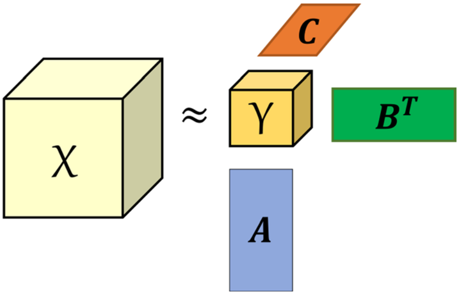
is minimized. Tensor $\mathcal{Y} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$ is interpreted as an approximation of $\mathcal{X}$, and therefore is denoted as $\hat{\mathcal{X}}$. Elementwise,

$$\hat{\mathcal{X}}(i,j,k) = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} \mathcal{Y}(r_1, r_2, r_3)\mathbf{A}(i, r_1)\mathbf{B}(j, r_2)\mathbf{C}(k, r_3).$$

The decomposition result is illustrated in Fig. 1.

It is noteworthy that the quality of the tensor decomposition approximation depends on the number of factors used in each mode, that is, on the values of R_1, R_2 and R_3. In this context, the DIFFerence in Fit (DIFFIT) (Timmerman and Kiers 2000; Kiers and Kinderen 2003) is the most popular method to estimate such values. The method is built on two key ideas: first, given two models with the same "complexity", by which we mean equal $\sum_{i=1}^{M} R_i$, the best number of factors is the one associated with the model with the highest fitting rate, given by

Fig. 1 Illustration of the output of the Tucker decomposition of a three order tensor



$$fit_{\%} = \left(1 - \frac{||\mathcal{X} - \hat{\mathcal{X}}||}{||\mathcal{X}||}\right) \times 100.$$

Then, given the best model associated with each complexity, the number of factors should be set so that an increase in the complexity of the model does not lead to a considerable increase on the fitting of the model. The goal of this method is to find a trade off between model complexity and fitting rate. More recent approaches include Numerical Convex Hull (NumConvHull) (Ceulemans and Kiers 2006) and Automatic Relevance Determination (ARD) (Mørup and Hansen 2009).

2.2 CANDECOMP/PARAFAC decomposition

CANonical DECOMposition (CANDECOM) (Carroll and Chang 1970), also called PARAllel FACtors (PARAFAC) (Harshman 1970) or simply CP (Kiers 2000) is another well known decomposition method in which the tensor is decomposed into the sum of rank-one tensors, that is, into the sum of tensors which can be obtained from the outer product of a single vector for each mode.

More formally, in CP decomposition, given a tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ and the number of factors R (which is common to all modes), the problem consists of finding factor matrices $\mathbf{A} \in \mathbb{R}^{I \times R}, \mathbf{B} \in \mathbb{R}^{J \times R}, \mathbf{C} \in \mathbb{R}^{K \times R}$ such that:

$$\left\| \mathcal{X} - \sum_{r=1}^{R} \mathbf{A}(:,r) \circ \mathbf{B}(:,r) \circ \mathbf{C}(:,r) \right\| \tag{2}$$

is minimized. In this model,

$$\hat{\mathcal{X}}(i,j,k) = \sum_{r=1}^{R} \mathbf{A}(i,r)\mathbf{B}(j,r)\mathbf{C}(k,r).$$

In Fig. 2, the output of CP decomposition is illustrated.

When compared with Tucker, CP is a less complex model (as it requires less parameters—the number of factors is the same for all dimensions). Moreover, it is often chosen for its interpretability (Papalexakis et al. 2016).

Similarly to Tucker decomposition, heuristics have been proposed to estimate the most appropriate number of factors in CP. CORe CONSistency DIAgnostic (CORCONDIA) (Bro and Kiers 2003; Papalexakis 2016) is one of the most well known proposed methods. Briefly, CORCONDIA is built on the following assumption: if the CP decomposition model captures the structure of the original tensor then it is expected that when applying Tucker decomposition using the same number of components, the core tensor obtained is close to superdiagonal. Based on this, the quality of the model is measured based on the non-zero non-diagonal entries of the core tensor (higher values are associated with lack of fitness). Other methods include the Minimum Description Length (MDL) approach (Liu et al. 2016) and the adaptations of the methods designed for estimating the number of factors in Tucker decomposition.

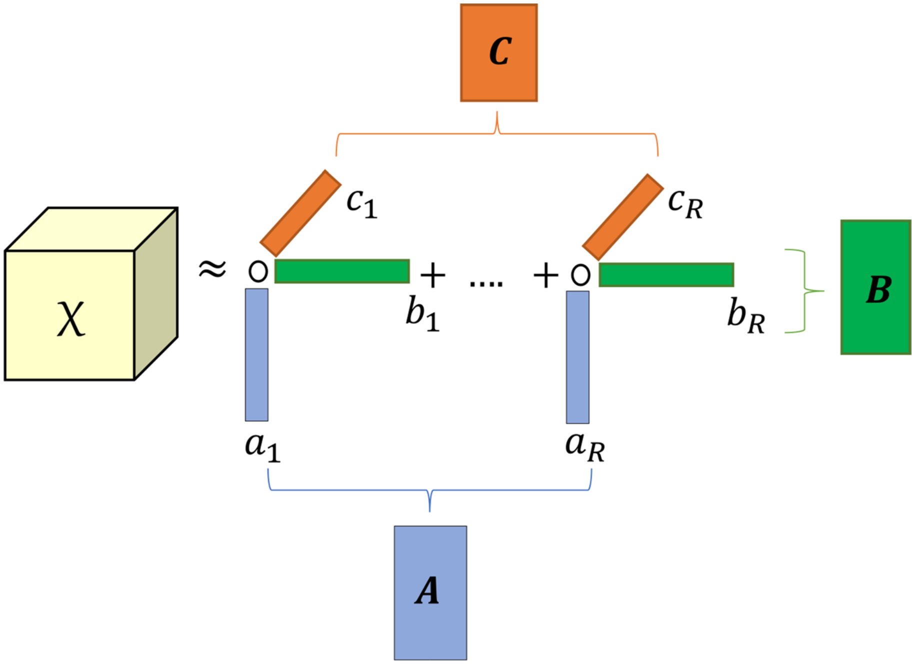


Fig. 2 Illustration of the output of the CP decomposition of a three order tensor

2.3 Variations, extensions and improvements

The initially proposed algorithms for tensor decomposition exhibited some issues in terms of scalability, interpretability and data modelling. Some of such issues have been addressed in the literature over the last two decades.

Regarding the data modelling, in Chi and Kolda (2012), the authors proposed CP Alternating Poisson Regression (CP-APR) which is designed for sparse count data and therefore is appropriate to deal with weighted time-evolving networks. Moreover, as the name suggests, in Boolean CP (Erdos and Miettinen 2013), the data is assumed to be boolean. Boolean CP is suitable when dealing with unweighted time-evolving networks.

DEDICOM (Kiers 1993) was designed to capture asymmetric relations. In this model, two modes are assumed to be associated with the same objects (which implies that they have the same size). Thus, for a tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times K}$, the output of this algorithm is defined by three structures: two matrices $\mathbf{A} \in \mathbb{R}^{I \times R}$ and $\mathbf{R} \in \mathbb{R}^{R \times R}$, and a tensor $\mathcal{D} \in \mathbb{R}^{R \times R \times K}$, which are obtained by minimizing:

$$\sum_{k=1}^{K} ||\mathcal{X}(:,:,k) - \mathbf{A}\mathcal{D}(:,:,k)\mathbf{R}\mathcal{D}(:,:,k)\mathbf{A}^T||,$$

where $\mathcal{D}(:,:,k)$ corresponds to a diagonal matrix, as illustrated in Fig. 3.

Additionally, some constraints may also be incorporated into the decomposition models, namely, sparsity (Papalexakis et al. 2013) and non-negativity (Bro and De Jong 1997; Shashua and Hazan 2005; Mørup et al. 2008) which allow for a more interpretable analysis of the decomposition result.

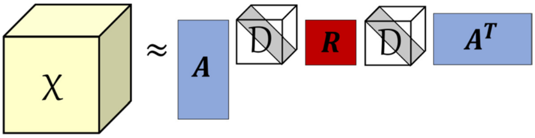


Fig. 3 Illustration of the output of the DEDICOM decomposition of a three order tensor

With respect to scalability, most of the proposed scalable methods for large datasets rely whether on distributed and/or parallel computing (Phan and Cichocki 2011; Kang et al. 2012; Choi and Vishwanathan 2014; Zou et al. 2015; Austin et al. 2016).

In some scenarios, additional data may be available. For example, for multi-relational data, one may have a tensor of type *entities* × *entities* × *relations* describing the different types of relations between the entities and also have a matrix of attributes to describe each entity. In this scenario, the tensor and the matrix share one dimension: the entities. In coupled tensor/matrix decomposition (Acar et al. 2011; Beutel et al. 2014; Jeon et al. 2016), the idea consists of decomposing the multiple data structures jointly by taking into account that they share some of their dimensions.

