

# Visualization of Evolving Social Networks using Actor-Level and Community-Level Trajectories

Márcia Oliveira<sup>†,\*</sup> and João Gama<sup>†,\*</sup>

<sup>†</sup>FEP, School of Economics and Business, University of Porto, Rua Dr. Roberto Frias,  
4200-464 Porto, Portugal, [mdbo@inescporto.pt](mailto:mdbo@inescporto.pt), [jgama@fep.up.pt](mailto:jgama@fep.up.pt)

<sup>\*</sup>LIAAD/INESC TEC, Campus FEUP, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal

## Abstract

Visualization of static social networks is a mature research field in information visualization. Conventional approaches rely on node-link diagrams which provide a representation of the network topology by representing nodes as points and links between them as lines. However, the increasing availability of longitudinal network data has spurred interest in visualization techniques that go beyond the static node-link representation of a network. In temporal settings, the focus is on the network dynamics at different levels of analysis (e.g. node, communities, whole network). Yet, the development of visualizations that are able to provide actionable insights into different types of changes occurring in the network and their impact on both the neighborhood and the overall network structure is a challenging task. In such settings, traditional node-link representations can prove to be limited (Yi et al., 2010). Alternative methods, such as matrix graph representations, fail in tasks involving path finding (Ghoniem et al., 2005). This work attempts to overcome these issues by proposing a methodology for tracking the evolution of dynamic social networks, at both the node-level and the community-level, based on the concept of temporal trajectory. We resort to three-order tensors to represent evolving social networks and we further decompose them using a Tucker3 model. The two most representative components of this model defines the 2D space where the trajectories of social entities are projected. To illustrate the proposed methodology we conduct a case study using a set of temporal self-reported friendship networks.

**Keywords:** Dynamic Social Networks, Multiway Analysis, Network Visualization, Trajectories, Tucker3 model.

## 1 Introduction

A social network is constructed from relational data and can be defined as a set of social entities, such as people, groups and organizations, with some pattern of interaction between them. These networks are usually modeled by mathematical graphs, where vertices (aka nodes or actors) represent the social entities and edges (aka links or ties) represent the ties established between them. This kind of representation is also known as *node-link diagram*. Other common ways of representing a network are based on both matrix structures (e.g. adjacency matrices) or list structures (e.g. edge lists). The underlying structure of such networks is the object of study of *Social Network Analysis* (henceforth SNA). Hence, the focus of SNA is on the relationships established between social entities rather in the social entities themselves.

Until recently, this type of analysis was mainly a static investigation by considering independent graphs at different snapshots of time, or one aggregated graph over the time period (Takaffoli et al., 2010). Nonetheless, one of the key features of a wide range of real-world social networks is that their structure evolves over time, through the addition or deletion of nodes and edges, so approaches focusing on the analysis of a fixed snapshot of the network may fail to capture the dynamics of the evolving network. These dynamics are a reflection of frequent changes in the interaction among social entities and translate into rich evolutionary patterns. By becoming more aware of this important feature of social networks, researchers have devoted themselves to the development of visualization methods and tools for the analysis of social networks that take into account temporal information (Moody et al., 2005). This trend arose mainly as a consequence of the growing interest in evolving social networks and the recognition that visualization of time-oriented data gleans insights into the dynamics of the underlying

phenomena, allowing for a better understanding of temporal relations. However, the development of a visualization that facilitates the understanding and analysis of the dynamics of a network in a compact way is a challenging task. In order to produce a good visualization, it is important to detect not only when the changes occur but also the nature of those changes with respect to social network constructs, such as popularity and reachability.

Following this trend, this paper attempts to study the structure of evolving social networks at two different levels of analysis: actor-level and community-level, using as a basis the concept of *temporal trajectory*. In order to meet the above-mentioned requirements, we propose the generation of trajectories of social entities in a bidimensional space (2D), which accounts for most variation in the original data, based on the output of a Tucker3 model with orthogonality constraints. These social entities can be actors or groups of actors that interact more often with each other (also known as *communities*). Similarly to Ahn et al. (2011), we use the term *social entities* to refer to the objects analysts are interested in (e.g. nodes or communities). By allowing the visualization of different types of social entities, our approach is able to support the analysis of the temporal activity of networks at multiple granularities.

As an example, consider a group of students entering college and facing a new social environment. While a few students may find acquaintances and friends from high-school attending the same college, it is likely that most of them do not know their classmates. When facing this kind of social contexts, individuals tend to invest more effort in communication and in trying to establish relationships with their peers. This socialization process is expected to be more intense during the first weeks of college, which typically correspond to periods of social exploration and adaptation. However, after some time the friendship groups start to settle down and the intensity of students' socialization tends to stabilize. Since the relationships established among college freshmen over time can be mapped as an evolving social network, this socialization phenomenon can be investigated by means of the proposed visualization. This visualization would not only provide an overview of the general evolution of the social behavior of the whole set of students, but would also allow to delve into questions concerned with the social dynamics of each student (e.g. changes in popularity and the moments when they occurred) or with the evolution of each group of students (e.g. identification of temporally cohesive groups and/or highly dynamic groups and examination of the corresponding reasons).

The main contribution of this paper is a methodology for the visualization of evolving networks that relies on displaying actor-level (or community-level) trajectories in a concise space. While the proposed visualization focuses on observing trajectories of single actors (or single communities), some of the metrics used to describe those actors depend on the global network structure, thus allowing to capture latent network properties in the 2D visualization. This way, the trajectories are able to encapsulate both temporal and relational aspects of the network dynamics by indicating when a global or local change occurred and the meaning of such change. The use of Tucker3 model to map the information into a 2D space guarantees that this space describes the most important information contained in the original three-way data while preserving the temporal interdependencies. On the other hand, since each axis corresponds to a component of the Tucker3 model, this space is interpretable in terms of the network metrics used to describe the social entities comprising the network.

The remainder of this document is organized as follows. In Section 2 we begin by introducing the preliminaries of tensor algebra and the foundations of Tucker3 decomposition. In Section 3 we address the problem of representing dynamic social networks and introduce our approach to visualize their evolution using temporal trajectories. In Section 4 we show and discuss a case study using temporal friendship networks and in Section 5 we briefly present some related work. Section 6 concludes the paper and discusses possible future research directions.

## 2 Preliminaries

In this section, we introduce some preliminary concepts, terminology and notation of tensor algebra, as well as the foundations of Tucker3 model, which will be useful to understand the proposed methodology.

### 2.1 Tensors

#### 2.1.1 The Concept of Tensor

Traditional data analysis techniques, such as Regression, Principal Component Analysis (henceforth PCA) and Clustering, were devised to extract relevant knowledge from two-way (aka two-mode or two-order) data. Two-order data is usually represented in matrix form, with rows corresponding to the objects and

columns to the variables. Although two-order data constitutes the basis of numerous and interesting analyses, in most of cases it can only provide a static view of the world, for a specific point in time. Since many phenomena are inherently multidimensional and dynamic, in several settings one should adopt data representation schemes able to model simultaneously all dimensions, including the temporal one. In such cases, high-order *tensors* (also known as *hypermatrices*, *multiway models*, *multiway arrays* or *multidimensional arrays*) appear as more natural and appropriate data representations than matrices, since they are able to explicitly model a higher number of dimensions (e.g. objects, variables and time) without collapsing the data and, therefore, without losing information about their mutual dependencies.

One of the desirable features of modeling three-mode data as tensors and explore techniques especially devised to deal with these data structures, is the possibility of preserving all mutual dependencies established between the dimensions of different modes.

### 2.1.2 Tensor Notation

Regarding notation, we follow the typical conventions and, in this paper, we use the standardized notation and terminology for multiway analysis as proposed by Kiers (2000). As previously mentioned, a tensor is a  $N$ -way data array, where  $N$  is the *order* (also referred to as *ways*) of the tensor. For instance, a three-order tensor ( $N = 3$ ) encapsulates three modes: the row-entities mode (mode  $A$ ), the column-entities mode (mode  $B$ ) and the fiber-entities mode (mode  $C$ ). We will then use the term *mode* to refer to a set of entities. High-order tensors, denoted by calligraphic letter  $\mathcal{X}$ , are generalizations of scalars (order 0), vectors (order 1) and matrices (order 2) to 3 or higher orders. The element  $(i, j, k)$  of a three-order tensor  $\mathcal{X}$  is denoted by  $x_{ijk}$ , where index  $i$  ( $j$  and  $k$ , respectively) refers to the entities of mode  $A$  (mode  $B$  and mode  $C$ , respectively). Indexes typically range from 1 to their capital version:  $i = 1, \dots, I$ ,  $j = 1, \dots, J$  and  $k = 1, \dots, K$ .

## 2.2 Tucker3 Model

Three-way methods are unsupervised multivariate data analysis tools that compress simultaneous variation of combinations of variables and entities (Smilde, 1992). These methods are able to summarize the information contained in three-order tensors, both in terms of main effects and two and three-way interactions among the modes, by means of a few number of components. Besides, these methods generate the so-called *core tensor* (or *core array*), which is a highly informative structure that captures and describes the relations and three-way interactions between the different modes of data, in terms of their summarized entities (Kiers and Van Mechelen, 2001). For a thorough review on unsupervised multiway analysis, please refer to Acar and Yener (2009).

The most widely known three-way methods are the CANDECOMP/PARAFAC (CP) (Carroll and Chang, 1970; Harshman, 1970) and the Tucker3 decomposition (Tucker, 1963, 1966). In this article we focus on Tucker decomposition since it is more flexible, easier to interpret (in the sense that the solution can be rotated) and has less constraints (in fact, the CP model can be seen as a constrained variant of the three-way Tucker model, where the core tensor is superdiagonal).

Tucker (1963) introduced the tensor decomposition, which inherits his name, in 1963. Refinements of this model were then performed by Levin (1963) and Tucker (1966). An in-depth study of Tucker decomposition was undertaken by De Lathauwer (2000), which coined its orthogonality-constrained version as High-Order Singular Value Decomposition (HOSVD). We briefly introduce the foundations of this decomposition following closely the definitions provided by Tucker (1966), Kroonenberg (1983), De Lathauwer (2000), Skillicorn (2007) and Kolda and Bade (2009).

The Tucker decomposition can be thought as a form of higher-order principal component analysis or as a multilinear generalization of Singular Value Decomposition (henceforth SVD). The three-way version of this decomposition is usually called *Tucker3 model*. The term derives from the fact that the reduction of data is performed in all three modes of the tensor (in our case, network entities mode, network metrics mode and time mode). The general Tucker3 model can be written elementwise as:

$$\hat{x}_{ijk} = \sum_{p=1}^P \sum_{q=1}^Q \sum_{r=1}^R a_{ip} b_{jq} c_{kr} g_{pqr} \quad (1)$$

where  $i = 1, \dots, I$ ,  $j = 1, \dots, J$  and  $k = 1, \dots, K$ . Here, the coefficients  $a_{ip}$ ,  $b_{jq}$  and  $c_{kr}$  represent the entries of orthonormal matrices, also referred to as *component matrices*  $\mathbf{A} \in \mathbb{R}^{I \times P}$ ,  $\mathbf{B} \in \mathbb{R}^{J \times Q}$  and  $\mathbf{C} \in \mathbb{R}^{K \times R}$ . These matrices can be thought as the principal components in each mode. In turn, the coefficient  $g_{pqr}$  represents the entry of the so-called *core tensor*  $\mathcal{G} \in \mathbb{R}^{P \times Q \times R}$ . The number of entities in each mode are

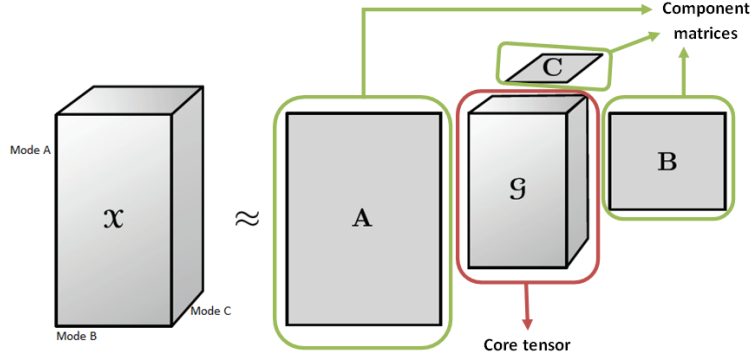


Figure 1: The Tucker3 decomposition of a three-order tensor (adapted from (Kolda and Bade, 2009)).

represented by letters  $I$ ,  $J$  and  $K$ . The number of components, or levels (ie number of columns of the matrices  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$ ) in the first, second and third mode of the tensor are represented by letters  $P$ ,  $Q$  and  $R$  (usually,  $P < I$ ,  $Q < J$  and  $R < K$ ), respectively. These components summarize the entities of the corresponding mode. We can see the *core tensor*  $\mathcal{G}$  as a compressed version of the original tensor  $\mathcal{X}$  if  $P$ ,  $Q$  and  $R$  are smaller than  $I$ ,  $J$  and  $K$ . Tucker suggested interpreting the *core tensor* as describing the latent structure in data, since it has information about the level of interaction between the different components, and the *component matrices* as mixing this structure to give the observed data (Tucker, 1966). The *core* values can also be interpreted as a generalization of the eigenvalues, or of the singular values of SVD and it constitutes a further partitioning of the "explained" variation as is indicated by the eigenvalues of the standard PCA. In fact, when the component matrices are orthogonal, the square of each entry of  $\mathcal{G}$  is proportional to the amount of variance explained by the corresponding combination of components, in terms of total fit. Besides, the sign and magnitude of the core values provide clues about how the various components relate to each other. Matrices  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$  are assumed to be columnwise orthogonal. The orthogonality is desirable since it facilitates the analysis and hastens the computation of the decomposition.

The basic idea of the tensor decomposition proposed by Tucker (1966) is to find those components that best capture the variation in each mode  $n$ . Or, in other words, the goal of Tucker's method is to find a set of matrices  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$ , and a small tensor  $\mathcal{G}$  that, in general, have less dimensionality than the original tensor, but are able to reconstruct the most important information contained in data, as depicted in Figure 1.

The problem of decomposing a tensor, which can be translated into finding the best estimation of the model presented in Equation 1, can be reduced to a straightforward optimization problem, which is usually solved using an *Alternating Least Squares* (ALS) approach. More detailed information about the ALS algorithm can be found in Kroonenberg (1983).

### 3 Methodology for Visualizing the Evolution of Dynamic Social Networks

In this section we present two alternatives to represent social networks and argue which one is the best for representing time-oriented sociometric data. We also introduce our visualization approach, which is based on the development of trajectories of social actors, or social communities, in low-dimensional spaces. These trajectories are obtained by exploring the compressed information produced by an estimated Tucker3 model.

#### 3.1 Representation of Dynamic Social Networks as Three-order Tensors

In this work, our endeavor is primarily directed towards the production of an accurate, yet compact, visualization approach to represent the evolution of actors, or communities of actors, engaged in a given type of social relation. By compact we mean that this representation is not only able to be grasped by human eye, through projections in low-dimensional spaces, but it is also focused in the structural and relevant information contained in raw sociometric data. This is an important aspect that guided our understanding of how a social network should be represented in order to meet requirements such as interpretability of results. In fact, if we choose to represent a social network, comprised of  $n$  nodes

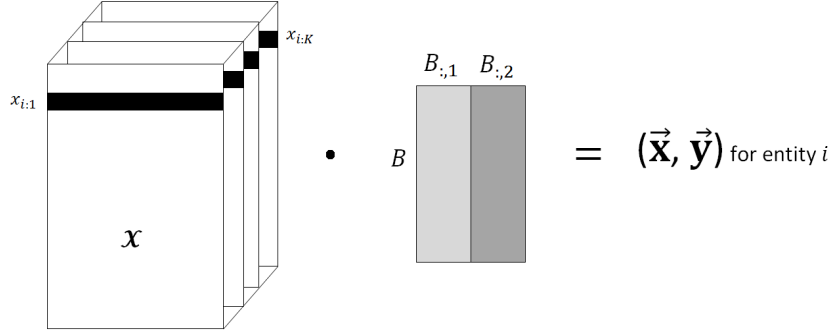


Figure 2: Illustration of the process of generating the coordinates of the trajectory of a given A-mode entity. Being a trajectory defined as a set of pairs  $(x, y)$ , that are linked according to the given temporal order, we obtain each one of these pairwise values by computing the dot product between each horizontal fiber of a given A-mode entity and the first two columns of component matrix  $\mathbf{B}$ . In this example, the trajectory of entity  $i$  will be defined by a set of  $K$  ( $K = 4$ ) coordinates  $(x, y)$ , each one corresponding to a given  $k$ .