The opposite scenario may also occur, that is, there might be data missing in the tensor. This problem has been addressed in Liu et al. (2013) and Tan et al. (2013a). More details on tensor decomposition methods and multi-linear algebra can be found in Kolda and Bader (2009) and Sidiropoulos et al. (2017).

2.4 Computational notes

The complexity of the decomposition methods is highly dependent on the optimization algorithm considered to solve the error minimization problem. Nonetheless, nowadays, we already find methods able to manage large datasets efficiently (Kang et al. 2012). For detailed complexity analysis of some of the main tensor decomposition algorithms, the reader may refer to Bader and Kolda (2008).

Moreover, it is worth mentioning that we already find toolboxes with a variety of tensor decomposition algorithms efficiently implemented, namely MATLAB tensor toolbox (Brett et al. 2020), tensorlab (Vervliet et al. 2016) and the n-way toolbox (Andersson and Bro 2000), which are the most well-known. Other toolboxes include the R `rTensor` package (Li et al. 2018) and the Python `TensorLy` library (Kossaifi et al. 2019).

3 Tensor-based SNA

3.1 Evolving social networks as tensors

Typically, networks are characterized by a set of nodes V and a set of edges E, $G = (V, E)$ so that $e = v_1 v_2 \in E \implies v_1, v_2 \in V$. However, not only a network may map multiple relation types $r \in \mathcal{R}$ as the nodes may be of different types and the relations may be directed/asymmetric and involve an arbitrary number of nodes.

Additionally, these networks may evolve over time. The time dimension of these networks has been captured via probabilistic models (Gahrooei and Paynabar 2018) [generally associated with scalability issues (Ranshous et al. 2015)], sketches (Ahn et al. 2012) and matrix-based representations (Tsalouchidou et al. 2018). Nonetheless, in such modelling, time is (only) implicitly modelled and, consequently temporal relations between different network states may be lost. Thus, not only tensors are the natural way to model these networks, but they, along with tensor decomposition approaches, overcome these limitations.

Based on this, given their complexity and high-dimensionality, the natural way to model time-evolving networks is by resorting to tensors. In this context, the most straightforward way of modelling a time-evolving social network as a three-way tensor is illustrated in Fig. 4. However, depending on the data available, different modelling may be considered. For example, assuming the network describes r types of relations, then it can be described by a tensor of order $r + 1$, where the last dimension refers to time. Another possible scenario occurs when the relation involves more than two objects. An example of such modelling would be a tensor of format $\textit{sender} \times \textit{receiver} \times \textit{word} \times \textit{time}$ to describe the number of emails sent from a given employee to another at a given time that contains a given word (Görlitz et al. 2008).

Given the tensor modelling a time-evolving social network, the decomposition result may be directly (without further processing) used to carry out the analysis of the data as it provides a multi-way view of the data. These views of the data are illustrated in Figs. 5, 6 and 7: the same representation may be analysed from different perspectives. In mode-wise view (Fig. 5), each factor matrix is interpreted as a new representation of the corresponding dimension, this new representation can be further subjected to cluster techniques so that nodes exhibiting a similar connection behaviour are identified. Similarly, we can also identify instants of time in which the network behaves identically. In a transversal view (Figs. 6 and 7), each r-tuple of the form $\mathbf{a}_r, \mathbf{b}_r, \mathbf{c}_r$ represents a pattern of the data so that, by analysing the largest entries of such vectors, we are able to find which nodes interact with which and at what times. Thus, in CP, one can interpret each outer product $\mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$ as a concept in the data. In this context, the number of concepts/patterns is given by the number of components R.

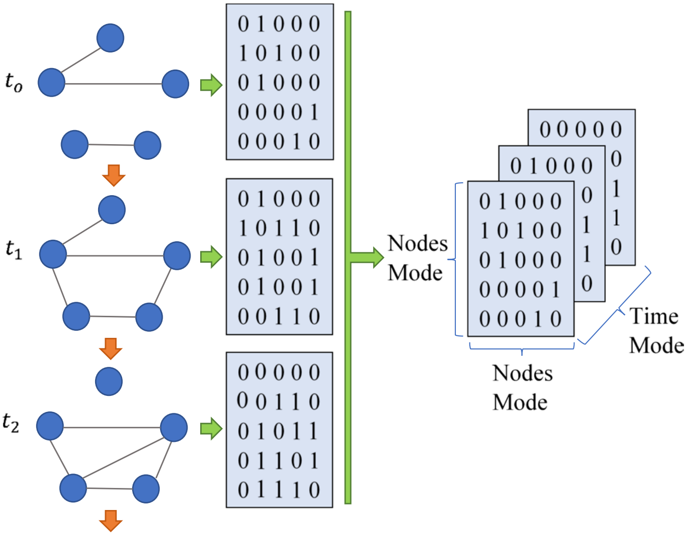


Fig. 4 Example of the modelling of a time-evolving network into a tensor: the network timestamps (left) may be described by their adjacency matrix (middle) forming a sequence over time, corresponding to a tensor (right)

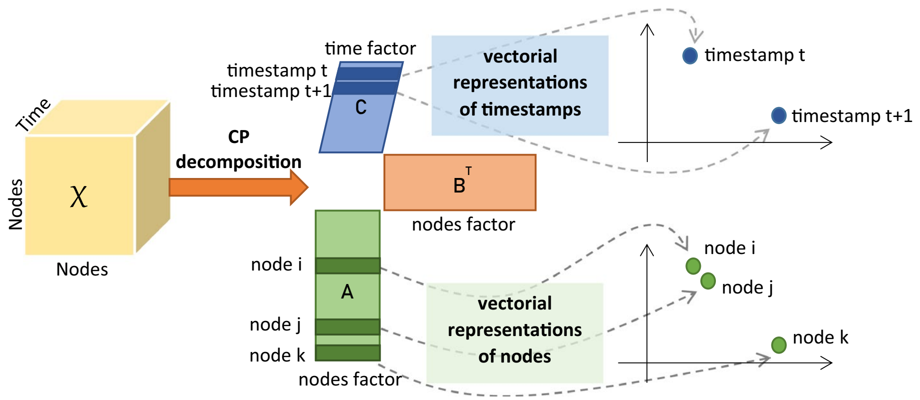
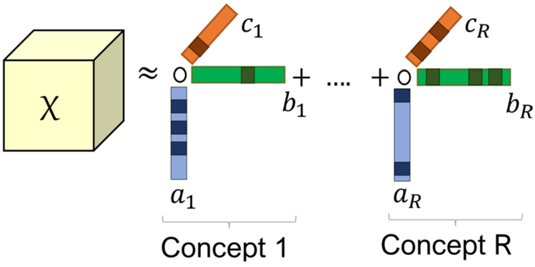


Fig. 5 Mode-wise view: by applying tensor decomposition in the network, we obtain a new (vectorial) representation of each node and timestamp (for illustration purposes we considered a decomposition model of $R = 2$ factors, thus originating 2-D representations)

Fig. 6 Illustration of the exploratory discovery in non-negative CP decomposition concepts



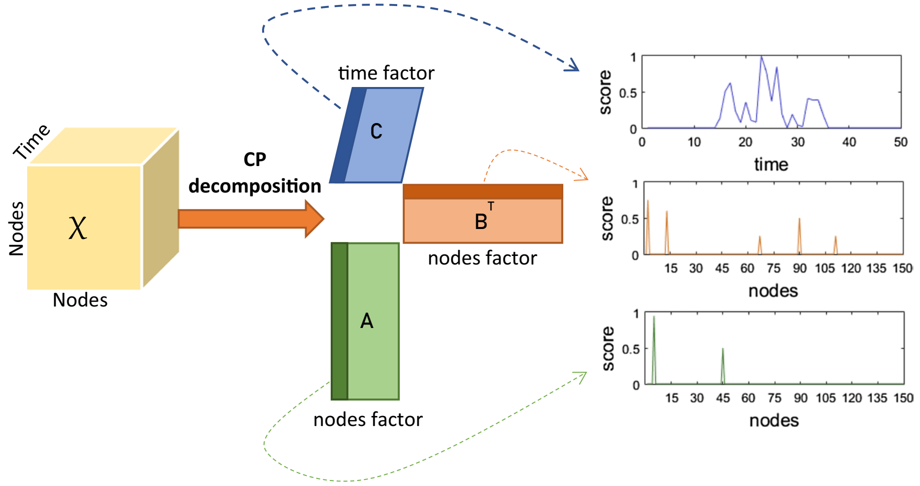


Fig. 7 Transversal view: by applying tensor decomposition in the network, we obtain triples of vectors that describe a pattern in the network

For example, Papalexakis et al. (2012) carried out a study in which they modelled the data as *employee* × *employee* × *month* where entry (i, j, k) is the number of emails exchanged between employees i and j at time k. Given the decomposition result, each concept r, defined by $\{\mathbf{a}_r, \mathbf{b}_r, \mathbf{c}_r\}$, corresponded to a highly interacting group of employees and their communications over time: the entries of each vector are activity scores so that $\mathbf{a}_r$ and $\mathbf{b}_r$ reveals the most active employees in r, while $\mathbf{c}_r$ describes the temporal evolution of the discussion. In other words, each concept is a discussion between some employees (the most active in $\mathbf{a}_r$ and in $\mathbf{b}_r$) and such discussion evolves according to $\mathbf{c}_r$. Additionally, a non-negativity constraint is imposed on the decomposition result to increase interpretability.