(or actors), by means of an adjacency matrix  $\mathbf{A}_{n \times n}$ , and factorize it, we obtain new entities that are difficult to interpret, especially if we are dealing with directed networks, where the rows and the columns represent different concepts. In such cases, where the adjacency matrix is not symmetric, the rows are closely related to the out-neighborhood of social actors and the columns are related to their in-neighborhood. On the other hand, when working with symmetric adjacency matrices, the new entities returned by the decomposition are typically the same for both modes  $A$  and  $B$ , thus creating redundant results and less valuable information than one would expect from rich network structures. To overcome the barriers posed by the adoption of a standard social network representation, we suggest computing SNA metrics (e.g. degree, eigenvector centrality, closeness, betweenness and clustering coefficient), to embed richer structural knowledge into the *snapshot matrices* (these matrices represent snapshots of the state of the world for a specific point in time). In the context of social networks, these metrics are useful in the sense that they provide us higher-level information regarding the network structure, thus giving us insight about the position and importance of each node/community in the network. Moreover, this strategy helps mitigate the differences between directed and undirected networks and it greatly improves the interpretability of the component matrices yielded by factorial methods.

After using this strategy to build the snapshot matrices, the process of converting them into a tensor becomes trivial. To do so, one just needs to introduce an additional mode  $C$ , by ordering them by time and putting them together into a unique structure, thus obtaining a three-order tensor  $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ , where  $I$  ( $i = 1, \dots, I$ ) denotes the number of entities of the first mode  $A$  (the *row-entities*, which are defined along the horizontal axis),  $J$  ( $j = 1, \dots, J$ ) refers to the number of entities of the second mode  $B$  (the *column-entities*, which are defined along the vertical axis) and  $K$  ( $k = 1, \dots, K$ ) indicates the number of entities of the third mode  $C$  (the *fiber-entities*, which are defined along the depth axis). In the social networks context, the mode  $A$  is the dimension of social actors, or social communities, mode  $B$  is the dimension of actor-level network metrics and mode  $C$  is the dimension of time.

## 3.2 Trajectories in Social Networks

A trajectory can be defined as a sequence of time-stamped points  $Traj = p_0 \rightarrow p_1 \rightarrow \dots \rightarrow p_k$ , where  $p_i = (x_i, y_i, t_i)$  ( $i = 1, 2, \dots, k$ ) represents the position of a given object at time point  $t_i$ , and  $(x_i, y_i)$  are the coordinates of the object, in a 2D space. Typically, these trajectories are defined in low-dimensional representative spaces and are graphically represented by a line that connects the coordinates of an object for different time points. It is common to resort to 2D, instead of 3D spaces, since they are simpler to analyse and, at the same time, allow for an effective data analysis. Thus, we use two-dimensional projections and encode the third dimension as a trajectory over the plane. In such way, we are able to map a given actor's, or community, trajectory along time, by simply using two-dimensional projections.

### 3.2.1 Actor-Level Trajectories

In order to generate the trajectories of each actor we decompose the original three-order tensor by estimating a Tucker3 model. Afterwards, we consider the bidimensional space spanned by the two most

representative components of matrix  $\mathbf{B}$ , as given by the entries of the core tensor  $\mathcal{G}$ , and define the  $x$  and  $y$  coordinates for each time point  $k$  ( $k = 1, \dots, K$ ) of the trajectory. We obtain these coordinates for each actor  $i$  ( $i = 1, \dots, I$ ), by computing the *dot product* between  $x_{i,:k}$  (horizontal fibers of  $\mathcal{X}$ ) and each column of component matrix  $\mathbf{B}$  (the two components that explain the highest portion of the total data variance are assigned to the  $x$ -axis and  $y$ -axis, respectively). This vector operation returns the coordinates, or the bidimensional position vector, of the time points, for each actor of the network. The process for generating trajectories is depicted in Figure 2.

The last step of this analysis is to interpret the social micro-evolution through critical observation of an entity’s trajectory. In this paper, we use the term *social micro-evolution* to refer to the study of the temporal behavior of a social network actor and, analogously, we use the term *social macro-evolution* to refer to the study of evolving communities of actors. The movement, or trajectory, of a given actor can be characterized by a *direction* (e.g. upwards, downwards, leftwards, rightwards), which can be more regular or more irregular; and by an *amplitude*, which can be higher, thus covering a larger space area, or lower, by keeping its position in the plane almost unchanged over time. Also, by computing the distance between pairs of trajectories, it is possible to identify actors with similar/dissimilar trajectories. We will take these features into consideration when analysing the trajectories in the case study.

In short, we can say that our visualization approach maps each snapshot of the social network of a given entity (actor or community) into a point in the Tucker decomposition space, and links these points in order to define a trajectory that represents the dynamic behavior of this single entity. Using this strategy it is possible to naturally take temporal information into account by adopting tensorial representations of social networks.

Our approach also holds when dealing with large social networks. Sun et al. (2008) proposed an incremental algorithm to compute Tucker decomposition for large tensors. To estimate the adequate number of components of the Tucker3 model, instead of using heuristic methods, such as DIFFIT (Timmerman and Kiers, 2000), one can use more efficient methods, such as the ARD framework (Morup and Hansen, 2009).

Regarding the computational complexity, using the Tucker-ALS algorithm, it has been shown by Oseledets et al. (2009) to be  $\mathcal{O}(n^3r + n^2r^2 + nr^3)$ , where  $n$  is the mode size<sup>1</sup> and  $r$  is the number of retained components (assuming  $r = P = Q = R$ ).

### 3.2.2 Community-Level Trajectories

The previous analysis can also be extended to deal with communities of actors. The need for studying the evolution of communities, instead of actors, gains importance especially when dealing with large social networks. In such scenarios, studying the evolution of all nodes in a network can be a cumbersome task. So, unless one is interested in analysing one specific actor, changing the level of analysis to communities can provide a higher-level view of the dynamics occurring in the network. Notwithstanding, both types of trajectories can be used as complements in the analysis of evolving social networks, rather than substitutes.

The use of communities in visualization approaches, instead of single actors, is justified by its importance in the SNA field. In fact, one of the unique features of real social networks is that they tend to show *community structure* (Newman, 2003). This property usually arises as a consequence of both global and local heterogeneity of edges’ distribution in a graph. Thus, we often find high concentrations of edges within certain regions of the graph, that we call *communities*, and low concentration of edges between those regions. *Communities*, also known as *modules* or *clusters*, can be understood as densely connected groups of nodes in the network, with sparser connections between them (Newman and Girvan, 2004). Society is a rich environment for finding such communities, once people have the natural tendency to form tight groups (e.g. circles of friends, families, working/religious groups).

The importance of studying these communities is intuitive in domains such as SNA. To highlight this importance, Fortunato (2010) states that the analysis of the structural position of actors, in each community, can help identify central actors, often associated to group control and stability functions, as well as intermediate actors, who are those who lie at the boundaries of communities and play a key role in the spread and exchange of new ideas and information. This importance becomes more apparent when studying the dynamics of social networks since communities are unstable patterns that can evolve in both membership and content. In dynamic scenarios, communities may undergo a series of significant evolutionary events, such as splits, merges, survival, deaths and births that characterize their *life-cycle* (Palla et al., 2007; Greene et al., 2010). This has implications in the way trajectories are defined. For

<sup>1</sup>Note that if the size of, at least, one mode, differs from the size of the others,  $n$  will be regarded as  $n = \max \{I, J, K\}$ .

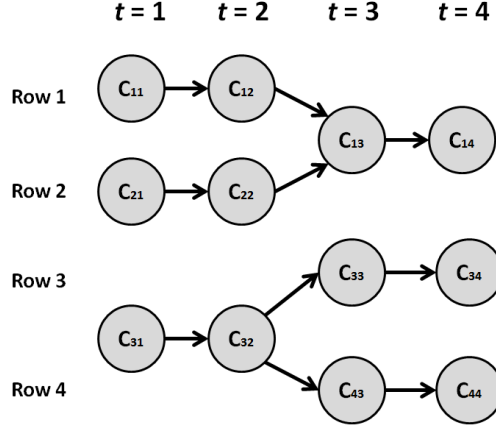


Figure 3: Example of three dynamic communities tracked over four time steps, featuring merging and splitting life-cycle events (adapted from Greene et al. (2010)).

instance, the trajectory of a community at time point  $t_i$  may separate into several trajectory paths in time point  $t_{(i+1)}$  if the initial community splits into two or more communities. To capture these events we resort to an adapted version of MEC (Monitor of the Evolution of Clusters) framework (Oliveira and Gama, 2012), and we incorporate this information in the process of generating the trajectories. The output of the adapted version of MEC provides the succession of events (e.g. birth, split, merge, survival, death) undergone by each community, from the moment the community first appeared until the moment it faded away. In Figure 3, we show a visual example of the life-cycle of three communities for a time span of four time points.

The task of *community detection*, which aims at finding meaningful group structures in networks, is itself an important strand of research on the field of SNA and a significant number of methods and algorithms have been proposed for this purpose (for a thorough review, please refer to Fortunato (2010)). In our approach we propose to detect communities in raw social networks using the so-called *Louvain method* (Blondel et al., 2008). This is a greedy optimization method that performs a hierarchical modularity optimization. This method comprises two phases. The first phase optimizes modularity in a local way and encompasses the following steps: first, and similar to the procedure of a bottom-up hierarchical clustering, it considers each node as a distinct community; then, for each node, it looks for positive gains in modularity by moving the isolated node from its community to a neighboring community; this search for a local optima is sequentially and repeatedly applied until no further improvements are possible. At the end of this phase, one obtains a first level partition. The second phase is similar to the first one, with the difference that now we deal with a modified network, where each vertex is a supervertex, since it represents the previously found communities (each node is, therefore, an aggregation of the original nodes); two supervertices are connected by an edge if there is at least one edge linking two vertices inside the corresponding community. Considering this higher level setting, the steps of the first phase are repeated iteratively until a maximum of modularity is attained and new hierarchical levels and supergraphs are yielded. The algorithm stops when modularity converges to a value where no more gains are possible. The main advantages of this method is being parameter-free and very fast. In fact, the *Louvain method* is able to extract good quality communities from large networks in a fast way and with low computational cost. One of its drawbacks is being a order-sensitive algorithm, returning different results according to the order of the vertices in the sequential analysis of modularity gains.

Due to its benefits, in this study we resort to *Louvain method* to identify the communities of actors. In order to obtain the trajectories, we build a new three-order tensor, where each entity of mode  $A$  is now a community (instead of an actor) and  $I$  ( $i = 1, \dots, I$ ) is given by the sum of the maximum number of sub-communities (mostly resulting from splitting events) sequentially derived from the life-cycle of each community. We define  $I$  this way in order to guarantee the consistency of the row-entities. Consistency is observed whenever each row represents the same entity, regardless of the time point we are considering. For the example provided in Figure 3, the maximum number of sub-communities in the whole time span for community  $C_{11}$  would be 1, which means a single row would be assigned to it. The same applies for community  $C_{21}$ . However, for the case of community  $C_{31}$ , two rows would be needed to represent it, since this community undergoes a split in the third time point. Each row would then describe the two different trajectory paths taken by this evolving community. Thus, in this case, the number of rows of

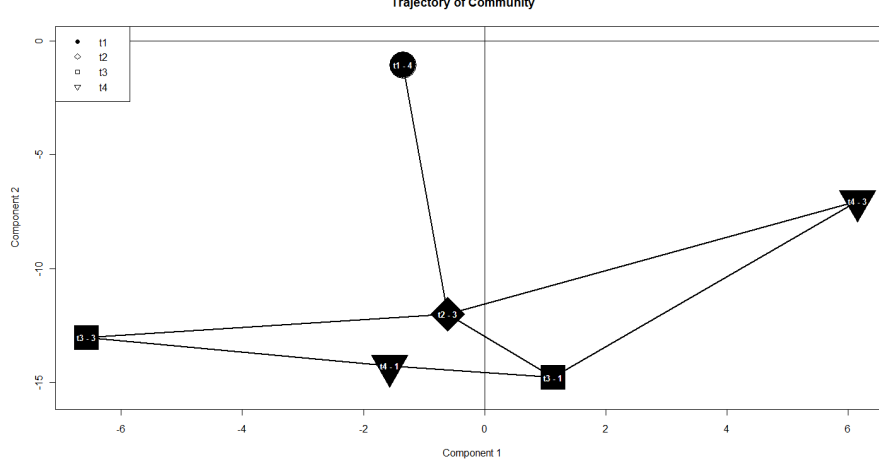


Figure 4: Example of a trajectory of an evolving community of social actors, for a time horizon of four time points  $T = \{t_1, t_2, t_3, t_4\}$ , in the space spanned by the two most representative components of matrix  $\mathbf{B}$ . The *life cycle* of this community is  $LF = \{C4_{t1}; C3_{t2}; C1_{t3}; C1_{t4} \& C3_{t4}; C3_{t3}; C1_{t4}\}$ .

the tensor would be  $I = 1 + 1 + 2 = 4$ . Mode  $B$  and Mode  $C$  refer to the actor-level metrics and to the time points, respectively. The only difference is that now the community will be described by the average of the actor-level metrics of all actors that were assigned to it. After building this new data structure, we obtain the coordinates of each community by following the same procedure described to obtain the actor-level trajectories. When using both analyses as complements, it is advisable to represent the trajectories of communities in the same 2D space where the actor-level trajectories were represented. For this purpose one just needs to keep the same component matrix  $\mathbf{B}$  and use this new three-order tensor to compute the dot product. Then, the returned coordinates are projected in the bidimensional space spanned by the two most representative components of the mentioned matrix, in order to define the community-level trajectories. The interpretation of the social macro-evolution is also identical to the one described for social micro-evolution. We provide an illustrative example of the trajectory of one community in Figure 4.

## 4 Social Networks Case Study: Actor-Level Trajectories

In this section we conduct a small case study using a set of friendship social networks collected at seven different moments in time, for the same individuals. The idea is to verify the suitability of the proposed visualization approach for representing the dynamics of social networks and, at the same time, to yield meaningful insights into social micro-evolution (at the actor-level). To do so, we perform the following sequential steps: first, we select and compute the actor-level metrics; then, we model the resulting matrices as a three-order tensor, where the third dimension is time; the third and fourth steps consist in preprocessing the data and applying the Tucker3 decomposition to this tensor; finally, the 2D coordinates of each entity’s trajectory are computed and interpreted based on the results of the decomposition. Note that this case study serves to demonstrate the applicability of the introduced methodology, not to support specific theoretical claims.

### 4.1 Data Description

The data we use comprise several friendship networks, collected by Van de Bunt et al. (1999) among a group of university freshmen, in order to provide an explanation of some of the important factors behind friendship formation. This information was gathered by questionnaires which were delivered to 49 students in seven different occasions asking them to rate their social relationships in a six point scale. The mentioned occasions are not equally spaced in time, since the first four time points are three weeks apart, and the last three time points are six weeks apart. There were also changes in the number of students, which drops from 49 to 32, due to university “drop-outs” and non-response of questionnaires in at least four of the total seven occasions. It is also important to note that, with a few exceptions, the respondents were initially mutual strangers (freshmen), which reflects in a sparse adjacency matrix



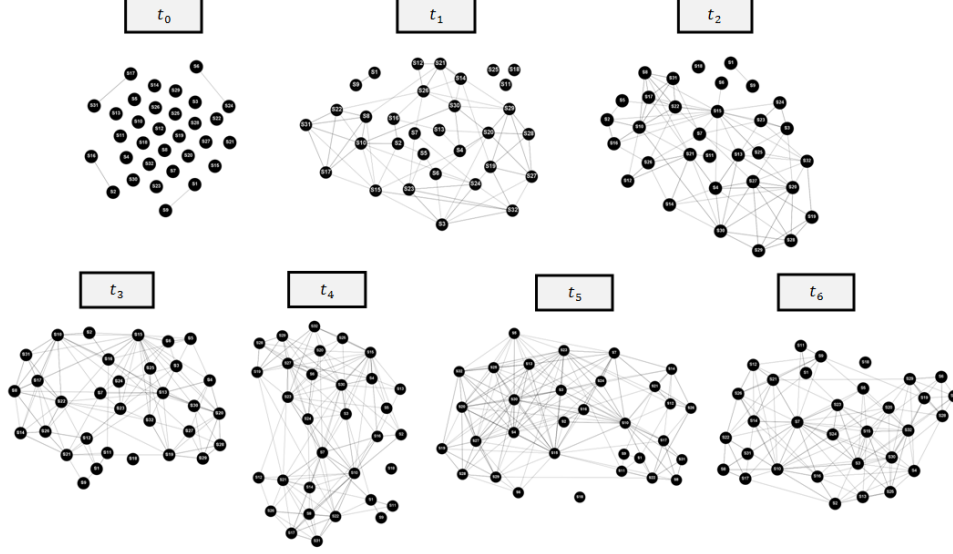


Figure 5: Conventional graph representation of the static snapshots of the modified versions of Van de Bunt's friendship networks. Each graph maps the network detailing the positively connoted relationships among a set of freshman students at a given time point.

for the first measurement ( $time = t_0$ ). Originally, the ties linking each pair of students were weighted according to a specific coding scale, representing the rating of the established relationships. For the sake of simplicity, and in order to improve the interpretability of the trajectories, we transform the initial network into an unweighted one, by replacing codes 6 = *item non-response*, 9 = *actor non-response*, 4 = *neutral* and 5 = *troubled relation* by 0 and codes 1 = *best friend*, 2 = *friend* and 3 = *friendly relation* by 1. Therefore, the absence of a tie, coded by 0, represents either the absence of relationship between a given pair of students, or the lack of intention to establish a friendship on the part of, at least, one of the students. The existence of a tie, coded by 1, means that there is a positively connoted relationship between the corresponding pair of students, which reflects in a stronger willingness to interact with each other. Formally, each entry of the adjacency matrix  $\mathbf{X}$ , generated in time point  $t_k$  ( $t = 1, \dots, K$ ), can assume the following values:  $x_{ij} \in \{0, 1\}$ . The graph representation of the modified version of these friendship networks is depicted in Figure 5. Besides sociometric data, also additional information about students' gender, smoking behavior and education program is available. These variables can prove to be useful to achieve a further macro understanding of the results.