A similar idea applies when considering the Tucker decomposition. In Sun et al. (2008), for each factor matrix, the maximum absolute values of column r corresponded to the most relevant dimensions of concept r. Additionally, entry (i, j, k) of the core tensor $\mathcal{Y}$ reflects the "strength" of the corresponding concept, defined by $\{\mathbf{a}_i, \mathbf{b}_j, \mathbf{c}_k\}$.

At an abstraction level, the concepts found by tensor decomposition may be interpreted as patterns describing the data. This approach has been exploited not only for exploratory analysis (Kang et al. 2012; Jeon et al. 2015; Park et al. 2016) but also for specific tasks such as community detection and tracking (Gauvin et al. 2014), anomaly detection (Papalexakis et al. 2014) or temporal co-clustering (Peng and Li 2011).

This type of exploratory analysis is illustrated in Fig. 6 (darker entries are associated with higher values). In Fig. 8, we exhibit one of the concepts detected when decomposing the *employee* × *employee* × *month* Enron dataset. When analysing this concept we verify that there were two most active employees: employee 28, corresponding to David Delainey (in mode 1) and employee 83, corresponding to John Lavorato (in mode 2), two senior executives of the company. The email exchange between these employees was more intense during months 18 and 33, corresponding to October (2000) and January (2002). Delainey was a CEO of the Enron North America and became CEO of the trading unit in the Energy Services at the beginning of 2001, period in which its interaction with Lavorato was higher. This interaction decreased along the declining period of the company.

While the result of tensor decomposition may be directly used for mining these networks, there are more complex and diverse tensor decomposition-based approaches designed to target each of the SNA tasks. Tables 3 and 4 summarize the literature on the topic by listing the existing methods and their specificities. In particular, Table 3 provides

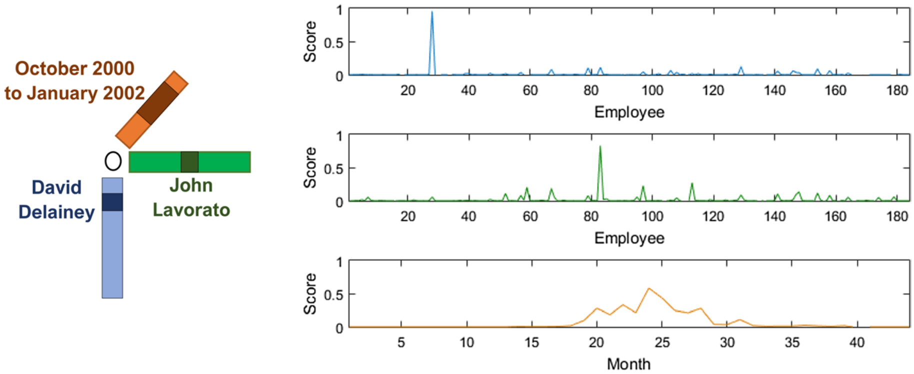


Fig. 8 Illustration of one concept in the *employee* × *employee* × *month* version of Enron dataset. This concept refers to the emails exchange between David Delainey (employee 28) John Lavorato (employee 83)

Table 3 Network type and modelling considered in each of the tensor decompostion-based SNA methods (where $nodes^{(i)}$ denotes the subset of nodes of type i)

Task	Method	Network type	Network modelling
Community detection	Gauvin et al. (2014)	Undirected/directed unweighted	$nodes \times nodes \times time$
	MetaFac (Lin et al. 2009)	Heterogeneous hypergraph	$\{\{\prod_{i\in\{1,\ldots,k\}} nodes^{(i)}\}_{r\in\mathcal{R}}\}_{t=1}^{T}$
	Com2 (Araujo et al. 2014)	Directed unweighted	$nodes \times nodes \times time$
	EgoTen (Sheikholeslami and Giannakis 2017)	(Undirected/directed) unweighted	$nodes \times nodes \times nodes \times time$
	Al-Sharoa et al. (2017)	Directed weighted	$nodes \times nodes \times time$
	LARC-TF (Gorovits et al. 2018)	Weighted undirected	$nodes \times nodes \times time$
Link prediction	Dunlavy et al. (2011)	(Undirected/directed) unweighted	$nodes \times nodes \times time$
	Spiegel et al. (2011)	Uni and bipartite	$nodes^{(i)} \times nodes^{(j)} \times time$
	TriMine-F (Matsubara et al. 2012)	Weighted bipartite	$nodes^{(i)} \times nodes^{(j)} \times time$
	TensorCast (Araujo et al. 2017)	Unweighted	$\{nodes^{(i)} \times nodes^{(j)} \times time\}_{r\in\mathcal{R}}$
Event detection	Maruhashi et al. (2011)	Heterogeneous	$nodes^{(i)} \times nodes^{(j)} \times time$
	Kolda and Sun (2008)	Undirected/directed weighted/unweighted	$(\prod_{i\in\{1,\ldots,k\}} nodes^{(i)}) \times time$
	Koutra et al. (2012)	Undirected/directed weighted/unweighted	$(\prod_{i\in\{1,\ldots,k\}} nodes^{(i)}) \times time$
	Papalexakis et al. (2014)	Unweighted bipartite	$nodes^{(i)} \times nodes^{(j)} \times time$
	Sapienza et al. (2015)	Undirected unweighted	$nodes \times nodes \times time$
	STenSr (Shi et al. 2015)	Undirected/directed weighted/unweighted	$nodes^{(i)} \times nodes^{(j)} \times time$
Concept drift summarization	SeekandDestroy (Pasricha et al. 2018)	Undirected/directed weighted/unweighted	$nodes \times nodes \times time$
	Bader et al. (2007)	Directed weighted	$nodes \times nodes \times time$
	tenClustS (Fernandes et al. 2018)	Undirected unweighted	$nodes \times nodes \times time$

Table 4 Characterization of the tensor decompostion-based SNA methods

Task	Method	Decomposition	Setting	Category	Code avail-ability
Community detection	Gauvin et al. (2014)	CP	Offline	Overlapping community detection	
	MetaFac (Lin et al. 2009)	Coupled-CP	Offline	Non-overlapping community detection	
	Com2 (Araujo et al. 2014)	CP	Offline	Overlapping community detection	✓
	EgoTen (Sheikholeslami and Giannakis 2017)	CP	Offline	Overlapping community detection	✓
	Al-Sharoa et al. (2017)	Tucker	Windowed	Non-overlapping community detection	
	LARC-TF (Gorovits et al. 2018)	CP	Offline	Overlapping community detection	✓
Link prediction	Dunlavy et al. (2011)	CP	Offline	Multiple time step forecast	
	Spiegel et al. (2011)	CP	Offline	Single time step forecast	
	TriMine-F (Matsubara et al. 2012)	CP	Offline	Multiple time step forecast	✓
	TensorCast (Araujo et al. 2017)	Coupled-CP	Offline	Multiple time step forecast	✓
Event detection	Maruhashi et al. (2011)	CP	Offline	Dense block detection	
	Kolda and Sun (2008)	Tucker	Offline	Anomaly detection	
	Koutra et al. (2012)	CP	Offline	Anomaly detection	
	Papalexakis et al. (2014)	CP	Offline	Anomaly detection	
	Sapienza et al. (2015)	CP	Offline	Cleaning procedure	
	STenSr (Shi et al. 2015)	CP	Online	Regular behaviour deviation detection	
Concept drift summarization	SeekandDestroy (Pasricha et al. 2018)	CP	Online	Communication patterns drift	✓
	Bader et al. (2007)	DEDICOM	Offline	Interpretable summarization	
	tenClustS (Fernandes et al. 2018)	CP	Windowed	Structural summarization	✓

details on the network types considered in each method and their modelling as tensors. In Table 4, we characterize the works according to the tensor decomposition type and setting (online, sliding window or offline) as well as the SNA subproblem addressed. These works are explained in more detail in the following sections.

3.2 Community detection

Informally the concept of community is associated with a set of individuals that have something in common: the location, an interest, and so forth. In the context of social networks, a community usually refers to a set of nodes which highly interact within each other, but loosely interact with the remaining network (Fortunato 2010). For example, in a friendship network, usually one can find different communities: friends from school, work colleagues or family. These communities may have elements in common. A case of community overlap would be a cousin which worked in the same company and consequently was also a work colleague.