The main reason behind the choice of this data is the fact that it includes a temporal dimension, which makes it possible to track changes over time in students' social behavior.

## 4.2 Metrics Selection, Preliminary Tests, Preprocessing and Tucker3 Model Generation

In order to obtain the trajectories for each freshman student, we first need to transform the initial adjacency matrices into snapshot matrices embedded with structural information about the individuals. To do so, one needs to decide which network metrics should be used to characterize these individuals. In this case study, we resort to the normalized versions of the following actor-level metrics. Notice however that the proposed framework is not constrained by this specific set of metrics and can incorporate other types of variables, contingent on both the goal of the analysis and the nature of the networks at hand. Here we provide a brief description of the metrics within the context of the students' friendship networks:

- **Degree:** indicates the number of affective relationships a given actor states he/she has;
- **Eigenvector Centrality:** ascribes a relative score to each actor and measures how well a given actor is connected to other well-connected actors in the network;
- **Closeness:** measure of reachability that measures how fast can a given actor reach everyone in the network;

Table 1: Three-way ANOVA of Van de Bunt data, after subtraction of the grand mean, with *students*, *actor-level metrics* and *occasions* (or time) as fixed factors.

Effects	Sum of Squares	Percentage
Students	4.48	5.5%
Actor-level metrics	37.78	<b>46.4%</b>
Occasions	5.46	6.71%
Students $\times$ Actor-level metrics	8.3	10.19%
Students $\times$ Occasions	5.09	6.25%
Actor-level metrics $\times$ Occasions	5.39	6.62%
Students $\times$ Actor-level metrics $\times$ Occasions + Error	14.92	<b>18.33%</b>

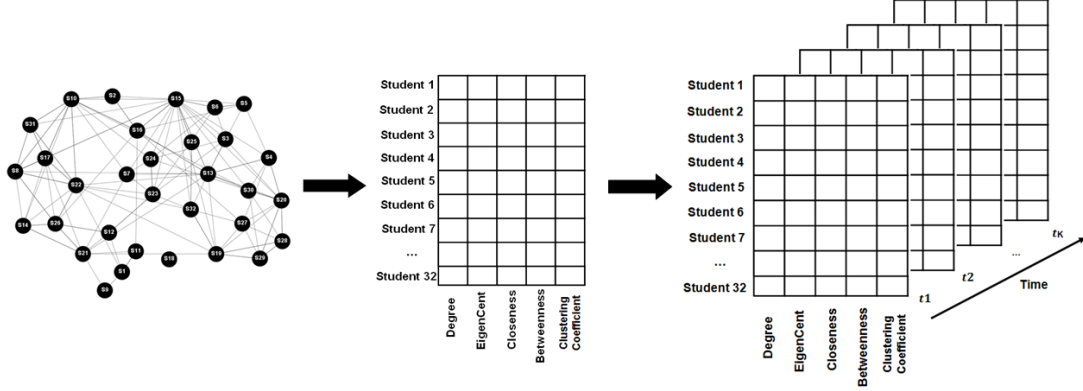


Figure 6: Illustration of the process of converting a social network that evolves over time into a three-order tensor. Given a snapshot of a social network, actor-level metrics are computed for all actors and the resulting data is organized as a matrix. These steps are repeated for all available snapshots of a network. The set of matrices produced by these steps is organized as a three-order tensor.

- **Betweenness:** measures the extent to which actor  $i$  lies between other students in the network. Students with high betweenness occupy critical roles in the network structure, since they usually have a network position that allow them to work as an interface between tightly-knit groups, being "vital" elements in the connection between different regions of the network;
- **Clustering coefficient:** the local version of this metric quantifies the transitivity in the neighborhood of actor  $i$ , by indicating the level of cohesion between his/her neighbors.

Although the proposed visualization focuses on observing trajectories of single actors (or single communities), some of the metrics we use to describe these actors depend on the global network structure (e.g. *eigenvector centrality*, *closeness* and *betweenness*). The use of such measures allows us to capture latent properties of the whole network, in addition to structural information about the actors themselves, when reducing the original measures space into a bidimensional one. As a consequence, the single actors' trajectories that map the temporal social information of actors, will also reflect changes in the global network structure and not only changes at the actor-level. Such global changes can be related to, for instance, the increase/decrease of the number of connections of a given node in the network that is not directly connected to the actor under analysis.

After selecting the metrics we organize the snapshot matrices into a three-order tensor  $\mathcal{X} \in \mathbb{R}^{32 \times 5 \times 7}$ , where the first mode encompasses the 32 students, the second mode refers to the 5 actor-level network metrics and the third mode is related to the 7 different time moments when the questionnaires were applied. Each entry of this tensor gives the score of student  $i$  ( $i = 1, \dots, I$ ) on metric  $j$  ( $j = 1, \dots, J$ ) at measurement occasion  $k$  ( $k = 1, \dots, K$ ). The process of converting the set of social networks into a three-order tensor is represented in Figure 6.

Since three-way analysis is only recommended for cases where the data contain a non-negligible three-way interaction across the three data modes, we carry out a simple fixed effects three-way ANalysis Of VAriance (henceforth ANOVA) on the resulting three-way array  $\mathcal{X} \in \mathbb{R}^{32 \times 5 \times 7}$ , using the three different modes as factors. The goal is to assess the effect size of all variance components, where the *total variance* to be explained is the variance of all data elements with respect to the grand mean (Kiers and Van Mechelen, 2001). This step is important for understanding if a two-way analysis (e.g. PCA, SVD) on data averaged over one of the modes (e.g. the students mode) would suffice to capture the main information contained

in data. Only the cases where a substantial effect size of a three-way interaction (or more than one two-way interaction) is present warrants the use of a three-way analysis (e.g. Tucker3 model). The results of the fixed effects three-way ANOVA for the friendship networks are reported in Table 1. From the analysis of the decomposition into sum of squares of all main effects, two-way interactions and three-way interaction, given by this table, we deduce that the highest contribution derives from a main effect, namely the *actor-level metrics*, followed by a sizable three-way interaction (plus error term) established between all three modes. These results testify to the presence of a substantial three-way interaction that can only be captured through means of a three-way analysis, thus validating the use of a Tucker3 model. On the other hand, the large effect size of the actor-level metrics term suggests the appropriateness of using mode *B*, rather than other modes, to generate the space where the trajectories will be mapped.

After checking if the data at hand legitimizes the conduction of a three-way analysis, we turn to the second step of preprocessing the data. We preprocess the three-way array  $\mathcal{X} \in \mathbb{R}^{32 \times 5 \times 7}$  by first *centering* it across mode *B*, and then *scaling* it within mode *B*. While centering is a projection step that removes the offset terms, scaling adjusts scale differences and accommodates heteroscedasticity. To perform these steps, we follow the guidelines of Bro and Smilde (2003). Thus, centering was done by first rearranging the array into a two-way matrix  $J \times IK$ , and then centering this matrix across mode *B* as in ordinary two-way analysis. Scaling was carried out by multiplying the array columns by a scalar.

The next step of our methodology consists of decomposing the three-order tensor into a small core tensor  $\mathcal{G}$  and a set of component matrices **A**, **B** and **C** that summarize the entities in the three respective modes. To choose among the myriad of possible solutions for Tucker3 decomposition, we created a model for each combination of components ( $P \times Q \times R$ ), where *P*, *Q* and *R* are bounded above by five components (ie the maximum possible order tested was  $(5 \times 5 \times 5)$ ), which is the lowest number of entities in all three tensor modes. The respective fit values were then computed for each possible order. We selected our model using the DIFFIT method proposed by Timmerman and Kiers (2000), which is a variant of the scree test for three-way analyses. This method finds the number of components *P*, *Q* and *R* ( $P < I, Q < J$  and  $R < K$ ), by balancing fit (ie amount of information accounted for) and parsimony (ie relatively low number of components) of the model.

To obtain these components, we estimate an orthogonality-constrained Tucker3 model of order  $(2 \times 3 \times 2)$ , which explains 76.93% of the total data variation. This order is a parameter of the model and refers to the number of components retained in each mode ( $P = 2, Q = 3$  and  $R = 2$ ). We have used the *ThreeWay* package (Giordani et al., 2012) for R environment (Team, 2008) in all the experiments reported here and the *igraph* (Csardi and Nepusz, 2006) package to compute the actor-level metrics. Learning times are not relevant, given that all the experiments took, on average, only 0.02 seconds.

### 4.3 Interpretation of the Coordinate Axes of a Tucker3 Space

In this subsection we interpret the output of the resulting Tucker3 model. Within the scope of our framework, the essential step is to interpret the meaning of the coordinate axes of the maximum-variance 2D space where the trajectories will be projected. The procedure we use for the interpretation of the 2D space is independent of the level of analysis (micro-level or macro-level). The interpretation is ideally done by observing each component as a latent variable and then assigning a label that expresses the meaning of its contents. However, and similarly to two-way analyses (e.g. PCA, SVD), the interpretation of these components as theoretical constructs is not straightforward (Kroonenberg, 2008) and highly prone to subjectivity. To mitigate this problem, it is a common practice to rotate the component matrices using procedures such as the *varimax orthogonal rotation*. The rotational freedom of the Tucker3 model allows the exploration of different directions for interpretation, without compromising the model fit. This property makes it possible to choose the solution that is easiest to interpret by optimizing the simplicity of the component scores (ie large values for the entities that contribute more to a given component, and values close to zero for those entities with negligible contribution). For the present data, we rotated the initial solution in such a way that both the core and component matrix **B** jointly tend to a simpler structure. Then, we assigned a meaning to each component based on the analysis of the largest coefficients.

Before delving into the details of the interpretation process, it is important to understand the meaning of the model outputs. The *core tensor*  $\mathcal{G}$  contains the weights of all possible triads (combination of components, for the three modes) and can be understood as a strongly reduced version of the full data array (Kiers and Van Mechelen, 2001). These weights reflect the strength of the links between the components from different modes, thus providing information on the relative importance of the combination of these components, for all modes. For orthogonal models, like the one we use, it can be shown that the size of the squared core entries are proportional to the variation explained by the factor combinations

Table 2: Matricized core array resulting from the application of a Tucker3 model of order  $(2 \times 3 \times 2)$  to the modified version of Van de Bunt’s friendship networks. A1 and A2 refer to the first and second components of matrix **A**. The notation used for B and C follows the same reasoning. The largest entries are emphasized.

	B1xC1	B2xC1	B3xC1	B1xC2	B2xC2	B3xC2
<b>A1</b>	2.81	-1.36	0.75	-7.37	-0.04	-9.21
<b>A2</b>	4.06	-0.05	-5.23	0.15	-25.85	-0.57

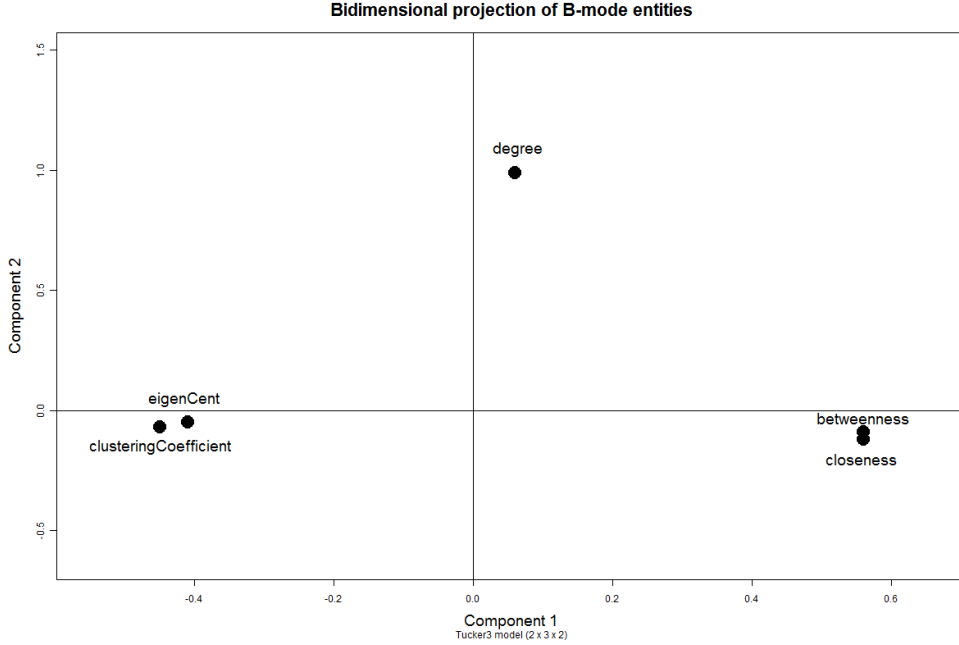


Figure 7: Projection of the coefficients of matrix **B** in the bidimensional space defined by the two most representative components of mode *B*, which is associated to the column-entities (ie actor-level metrics) of the original tensor  $\mathcal{X}$ .

(Kiers and Van Mechelen, 2001). Therefore, in order to find the most significant factor combination, one must find the largest entry in the core. According to Tucker (1966), each core element  $g_{pqr}$  is the score of a *subject type*  $p$  on a *latent variable*  $q$ , in a *prototype condition*  $r$ . The core array obtained for our case study is given in Table 2. The results tell us that the interactions of components that explain the highest portion of the sum of squares and, therefore, are the most important for understanding the data structure, are the interactions  $(2, 2, 2)$  and  $(1, 3, 2)$ . Note that the largest values are associated with both the second and the third components of matrix **B**. This suggests that a coordinate system incorporating these components will be able to recreate a faithful structural image of the original data. Hence, the maximum-variance 2D space where the trajectories will be projected will be defined by the second and the third components, so as to capture the maximum possible information contained in the raw data. In turn, the entries of the *component matrices* **A**, **B** and **C**, represent the weights (also referred to as *scores* or *coefficients*) of the corresponding entities (actors, metrics and time points, respectively) in a given level of a given mode. These *component matrices* have as many columns, or levels, as the number of components defined in the order of the estimated Tucker3 model.

For the purpose of our analysis, the most relevant component matrix is matrix **B** since the plane where we represent the social micro-evolution is defined by its levels. In order to carry out the interpretation, we first display the coefficients of each actor-level metric, given by the entries of **B**, in the space spanned by the two most representative components of the reduced mode *B*, as shown in Figure 7. As mentioned earlier, the most representative components are the second and the third ones. The meaning assigned to each component will be based on the entities that contributed most to its formation. This contribution is reflected in the magnitude of the entities’ coefficients in the component matrices. For this reason our analysis will be focused on the largest values<sup>2</sup> of each column (or component) of the component matrices.

Based on the analysis of the largest coefficients of the second component of matrix **B**, denoted by

<sup>2</sup>We consider *large values* those exceeding 0.4 in absolute sense.

$\mathbf{B}_{:,2}$ , we deduce that the corresponding  $x$ -axis opposes the *clustering coefficient* ( $B_{5,2} = -0.45$ ) and the *eigenvector centrality* ( $B_{2,2} = -0.41$ ) with the *closeness* ( $B_{3,2} = 0.56$ ) and *betweenness* ( $B_{4,2} = 0.56$ ) metrics. While the first pair of SNA metrics is more focused on both the quality (measured by the eigenvector centrality) and intensity (measured by the clustering coefficient) of an actor’s connections, the later set quantifies the agility of a given actor in navigating the network and accessing different sources of information. Therefore, we label this axis as “Sociability VS Accessibility”. The next step is to understand which side of the axis is associated with higher sociability and which side is associated to higher accessibility. To do so, we just need to observe the signs of the corresponding coefficients. Looking again at Figure 7, it is clear that the coefficients of the *closeness* and *betweenness* entities are positive, whereas the coefficients of the *eigenvector centrality* and *clustering coefficient* are negative. This indicates that large positive values of this component reflect high accessibility in the network, whilst large negative values are associated with high sociability in the network.

Regarding the third component of the same matrix, denoted by  $\mathbf{B}_{:,3}$ , in Figure 7 we observe that the corresponding  $y$ -axis is strongly associated with the *degree* metric, which can be used as a proxy for *social activity*. The importance of the degree metric to the formation of the third component is revealed by an extremely large positive coefficient of  $B_{1,3} = 0.99$ . The remaining metrics show values very close to zero and, therefore, have a negligible contribution to this component. Thus, we label the  $y$ -axis as *social activity*. Large positive values in this axis correspond to high social activity (ie high number of immediate connections in the network), whereas large negative values in this axis correspond to low social activity.