In community detection, the goal is to find the inherent communities. If the networks evolve over time, then the evolution of such communities may also be tracked in order to detect splitting, merging, birth and death of communities.

MetaFac (Lin et al. 2009) was designed for heterogeneous hypergraphs, a richer scenario in which nodes and edges can be of different types and edges relations may involve more than two nodes. This type of networks are useful to model interactions in social media platforms such as Facebook where one can find different types of objects (users, wall posts,...) and different types of relations (friendship between users, users reply to a wall post, ...). Based on this, the interactions of a relation type r are modelled using a tensor $\mathcal{X}^{(r)}$, whose modes correspond to the node types involved in such relation type. Given the set of relation tensors, the idea is to resort to coupled tensor factorization by accounting for the node types that are shared across different relation types. For example, in Facebook the friendship relation involves only the users and such relation can be modelled by a *users* × *users* tensor. On the other hand, the relation inherent to a user making a post at another user's wall at a given time can be modelled by a *users* × *users* × *time* tensor. In this case the two relations share the users mode. The number of communities K is set as the number of factors in the decomposition and the output of the decomposition is a core tensor shared by all the relation tensors and a set of factor matrices, each associated with one of the node types. The membership of element i of type α to community k is computed based on the kth diagonal entry of the core tensor and the (i, k) entry of factor matrix associated to node type α. The authors extended their work to deal with evolving scenarios, but did not model the time dimension explicitly in the $\mathcal{X}^{(r)}$ relation tensors. Instead, the decomposition in each instant is carried out similarly to the static scenario, with the difference that smoothing constraints between the factor matrices observed in the previous timestamp and the ones to model the incoming timestamp are added. Thus, in practice, a sequence of factor matrices for each timestamp is obtained and, since they map the community structure of the network, they can be used to track the communities evolution.

Com2 (Araujo et al. 2014) is a method to detect and track the communities evolution in directed unweighted time-evolving networks. In this method, tensor decomposition is used to obtain an initial community candidate. In particular, the authors apply CP with a single component so that the entries are interpreted as activity scores (similarly to the described in Sect. 3.1)—most active nodes and instants represent the community candidate. The community candidate is then used to drive the search for the most consistent community in terms of minimum description length (MDL) (Grünwald 2007). The structure found corresponds to a dense block of the network, which is expected to map a community and its evolution in the corresponding instants. In order to find other communities, the edges in the block are removed from the original tensor and this process may be repeated sequentially. In this method, the number of communities does not need to be predefined (and the method is parameter free). However, this method does not operate online.

In Al-Sharoa et al. (2017), the authors proposed a framework for spectral clustering in weighted undirected dynamic networks, by resorting to Tucker tensor decomposition. The network is modelled as a *nodes* × *nodes* × *time* tensor corresponding to the sequence of adjacency matrices over time. Since the network is undirected, an equality constraint is imposed on the decomposition output regarding the factor matrices associated with *nodes* dimensions. In other words, it is imposed that $\mathbf{A} = \mathbf{B}$ in (1). The decomposition result is used to track the community structure. In particular, the authors proposed a metric to quantify the level of change in the community structure. Given the change points, the number of clusters is estimated accordingly (Von Luxburg 2007) and the communities are estimated by applying k-means to the nodes factor matrix. Their methodology can be carried out using a sliding time window or a landmark time window (with forgetting factor).

Recently, a framework for Learning Activity-Regularized Overlapping Communities using Tensor Factorization (LARC-TF) (Gorovits et al. 2018) was proposed. In this work, the network is assumed to be weighted and undirected and is modelled as a 3-way tensor (*nodes* × *nodes* × *time*). The goal is to find two matrices: a "community matrix" describing the level of association of each node to the communities and a "activation matrix" describing the activity of each community over time. These matrices are found by resorting to regularized constrained tensor decomposition. The constraints include non-negativity (for interpretability purposes) and shared factor matrices across the two nodes modes ($\mathbf{A} = \mathbf{B}$ in (2)). The authors also considered a smoothness regularization term in order to appropriately model the activity and inactivity intervals in the activation matrix. Additionally, the number of communities is set as the number of components and is estimated based on an adaptation of the CORCONDIA (Bro and Kiers 2003; Papalexakis and Faloutsos 2015). Similarly, the authors proposed an ADMM-based (Boyd et al. 2011) approach to solve the decomposition and obtain the factor matrices. The nodes factor matrix and the temporal factor matrix are interpreted as the community and the activation matrices, respectively.

Finally, in order be able to find overlapping communities, Sheikholeslami and Giannakis (2017) proposed a different view of a network. In particular, the authors described a network by its set of ego-networks (that is, using the local subgraphs formed by a node and its neighbours). In this context, since each node is represented by the adjacency matrix of its ego-network, the network state at time t is modelled as a *nodes* × *nodes* × *nodes* tensor, formed by the set of such adjacency matrices. Given such tensor, the method consists of applying a constrained CP model to the data. The constraints include non-negativity on all the factor matrices and unit norm on the rows of the factor matrix associated with the third mode. By considering these constraints, the ith row of the third factor matrix can be interpreted as a community membership vector for node i. The authors also extended their work to a time-evolving scenario, in which case the tensor would be of order 4 (*nodes* × *nodes* × *nodes* × *time*) and the community membership would also be dependent on the temporal activity, encoded in the forth factor matrix. In more detail, assuming that the decomposition outputs matrices $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$, with $\mathbf{C}$ and $\mathbf{D}$ providing respectively the community assignment and the temporal evolution, then the membership score of node i to community k at time t is computed as the community assignment score, given by $\mathbf{C}(i, k)$, weighted by the community activity at that instant, given by $\mathbf{D}(t, k)$.

3.3 Link prediction

Let us assume we have access to the human proximity network over time, then in case of an epidemic disease spread, it would be useful to understand the proximity patterns and predict future contacts in order to avoid the propagation of the disease to uninfected individuals. The task of inferring which links will occur in the future defines the scope of temporal link prediction (Lü and Zhou 2011). In other words, given the states of the network in the previous instants, the goal of link prediction is to predict which entities will interact in the future.

Two of the first approaches to tackle this problem using tensor decomposition were proposed in Dunlavy et al. (2011) and Spiegel et al. (2011). The idea exploited in these works is similar: the network states between instants 1 and T are modelled as a *nodes* × *nodes* × *time* tensor and approximated via tensor decomposition. Given the decomposition factor matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$, the network state at time t is approximated by $\sum_r^R (\mathbf{a}_r \circ \mathbf{b}_r) \times \mathbf{c}_r(t)$. Therefore, the state of the network at time $T+1$ may be estimated as $\sum_r^R (\mathbf{a}_r \circ \mathbf{b}_r) \times \hat{\mathbf{c}}(T+1)$, with $\hat{\mathbf{c}}(T+1)$ being an estimate of the $T+1$ row of the temporal factor matrix $\mathbf{C}$. Based on this, the authors propose strategies to estimate the future temporal trend $\hat{\mathbf{c}}(T+1)$ based on the remaining temporal evolution (encoded in $\mathbf{C}$). In the resulting estimation, the highest score values are associated with the links which are more likely to occur. A major limitation of these methods is their high sensitivity to the initialization of the tensor decomposition result (Fernandes et al. 2017).

A similar idea was exploited in the works of Matsubara et al. (2012) and Araujo et al. (2017).

The novelty of TriMine-F (Matsubara et al. 2012) comes from the multiple time granularity levels (such as hour, day, week, month, ...) considered with the aim of improving the forecasting. In particular, data is modelled as *object* × *user* × *time* tensors, with varying granularity on the time mode. The tensor associated to the highest granularity (for example, hours) is decomposed into factor matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$. The factors matrices $\mathbf{A}$ and $\mathbf{B}$, associated respectively with objects and users, are shared when computing the decomposition of the remaining tensors (which are associated with a longer time windows—for example, days and weeks). Given the temporal factor matrices associated with each time granularity, forecasting is applied to them in order to estimate the future trend $\hat{\mathbf{c}}(T+1)$. It is noteworthy that, despite not specifically designed for time-evolving networks, TriMine-F can be applied to such type of data.

In TensorCast (Araujo et al. 2017), the idea is to incorporate additional information and resort to coupled decomposition to account for such side information. The authors also proposed a greedy strategy to obtain the top scores (associated with the links which are more likely to occur) without computing explicitly the future state of the network.

It is noteworthy that all of these methods are based on CP decomposition and exploit a similar idea, illustrated in Fig. 9. The main differences between the methods reside on the forecasting approach and on the optional usage of side information or multiple time granularities. Moreover, Spiegel et al. (2011) considered the problem of predicting the state of the network at the incoming timestamp, while in the other works, the goal was to predict the state of the network in the next L time instants. The main limitation of these methods is that they were designed to work in an offline setting and consequently, when new data arrives, the forecasting models need to be recomputed.