In short and based on this interpretation, we deduce that the most desirable social position in the 2D component space is both the rightmost part of the first quadrant and the leftmost part of the second quadrant. The former is associated with high accessibility and high social activity in the network, whereas the later is related to high sociability, in terms of not only quantity of connections (measured by the degree) but also quality and intensity of these connections (measured by the eigenvector centrality and by the clustering coefficient, respectively). Therefore, students whose trajectories take the direction of either the first or the second quadrants are improving their social status. Otherwise, if moving in the direction of the third or fourth quadrant, they are losing their social power. Regarding the amplitude, if most of the time points of a given student’s trajectory have the same coordinates, then one can assume that his/her social position is stable. An analogous reasoning holds for the opposite scenario. After decomposing the tensor and assign a meaning to the components, we define the trajectories of each student following the procedure described in Section 3.2.

#### 4.4 Analysis of Actor-Level Trajectories

As already stated by Van de Bunt et al. (1999), in these networks there is a clear transition between the moment where almost no student knows each other to the moment where there is a significant number of ties between groups of them. This fact can be ascertained by the analysis of the social networks depicted in Figure 5, where we observe a sharp increase in the network density from  $t_0$  to  $t_1$ . The rise of network density reveals an intense period of socialization and tie formation during the first weeks of college. In the following time points, the density of the networks stabilizes, since the friendship groups start to settle down. This phenomenon is also corroborated by the students’ trajectories in the low-dimensional space, illustrated in Figure 8. By projecting all these trajectories, we observe that the origin of most of them (indicated by a black circle) lies in the origin of this space. Such coordinates are indicative of the initial lack of connections among students. Nevertheless, there are a few exceptions related to old acquaintances of some students that attended the same former school, namely students 1, 2, 6, 9, 16, 17, 24 and 31. These students’ trajectories appear cluttered in Figure 5, thus being difficult to trace their origin in this graph. To obtain information regarding their trajectories’ origin one can either perform a visual analysis of the individual trajectories of students or identify the connected nodes on a conventional graph representation of the network for  $t_0$ .

Due to space constraints, in this paper we focus only on the analysis of two trajectories, namely, the trajectories of freshmen 15 and 31. The adopted procedure to select these students was based on the computation of the distance between trajectories and further selection of the pair of trajectories with highest distance. This way we are able to focus on the analysis of students showing the most distinctive social behavior over time. For computing the distance between pairs of trajectories we resort to *Dynamic Time Warping* (henceforth DTW) (Sakoe and Chiba, 1978). DTW is a feature matching technique that computes the distance between two sequences, not necessarily of the same size, through a dynamic programming procedure. The commonly used Euclidean distance can be seen as a special case of DTW, since DTW is upper-bounded by the Euclidean distance. However, according to Keogh

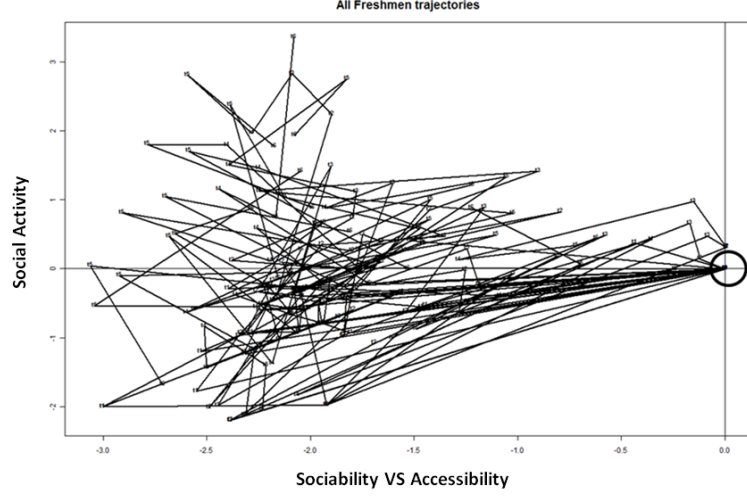


Figure 8: Trajectories of all students in the space spanned by the two most representative components of mode **B**.

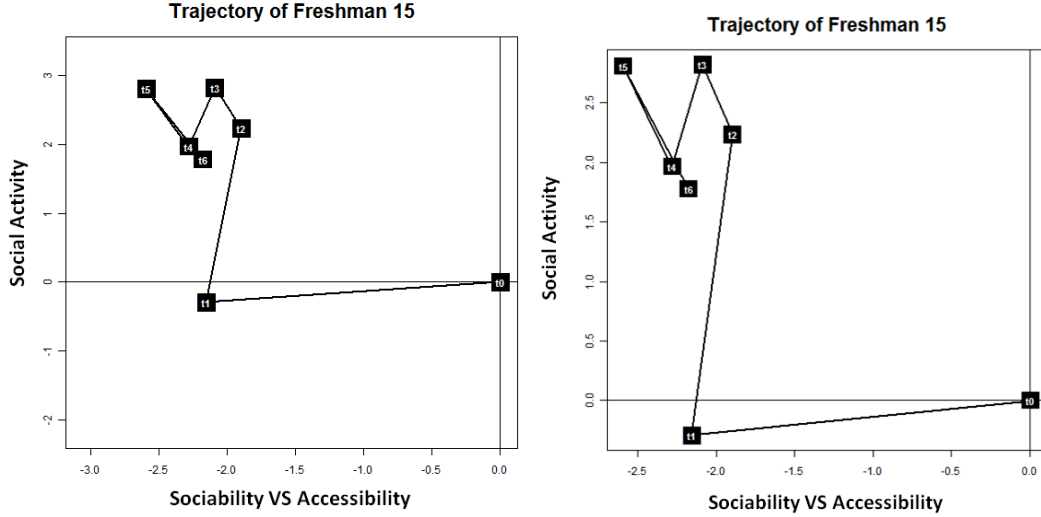


Figure 9: Trajectories of student 15 in the plane defined by the two components of matrix **B**. The right-hand side provides a closer look of the trajectory.

and Ratanamahatana (2005), DTW is a more robust and flexible distance measure than the Euclidean distance. The reasons are the ability of DTW to deal with sequences of different sizes and to compare sequences that, although similar, are misaligned with respect to one of the axes. For the specific case of our methodology, we consider DTW an appropriate measure for assessing the similarity of trajectories because the resulting trajectories might not be equal-sized and aligned with each other, especially within the context of communities.

#### 4.4.1 Trajectory of Student 15

The first trajectory to be analysed is the one pertaining to student 15, which is a non-smoker male attending the 4-year program. His trajectory is depicted in Figure 9. The trajectory is somewhat irregular, showing changes in direction from time to time, but if we consider an overall view we verify that the trajectory trend is to move upwards and slightly leftwards, with most of the time points being positioned in the second quadrant of the plane. With concern to amplitude, we observe that the relative distance between the two farthest time points (ie  $t_0$  and  $t_5$ ) is large compared to, for instance, the amplitude of freshman 31's trajectory (Figure 10), which is more compact. The visual comparison of the amplitudes should be based on the left subfigures, since they map the trajectories in a space with the same

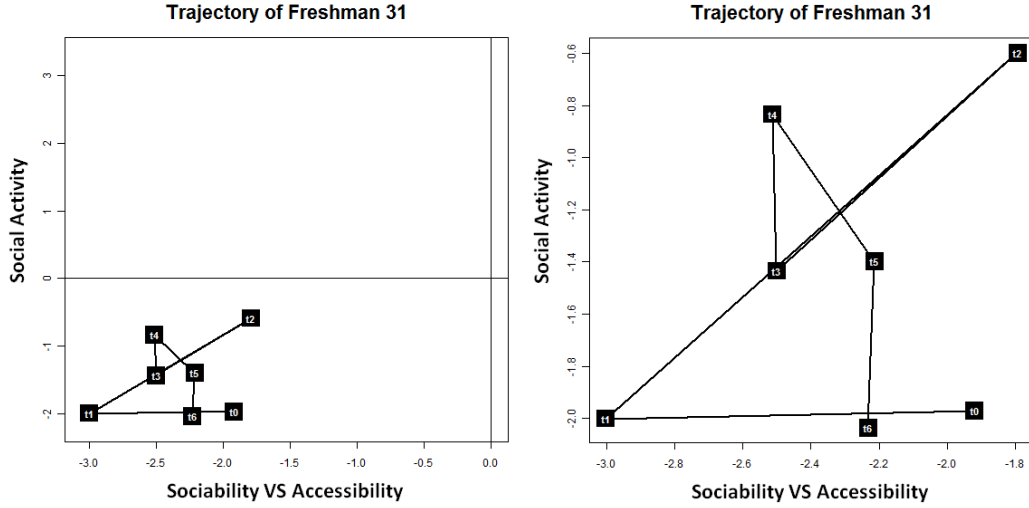


Figure 10: Trajectories of student 31 in the plane defined by the two components of matrix **B**. The right-hand side provides a closer look of the trajectory.