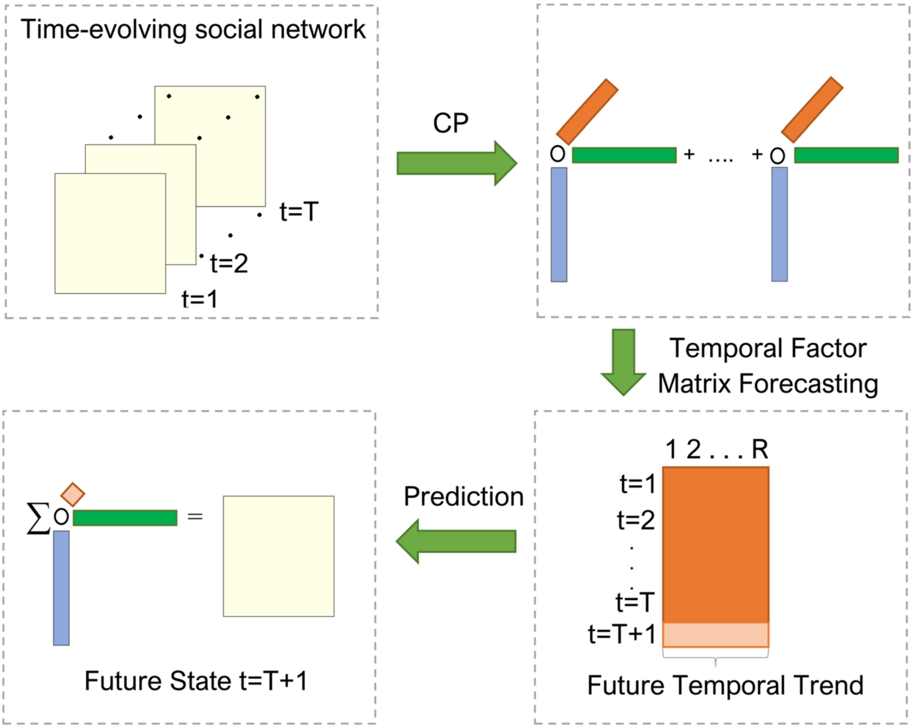


Fig. 9 Schematic illustration of the link prediction process based on tensor decomposition

4 Other applications

4.1 Anomaly detection

The goal of anomaly detection in time-evolving networks is to detect anomalous/deviation behaviors (Akoglu et al. 2015; Fanaee-T and Gama 2016b). According to Ranshous et al. (2015), anomalies are dubbed as events or changes based on its durability: events are assumed to be temporary while changes are associated with longer periods of time. Examples of events include: (1) abrupt changes in an IP network traffic, associated with malicious activities, such as denial-of-service attacks; (2) peak of wall-posts at a given user's Facebook profile due to his birthday. In a co-authorship network, if a given author starts working on a different field then he/she will probably collaborate with new people (with which the author has never worked before); such a drift is considered a change (not an event).

The most common and straightforward approach to detect anomalies using tensor decomposition is based on the following assumption: since tensor decomposition captures the general structure of the data, deviations are not captured. Consequently, when comparing the original tensor with the reconstructed one, one is able to detect the anomalies by spotting the tensor entries that have the highest reconstruction error. This approach is exploited in Kolda and Sun (2008), Koutra et al. (2012) and Sun et al. (2008). This strategy allows for an analysis at a global level by tracking the reconstruction error of each temporal network state and at a local level by tracking the reconstruction error of a given node. An example is provided in Fig. 10: the plot of the decomposition reconstruction error of the ChallengeNetwork dataset (Rayana and Akoglu 2014) shows 2 peaks (at 376 and 1226, marked with a vertical dashed line), corresponding to instants of anomalous high activity. Despite allowing the discovery of multiple types of anomalies (from abrupt interaction increase to concept drift), this type of approaches fail at being automatic, which may be discouraging when the number of concepts to analyse is high.

Sapienza et al. (2015) proposed a different perspective of anomaly in the context of networks. According to the authors, anomalies are noisy patterns in the data that should be removed. In this approach, tensor decomposition is used to find the communication patterns, which are further fed into a classifier in order to distinguish the anomalous activity periods from the normal. Given a detected anomaly, the nodes and time periods associated are discarded from the original tensor. This procedure is iteratively repeated on the "cleaned" tensor until no anomalies are detected. The major limitation pointed out by the authors is the inefficiency of the decomposition method considered. From a theoretical point of view, this approach may be formulated as recovering a "clean" version of the dataset (with the noise removed). This may be useful when dealing with sensor data (different environments may bias the sensors measurements). A similar problem was also addressed in Tan et al. (2013b), Goldfarb and Qin (2014) and Xie et al. (2017); the goal was to separate the noise from the original data by resorting to tensor decomposition models.

A limitation of most of the approaches previously described is that they do not work in real-time. STenSr (Shi et al. 2015) aims at tackling such issue. In particular, in STenSr the authors keep track of the tensor decomposition as new data arrives. Thus, when a new tensor is available, the old tensor decomposition is incrementally updated and the anomaly detection is carried out by accounting for the deviation of the new rows in the temporal factor matrix. This deviation is computed based on the distance of the new rows to the centroid of the remaining. Despite not being specifically developed for network data, the application of this method to such type of data is possible, as the authors demonstrate.

Pasricha et al. (2018) were pioneers in the field of tensor decomposition-based concept drift detection by proposing SeekandDestroy. The idea exploited in this work was to compare the decomposition result of the tensor observed so far with the new incoming tensor. In particular, the concept drift detection is carried out by analysing which concepts are new, missing or common across the old and the new data. Using such information, the authors are able to incrementally keep the decomposition result of the whole data and carry out the analysis. Since the concepts found depend on the number of factors considered, the success of this approach is highly dependent on the quality of the rank estimation method, as pointed out by the authors.

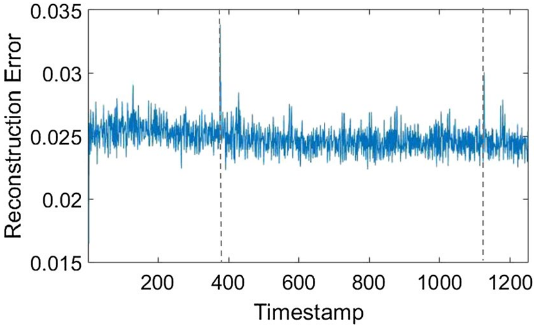


Fig. 10 Reconstruction error per timestamp in the (unweighted version) of the ChallengeNetwork dataset: dashed vertical lines correspond to anomalous timestamps

Finally, with respect to heterogeneous networks, MultiAspectForensics (Maruhashi et al. 2011) is a strategy for finding dense subgraphs in the network, having applications in network intrusion detection, for example. As a preliminary step, the heterogeneous network must be rearranged into a tensor by considering (1) each node type, (2) the edge attributes and (3) time as modes in the tensor. Since the factor matrices assign eigen-scores to the attributes, the idea consists of mining such information to find spikes of eigen-scores. These are expected to model a relevant and unexpected pattern. In this context, dense bipartite subgraphs in which all the edges are of the same type are regarded as anomalous and should be spotted by the method.

4.2 Summarization

Briefly, the goal of graph summarization is to find a concise representation of the original (often) large-scale network that preserves some of its properties (Liu et al. 2018). Depending on the goal of the summarization, different strategies are used.

Regarding tensor-based summarization techniques, it is noteworthy that the decomposition result itself may be interpreted as a summary of the data since it is a new and more compact representation of the original tensor. For example, for a time-evolving network modelled as $nodes \times nodes \times time$ tensor, each factor matrix is new representation either of the nodes or of the time evolution.

In Shah et al. (2015), the authors modelled the network as a tensor, but did not resort to tensor decomposition. The idea exploited was to use the MDL principle to uncover the linking patterns. The summary was composed by the set of discovered patterns.

In Bader et al. (2007), the authors considered DEDICOM tensor decomposition to capture the asymmetry in the relations between the entities. With the information provided by the decomposition results, the authors inferred groups with similar connection patterns and inferred also the interaction level between such groups. The nodes grouping is inferred from the columns of the node factor matrix $\mathbf{A}$, while the groups interactions are inferred from matrix $\mathbf{R}$ (recall Sect. 2.3). This results in a smaller graph mapping the main relations captured by the decomposition. A similar analysis was carried out in Peng and Li (2011).

More recently, Fernandes et al. (2018) proposed a tensor-based extension of GraSS (LeFevre and Terzi 2010) to dynamic networks. Contrary to Bader et al. (2007), this method was developed for unweighted undirected networks. Nonetheless, its goal is also to preserve the structural/topological properties of the network. In this work tensor decomposition is used to generate the nodes representation, which is then used to capture the connectivity patterns. An illustration of this type of summarization is provided in Fig. 11.

4.3 Visualization

The visualization of the networks may provide insights on the underlying communication patterns. However, given the large-scale of most real world networks, its visualization is not always feasible or useful. In this context, the usage of summaries may be a possible solution. Both Bader et al. (2007) and Fernandes et al. (2018)), described in the previous section, can be used for visualization purposes.

In Sun et al. (2009), the authors proposed MultiVis, a tensor decomposition-based tool designed for the purposes of single and cross mode clustering, which enables the visualization and navigation of such clustering results. This framework is characterized by two phases. In the first phase, the tensor is decomposed using Tucker (with biased

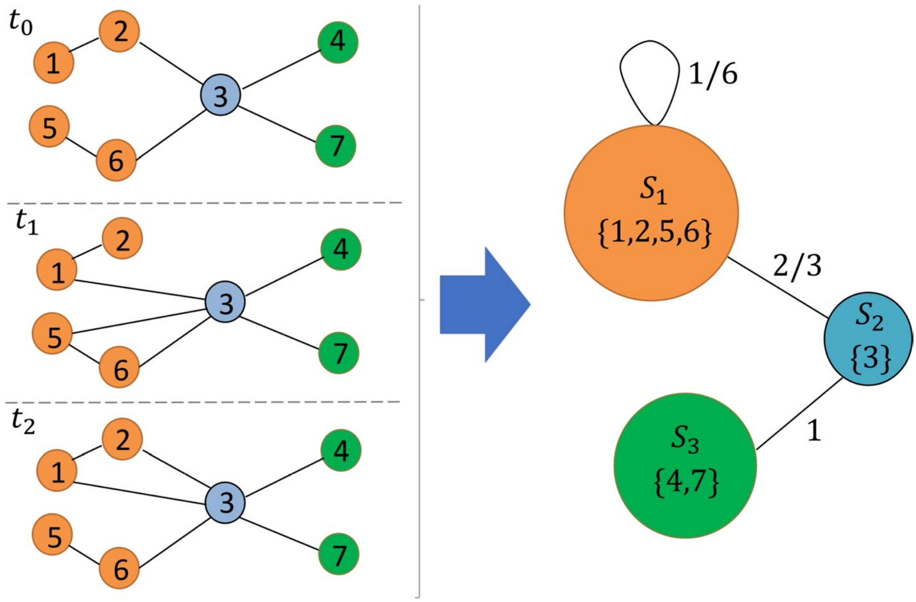


Fig. 11 Example of a summary of a unweighted undirected time-evolving network: 3 original network timestamps (left) and corresponding summary (right). The summary edges are weighted based on the number of links shared across the summary nodes elements

sampling strategies (Drineas and Mahoney 2007) in order to deal with large data). Then, the core tensor and the factor matrix of each mode of the tensor are combined to generate the representation of the corresponding modes. The elements in each mode are clustered based on such representation. Finally, the clustering results of each mode are combined for a cross-mode interpretation. As an illustrative example, let us consider an email network, then the tensor *sender* × *receiver* × *time* is decomposed and the clustering of the senders, receivers and time is first carried out separately and later combined to track the time evolution of the interactions between the clusters of senders and receivers. The second phase of the framework consists of the visualization of these clustering results. To visualize the network, the authors propose the application of hierarchical clustering in each mode, thus allowing for a multi-level view of the interactions in the data.

Additionally, Oliveira and Gama (2011) introduced the concept of social trajectories. In this work the networks are described as a *nodes* × *features* × *time* tensor $\mathcal{X}$ in which the set of node features included the degree, the closeness, the betweenness and (local) clustering coefficient. The idea exploited is to use the decomposition result to project the data into a new space. In more detail, the tensor modelling the network is decomposed using Tucker decomposition. Then the first two factors of the *features* factor matrix (matrix $\mathbf{B}$ in (1)) is interpreted as the new feature space and it is used to project the data. The result is a tensor $\mathcal{Y} = \mathcal{X} \times_2 \mathbf{B}(:, 1 : 2)$. In practice, tensor $\mathcal{Y}$ gives a two dimensional representation of each node for each timestamp. The set of two dimensional points for each node over time defines the social trajectory of the node. This representation facilitates the understanding of the evolution of the nodes over time. On one hand, the network features that most contribute to the components of the new feature space allow for an interpretation of the new feature space. Moreover, by visualizing the trajectories, one can identify groups of nodes with similar behaviour.

It is noteworthy that despite their differences, both methods previously described (Sun et al. 2009; Oliveira and Gama 2011) are derived from a similar idea: resort to Tucker decomposition to obtain new representations of the network data and use such representation to visualize the network behaviour.

5 Conclusion

In this work we cover the existing literature on tensor decomposition-based analysis of time-evolving networks. A growing research area that has attracted the researchers interest in recent years.

By comparing their models with other methods such as matrix decomposition and time-series based approaches, the tensor decomposition-based works suggest that the success of tensor decomposition in SNA arises from explicitly (and jointly) modelling the multiple dimensions of social networks.

In this context, tensor decomposition is usually considered to generate new, lower dimensional and more interpretable representations of the dimensions of the networks, namely the nodes and the time evolution. These new representations are then processed according to the analysis goal: in community detection, summarization and visualization tasks, the nodes factor matrix assumes a major role in the analysis; likewise, in the case of link prediction, the core of the methods resides on the processing/forecasting of the temporal factor matrix.

Despite the advances in this field, there are still some challenges which remain few explored. In particular, there are issues associated with tensor decomposition itself which are known to have impact in the context of SNA tasks, these include the choice of the number of components and the lack of unique solutions (Kolda and Bader 2009), according to which multiple runs of the algorithm may provide different solutions/factor matrices (Fernandes et al. 2017). Finally, it is also worth mentioning that traditional tensor decomposition usually fails at capturing local temporal dependencies, providing only a global perspective of the data and neglecting local dynamics (Fernandes et al. 2019). Another relevant issue that remains almost undressed is the incrementality (as it can be observed in Tables 3 and 4). In the context of social networks, real-time analysis assumes an important role: the discoveries must be made in real time. To meet this end the methods should be scalable and efficient to incorporate incoming data. This direction remains unexplored when referring to tasks such as link prediction, for example. Finally, we also observed that the potential of tensor decomposition for the analysis of hypergraphs and heterogeneous networks is few explored.

Based on this and given the promising results demonstrated so far, we believe this research area will continue to grow, with significant improvements if the challenges previously stated are tackled.

Acknowledgements This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project UIDB/50014/2020. Sofia Fernandes also acknowledges the support of FCT via the PhD scholarship PD/BD/114189/2016.

References

Acar E, Kolda TG, Dunlavy DM (2011) All-at-once optimization for coupled matrix and tensor factorizations. arXiv preprint arXiv:11053422

Ahn KJ, Guha S, McGregor A (2012) Graph sketches: sparsification, spanners, and subgraphs. In: Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI symposium on principles of database systems, pp 5–14

Akoglu L, Tong H, Koutra D (2015) Graph based anomaly detection and description: a survey. Data Min Knowl Discov 29(3):626–688

Al-Sharoa E, Al-khassaweneh M, Aviyente S (2017) A tensor based framework for community detection in dynamic networks. In: 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 2312–2316

Andersson CA, Bro R (2000) The n-way toolbox for matlab. Chemom Intell Lab Syst 52(1):1–4

Araujo MR, Ribeiro PMP, Faloutsos C (2017) Tensorcast: forecasting with context using coupled tensors (best paper award). In: 2017 IEEE international conference on data mining (ICDM). IEEE, pp 71–80

Araujo M, Papadimitriou S, Günnemann S, Faloutsos C, Basu P, Swami A, Papalexakis EE, Koutra D (2014) Com2: fast automatic discovery of temporal (‘comet’) communities. In: Pacific-Asia conference on knowledge discovery and data mining. Springer, pp 271–283

Austin W, Ballard G, Kolda TG (2016) Parallel tensor compression for large-scale scientific data. In: 2016 IEEE international parallel and distributed processing symposium. IEEE, pp 912–922

Bader BW, Harshman RA, Kolda TG (2007) Temporal analysis of semantic graphs using asalsan. In: ICDM. IEEE, pp 33–42

Bader BW, Kolda TG (2008) Efficient matlab computations with sparse and factored tensors. SIAM J Sci Comput 30(1):205–231

Bak P, Paczuski M, Shubik M (1996) Price variations in a stock market with many agents. arXiv preprint arXiv:cond-mat/9609144

Barabási AL, Albert R (1999) Emergence of scaling in random networks. Science 286(5439):509–512

Bauer F, Lizier JT (2012) Identifying influential spreaders and efficiently estimating infection numbers in epidemic models: a walk counting approach. EPL (Europhys Lett) 99(6):68007

Beutel A, Talukdar PP, Kumar A, Faloutsos C, Papalexakis EE, Xing EP (2014) Flexifact: scalable flexible factorization of coupled tensors on hadoop. In: Proceedings of the 2014 SIAM international conference on data mining. SIAM, pp 109–117

Billio M, Getmansky M, Lo AW, Pelizzon L (2012) Econometric measures of connectedness and systemic risk in the finance and insurance sectors. J Financ Econ 104(3):535–559

Boldi P, Vigna S (2004) The webgraph framework I: compression techniques. In: Proceedings of the 13th international conference on World Wide Web. ACM, pp 595–602

Boyd S, Parikh N, Chu E, Peleato B, Eckstein J et al (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach Learn 3(1):1–122

Brett W, Bader TGK et al (2020) Matlab tensor toolbox, version. https://www.tensortoolbox.org

Bro R, De Jong S (1997) A fast non-negativity-constrained least squares algorithm. J Chemom J Chemom Soc 11(5):393–401

Bro R, Kiers HA (2003) A new efficient method for determining the number of components in parafac models. J Chemom J Chemom Soc 17(5):274–286

Brunetti C, Harris JH, Mankad S, Michailidis G (2019) Interconnectedness in the interbank market. J Financ Econ 133(2):520–538

Carroll JD, Chang JJ (1970) Analysis of individual differences in multidimensional scaling via an n-way generalization of “eckart-young” decomposition. Psychometrika 35(3):283–319

Ceulemans E, Kiers HA (2006) Selecting among three-mode principal component models of different types and complexities: a numerical convex hull based method. Br J Math Stat Psychol 59(1):133–150

Chen H, Chung W, Qin J, Reid E, Sageman M, Weimann G (2008) Uncovering the dark web: a case study of jihad on the web. J Am Soc Inf Sci Technol 59(8):1347–1359

Chi EC, Kolda TG (2012) On tensors, sparsity, and nonnegative factorizations. SIAM J Matrix Anal Appl 33(4):1272–1299

Choi JH, Vishwanathan S (2014) Dfacto: distributed factorization of tensors. In: Advances in neural information processing systems, pp 1296–1304

da Silva Fernandes S, Tork HF, da Gama JMP (2017) The initialization and parameter setting problem in tensor decomposition-based link prediction. In: 2017 IEEE international conference on data science and advanced analytics (DSAA), pp 99–108

Drineas P, Mahoney MW (2007) A randomized algorithm for a tensor-based generalization of the singular value decomposition. Linear Algebra Appl 420(2–3):553–571

Dunlavy DM, Kolda TG, Acar E (2011) Temporal link prediction using matrix and tensor factorizations. ACM Trans Knowl Discov Data (TKDD) 5(2):10

Erdos D, Miettinen P (2013) Discovering facts with Boolean tensor tucker decomposition. In: Proceedings of the 22nd ACM international conference on conference on information and knowledge management. ACM, pp 1569–1572

Faloutsos M, Faloutsos P, Faloutsos C (1999) On power-law relationships of the internet topology. SIGCOMM Comput Commun Rev 29(4):251–262

Fanaee-T H, Gama J (2016a) Event detection from traffic tensors: a hybrid model. Neurocomputing 203:22–33

Fanaee-T H, Gama J (2016b) Tensor-based anomaly detection: an interdisciplinary survey. Knowl Based Syst 98:130–147

Fernandes S, Fanaee-T H, Gama J (2018) Dynamic graph summarization: a tensor decomposition approach. Data Min Knowl Discov 32(5):1397–1420

Fernandes S, Fanaee-T H, Gama J (2019) Evolving social networks analysis via tensor decompositions: from global event detection towards local pattern discovery and specification. In: International conference on discovery science. Springer, pp 385–395

Ferrara E, De Meo P, Catanese S, Fiumara G (2014) Detecting criminal organizations in mobile phone networks. Expert Syst Appl 41(13):5733–5750

Fortunato S (2010) Community detection in graphs. Phys Rep 486(3–5):75–174

Gahrooei MR, Paynabar K (2018) Change detection in a dynamic stream of attributed networks. J Qual Technol 50(4):418–430

Gauvin L, Panisson A, Cattuto C (2014) Detecting the community structure and activity patterns of temporal networks: a non-negative tensor factorization approach. PloS ONE 9(1):e86028

Génois M, Vestergaard CL, Fournet J, Panisson A, Bonmarin I, Barrat A (2015) Data on face-to-face contacts in an office building suggest a low-cost vaccination strategy based on community linkers. Netw Sci 3:326–347

Goldfarb D, Qin Z (2014) Robust low-rank tensor recovery: models and algorithms. SIAM J Matrix Anal Appl 35(1):225–253

Goñi J, Esteban FJ, de Mendizábal NV, Sepulcre J, Ardanza-Trevijano S, Agirrezabal I, Villoslada P (2008) A computational analysis of protein-protein interaction networks in neurodegenerative diseases. BMC Syst Biol 2(1):52

Görlitz O, Sizov S, Staab S (2008) Pints: peer-to-peer infrastructure for tagging systems. In: IPTPS, p 19

Gorovits A, Gujral E, Papalexakis EE, Bogdanov P (2018) Larc: learning activity-regularized overlapping communities across time. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 1465–1474

Grünwald PD (2007) The minimum description length principle. MIT Press, Cambridge

Harshman RA (1970) Foundations of the PARAFAC procedure: models and conditions for an “explanatory” multi-modal factor analysis. In: UCLA working papers in phonetics, vol 16, no 1, p 84

Heiberger RH (2018) Predicting economic growth with stock networks. Phys Stat Mech Appl 489:102–111

Hitchcock FL (1927) The expression of a tensor or a polyadic as a sum of products. J Math Phys 6(1–4):164–189

Isella L, Stehlé J, Barrat A, Cattuto C, Pinton JF, Van den Broeck W (2011) What’s in a crowd? Analysis of face-to-face behavioral networks. J Theor Biol 271(1):166–180

Jeon I, Papalexakis EE, Kang U, Faloutsos C (2015) Haten2: billion-scale tensor decompositions. In: 2015 IEEE 31st international conference on data engineering (ICDE). IEEE, pp 1047–1058

Jeon B, Jeon I, Sael L, Kang U (2016) Scout: scalable coupled matrix-tensor factorization-algorithm and discoveries. In: IEEE 32nd international conference on data engineering (ICDE), 2016. IEEE, pp 811–822

Jordán F, Nguyen TP, Wc L (2012) Studying protein–protein interaction networks: a systems view on diseases. Brief Funct Genom 11(6):497–504

Kang U, Papalexakis E, Harpale A, Faloutsos C (2012) Gigatensor: scaling tensor analysis up by 100 times-algorithms and discoveries. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 316–324

Keeling MJ, Eames KT (2005) Networks and epidemic models. J R Soc Interface 2(4):295–307

Kiers HA (1993) An alternating least squares algorithm for parafac2 and three-way dedicom. Comput Stat Data Anal 16(1):103–118

Kiers HAL (2000) Towards a standardized notation and terminology in multiway analysis. J Chemom 14(3):105–122

Kiers HA, Kinderen A (2003) A fast method for choosing the numbers of components in tucker3 analysis. Br J Math Stat Psychol 56(1):119–125

Kolda TG, Sun J (2008) Scalable tensor decompositions for multi-aspect data mining. In: Eighth IEEE international conference on data mining, 2008. ICDM'08. IEEE, pp 363–372

Kolda TG, Bader BW (2009) Tensor decompositions and applications. SIAM Rev 51(3):455–500

Kossaifi J, Panagakis Y, Anandkumar A, Pantic M (2019) Tensorly: tensor learning in python. J Mach Learn Res 20(1):925–930

Koutra D, Papalexakis EE, Faloutsos C (2012) Tensorsplat: spotting latent anomalies in time. In: 2012 16th Panhellenic conference on informatics (PCI). IEEE, pp 144–149

Kwak H, Lee C, Park H, Moon S (2010) What is twitter, a social network or a news media? In: Proceedings of the 19th international conference on World wide web. ACM, pp 591–600

LeFevre K, Terzi E (2010) Grass: graph structure summarization. In: Proceedings of the 2010 SIAM international conference on data mining. SIAM, pp 454–465

Leskovec J, Adamic LA, Huberman BA (2007) The dynamics of viral marketing. ACM Trans Web (TWEB) 1(1):5

Leskovec J, Kleinberg J, Faloutsos C (2005) Graphs over time: densification laws, shrinking diameters and possible explanations. In: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. ACM, pp 177–187

Leskovec J, Mcauley JJ (2012) Learning to discover social circles in ego networks. In: Advances in neural information processing systems, pp 539–547

Ley M (2002) The dblp computer science bibliography: evolution, research issues, perspectives. In: International symposium on string processing and information retrieval. Springer, pp 1–10

Li J, Bien J, Wells M, Li MJ (2018) Package 'rtensor'. J Stat Softw 87:1–31

Lin YR, Sun J, Castro P, Konuru R, Sundaram H, Kelliher A (2009) Metafac: community discovery via relational hypergraph factorization. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 527–536

Liu J, Musialski P, Wonka P, Ye J (2013) Tensor completion for estimating missing values in visual data. IEEE Trans Pattern Anal Mach Intell 35(1):208–220

Liu K, Da Costa JPCL, So HC, Huang L, Ye J (2016) Detection of number of components in candecomp/parafac models via minimum description length. Digit Signal Process Rev J 51:110–123

Liu Y, Safavi T, Dighe A, Koutra D (2018) Graph summarization methods and applications: a survey. ACM Comput Surv (CSUR) 51(3):62

Lü L, Zhou T (2011) Link prediction in complex networks: a survey. Phys Stat Mech Appl 390(6):1150–1170

Ma X, Dong D (2017) Evolutionary nonnegative matrix factorization algorithms for community detection in dynamic networks. IEEE Trans Knowl Data Eng 29(5):1045–1058

Mankad S, Michailidis G (2013) Structural and functional discovery in dynamic networks with non-negative matrix factorization. Phys Rev E 88(4):042812

Maruhashi K, Guo F, Faloutsos C (2011) Multiaspectforensics: pattern mining on large-scale heterogeneous networks with tensor analysis. In: 2011 International conference on advances in social networks analysis and mining (ASONAM). IEEE, pp 203–210

Mastrandrea R, Fournet J, Barrat A (2015) Contact patterns in a high school: a comparison between data collected using wearable sensors, contact diaries and friendship surveys. PloS ONE 10(9):e0136497

Matsubara Y, Sakurai Y, Faloutsos C, Iwata T, Yoshikawa M (2012) Fast mining and forecasting of complex time-stamped events. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 271–279

Michalski R, Palus S, Kazienko P (2011) Matching organizational structure and social network extracted from email communication. In: Lecture notes in business information processing, vol 87. Springer, Berlin, pp 197–206

Milgram S (1967) The small world problem. Psychol Today 2:60–67

Mørup M, Hansen LK (2009) Automatic relevance determination for multi-way models. J Chemom 23(7–8):352–363

Mørup M, Hansen LK, Arnfred SM (2008) Algorithms for sparse nonnegative tucker decompositions. Neural Comput 20(8):2112–2131

Oliveira M, Gama J (2011) Visualizing the evolution of social networks. In: Portuguese conference on artificial intelligence. Springer, pp 476–490

Papalexakis EE (2016) Automatic unsupervised tensor mining with quality assessment. In: Proceedings of the 2016 SIAM international conference on data mining. SIAM, pp 711–719

Papalexakis EE, Faloutsos C (2015) Fast efficient and scalable core consistency diagnostic for the parafac decomposition for big sparse tensors. In: ICASSP, pp 5441–5445

Papalexakis EE, Faloutsos C, Sidiropoulos ND (2012) Parcube: sparse parallelizable tensor decompositions. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp 521–536

Papalexakis EE, Sidiropoulos ND, Bro R (2013) From k-means to higher-way co-clustering: multilinear decomposition with sparse latent factors. IEEE Trans Signal Process 61(2):493–506

Papalexakis EE, Faloutsos C, Sidiropoulos ND (2016) Tensors for data mining and data fusion: models, applications, and scalable algorithms. ACM Trans Intell Syst Technol 8(2):1–44

Papalexakis E, Pelechrinis K, Faloutsos C (2014) Spotting misbehaviors in location-based social networks using tensors. In: Proceedings of the 23rd international conference on world wide web. ACM, pp 551–552

Park N, Jeon B, Lee J, Kang U (2016) Bigtensor: mining billion-scale tensor made easy. In: Proceedings of the 25th ACM international on conference on information and knowledge management. ACM, pp 2457–2460

Pasricha R, Gujral E, Papalexakis EE (2018) Identifying and alleviating concept drift in streaming tensor decomposition. arXiv preprint arXiv:180409619

Pavlopoulos GA, Secrier M, Moschopoulos CN, Soldatos TG, Kossida S, Aerts J, Schneider R, Bagos PG (2011) Using graph theory to analyze biological networks. BioData Min 4(1):1–27

Peng W, Li T (2011) Temporal relation co-clustering on directional social network and author-topic evolution. Knowl Inf Syst 26(3):467–486

Phan AH, Cichocki A (2011) Parafac algorithms for large-scale problems. Neurocomputing 74(11):1970–1984

Priebe CE, Conroy JM, Marchette DJ, Park Y (2005) Scan statistics on enron graphs. Comput Math Organ Theory 11(3):229–247

Ranshous S, Shen S, Koutra D, Harenberg S, Faloutsos C, Samatova NF (2015) Anomaly detection in dynamic networks: a survey. Wiley Interdiscip Rev Comput Stat 7(3):223–247

Rayana S, Akoglu L (2014) An ensemble approach for event detection and characterization in dynamic graphs. In: Proceedings of the 2nd ACM SIGKDD workshop on outlier detection and description under data diversity (ODD)

Rossetti G, Cazabet R (2018) Community discovery in dynamic networks: a survey. ACM Comput Surv (CSUR) 51(2):35

Sapienza A, Panisson A, Wu J, Gauvin L, Cattuto C (2015) 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, Porto. ECML PKDD, Portugal

Shah N, Koutra D, Zou T, Gallagher B, Faloutsos C (2015) Timecrunch: interpretable dynamic graph summarization. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 1055–1064

Shashua A, Hazan T (2005) Non-negative tensor factorization with applications to statistics and computer vision. In: Proceedings of the 22nd international conference on Machine learning. ACM, pp 792–799

Sheikholeslami F, Giannakis GB (2017) Identification of overlapping communities via constrained egonet tensor decomposition. arXiv preprint arXiv:170704607

Shi L, Gangopadhyay A, Janeja VP (2015) Stensr: spatio-temporal tensor streams for anomaly detection and pattern discovery. Knowl Inf Syst 43(2):333–353

Sidiropoulos ND, De Lathauwer L, Fu X, Huang K, Papalexakis EE, Faloutsos C (2017) Tensor decomposition for signal processing and machine learning. IEEE Trans Signal Process 65(13):3551–3582

Spiegel S, Clausen J, Albayrak S, Kunegis J (2011) Link prediction on evolving data using tensor factorization. In: New frontiers in applied data mining. Springer, pp 100–110

Sun J, Tao D, Papadimitriou S, Yu PS, Faloutsos C (2008) Incremental tensor analysis: theory and applications. ACM Trans Knowl Discov Data 2(3):11:1–11:37

Sun J, Papadimitriou S, Lin CY, Cao N, Liu S, Qian W (2009) Multivis: content-based social network exploration through multi-way visual analysis. In: Proceedings of the 2009 SIAM international conference on data mining. SIAM, pp 1064–1075

Sun J, Tao D, Faloutsos C (2006) Beyond streams and graphs: dynamic tensor analysis. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 374–383

Tabassum S, Pereira FSF, Fernandes S, Gama J (2018) Social network analysis: an overview. Wiley Interdiscip Rev Data Min Knowl Discov 8(5):e1256

Tan H, Feng G, Feng J, Wang W, Zhang YJ, Li F (2013a) A tensor-based method for missing traffic data completion. Transp Res C Emerg Technol 28:15–27

Tan H, Feng J, Feng G, Wang W, Zhang YJ (2013b) Traffic volume data outlier recovery via tensor model. Math Probl Eng 2013. https://doi.org/10.1155/2013/164810

Tang J, Sun J, Wang C, Yang Z (2009) Social influence analysis in large-scale networks. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 807–816

Timmerman ME, Kiers HA (2000) Three-mode principal components analysis: choosing the numbers of components and sensitivity to local optima. Br J Math Stat Psychol 53(1):1–16

Tsalouchidou I, Bonchi F, Morales GDF, Baeza-Yates R (2018) Scalable dynamic graph summarization. IEEE Trans Knowl Data Eng 32(2):360–373

Tucker LR (1966) Some mathematical notes on three-mode factor analysis. Psychometrika 31(3):279–311

Vanhems P, Barrat A, Cattuto C, Pinton JF, Khanafer N, Régis C, Ba K, Comte B, Voirin N (2013) Estimating potential infection transmission routes in hospital wards using wearable proximity sensors. PloS ONE 8(9):e73970

Vervliet N, Debals O, Sorber L, Van Barel M, De Lathauwer L (2016) Tensorlab 3.0. https://www.tensorlab.net

Viswanath B, Mislove A, Cha M, Gummadi KP (2009) On the evolution of user interaction in facebook. In: Proceedings of the 2nd ACM SIGCOMM workshop on social networks (WOSN'09)

Von Luxburg U (2007) A tutorial on spectral clustering. Stat Comput 17(4):395–416

Wang P, Xu B, Wu Y, Zhou X (2015) Link prediction in social networks: the state-of-the-art. Sci China Inf Sci 58(1):1–38

Xie K, Li X, Wang X, Xie G, Wen J, Cao J, Zhang D (2017) Fast tensor factorization for accurate internet anomaly detection. IEEE/ACM Trans Netw 25(6):3794–3807

Xu J, Chen H (2005) Criminal network analysis and visualization. Commun ACM 48(6):100–107

Zou B, Li C, Tan L, Chen H (2015) Gputensor: efficient tensor factorization for context-aware recommendations. Inf Sci 299:159–177