scale. However, if we subdivide the time horizon into two time spans,  $[t_0, t_1]$  and  $[t_2, t_3, t_4, t_5, t_6]$ , it is clear that after an early period of intense tie formation, revealed by a sharp rise in sociability ( $x$ -axis) from  $t_0$  to  $t_1$  and by a significant increase in social activity ( $y$ -axis) from  $t_1$  to  $t_2$ , his number of connections tend to stabilize. Nevertheless, even after this apparent transition and subsequent relative stabilization of his social activity, we can still observe some ups and downs in this activity, which are yet followed by a growth in sociability until  $t_5$ . Since social activity (measured by degree) is independent of other actors' number of connections, whereas sociability (as we defined it) is influenced by changes in the global network structure, a possible explanation for this evolution can be supported on the dichotomy quantity VS quality of social relationships. Thereby, although student 15 might change frequently the number of people with whom he connects with, overall he tends to socialize with students that are themselves well-connected in the network and, thus, can be regarded as "popular". On the other hand, since friendships need to be nurtured, at some moments he might had chosen to invest more of his time in strengthening his bonds with a smaller number of other students. This also reflects in higher sociability. In terms of coordinates, the best social position for student 15 was achieved in  $t_5$  (large scores in both axes). However, in  $t_6$  his position in the network deteriorates, which is indicated by a simultaneous decrease in degree and sociability. This might be related to the loss of some strategic connections in the network (e.g. popular students).

In a nutshell, we deduce that student 15 is a sociable individual who was able to successfully integrate the social circle of university freshmen and create important connections.

#### 4.4.2 Trajectory of Student 31

The second trajectory to be analysed is the one displaying the highest DTW distance from the trajectory of student 15. This trajectory is associated with a smoker male also attending the 4-year program, who we designate as student 31. His trajectory has low amplitude and is quite stable in terms of overall position in the plane, since all time points are located in the third quadrant. This can be ascertained in Figure 9. However, when focusing on the dynamics of this student, it is apparent some inconstancy and lack of trend in the trajectory's direction. The frequent oscillations can be indicative of social instability with respect to both the quantity ( $y$ -axis), quality and intensity ( $x$ -axis) of his relations. Another interesting fact we can draw from the analysis of this trajectory is that student 31 is one of the few freshmen who had a few acquaintances from the former school, which is indicated by the coordinates of the first time point  $t_0$ . While most students start their trajectories at the origin of the plane, student 31 begins with some social advantage, which is reflected in a higher sociability for  $t_0$  (ie higher  $x$ ). In the next time point he is able to increase this sociability, although this increase is not accompanied by an increase in social activity. This indicates that although his number of connections stay roughly the same, they become more valuable as a result of the socialization of his friends with other network's members. Nevertheless, we observe a turning point in  $t_2$ , revealed by the displacement of this student's position in the northeast direction



of the space. This movement can be interpreted as a broadening of the circle of friendships of student 31 during the time span  $[t_1, t_2]$ , indicated by a rise in social activity. By extending his social circle, the student also decreased his sociability, eventually because the new connections are not as well-connected as the few connections he had before, which reflects in the decrease of eigenvector centrality. Besides, this strengthening in the socialization process might also contributed to the decrease of his ego-network density, thus reducing the clustering coefficient, which was also one of the metrics associated with the  $x$ -axis. In the subsequent time points, the trajectory oscillates between periods of higher and lower degree and, likewise, higher and lower sociability, ultimately tending to a less desirable position in the last time point  $t_6$ .

In short, if we compare the first time point  $t_0$  with the last time point  $t_6$ , we observe that although there is some variation in the movement of student 31 across the space, during the intermediate time points, this student tends to return to a social position which is very similar to the one he had at the beginning of college. One possible reason behind this is that after a period of socialization outside his circle of connections, he decided to focus more on his closest friends and intensify his relationships with them, so as to create a tightly-knit group of friends.

## 5 Related Work

Traditional approaches to represent high-dimensional data include projections of data entities in low-dimensions spaces, that preserve most of the original data variation. Examples of such representations are the ones yielded by PCA, where the entities are usually plotted in a bidimensional space spanned by the first and the second principal components. On the other hand, when dealing with tensorial representations of data, the analysis of their decompositions are commonly performed through the analysis of graphical representations of the component matrices of each mode, where the  $x$ -axis represent the entities of the corresponding mode and the  $y$ -axis represent the entries of the component matrix. However, both these approaches do not meet the requirements we would like to embed in a visualization scheme for evolving data, which are compactness, substance and simplicity.

A good survey on time-oriented visualization approaches is provided by Aigner et al. (2007). The work of Sun et al. (2009) also discusses a topic similar to the one addressed in this paper. In this work, they propose a hybrid two-step approach that summarizes and extracts patterns from tensorial representations of large content-based social networks. In the first step, they perform a high-order dimensionality reduction using Tucker decomposition and, in the second step, the dimensions of each one of the decomposed modes are clustered (Sun et al., 2009). The extracted patterns are presented to the user by means of a hierarchical graph visualization, which is build upon the results of the cluster analysis. Nevertheless, they do not define temporal trajectories of entities, nor focus on the study of their evolution.

To the best of our knowledge the most related work to ours is the STATIS method, proposed by Lavit et al. (1994). Similarly to the Tucker3 model, STATIS is a three-way method for exploratory data analysis. However, it differs from the Tucker3 model in the sense that it cannot be generalized to  $N$ -way data and performs separately the exploration of each tensor mode (instead of simultaneously, as in the Tucker3 model). One of the steps of this method consists in projecting the trajectories of individuals over time in the so-called "compromise space". The main difference between ours and STATIS trajectories lies in the space where we define them which, in our case, is based on the decomposition yielded by the Tucker3 model. Besides, in STATIS the interpretation of the trajectories is made in relation to the average trajectory of a fictitious individual, while in our approach the interpretation is based on the properties of the trajectories themselves (e.g. direction, shape, amplitude, etc.).

## 6 Summary and Conclusions

In this paper we discussed a new approach to visualize dynamic social networks, which is based on the extraction of temporal trajectories from tensorial representations of time-evolving social networks. The projection of trajectories of single network entities in a concise 2D space helps understand the evolution of these entities with respect to latent properties of the underlying dynamic social network. The use of Tucker3 model to generate the 2D space guarantees that this space describes the most important information contained in the three-way raw data. A case study using a set of self-reported friendship networks among university freshmen revealed that the proposed approach has several desirable features,



being concomitantly simple, informative and compact, thus allowing the understanding of structural changes in the evolution of entities in dynamic social networks.

The proposed methodology has potential uses in several real-world business applications, such as *churn prediction* and *viral marketing*. In the former application, the goal is to predict the probability of *customer churn*<sup>3</sup> in telecommunications companies through the analysis of networks of phone calls between customers over time. In this context, our methodology could be used to first map the trajectory of each customer appearing in the "who-calls-whom" network, and then comparing the distance between the trajectories of current customers with the *churning trajectories* (e.g. trajectories of customers who abandoned the company). Churn prediction is a topic with inherent commercial interest and practical value for companies. The timely detection of a potential churn can help companies improve the efficiency of their marketing plans and develop targeted actions to avoid customer loss. Besides, it can contribute to increase the profits, since it is generally believed that it is more expensive to attract new customers than to retain existing ones. In the later application, the main goal is to analyse the social network of customers, find those with strongest influence in the market and then use them as "seeds" for promoting a product. By marketing primarily customers with high network value, companies can take advantage of word-of-mouth advertising, by leveraging customers themselves to carry out the most promotional effort. Marketing a small set of customers, who are expected to positively influence the probability of their immediate connections in buying the product, can prove to be more cost effective than traditional marketing methods. However, the customer selection process in viral marketing is usually based on a single snapshot of their social networks. Using our methodology, we could add temporal considerations to this process, by deriving the network value of a customer based on their temporal trajectory. This way we would expect to improve the success rate of word-of-mouth.

We acknowledge that our study is limited in ways that suggest opportunities for future research. A natural direction would be to test the performance and feasibility of our methodology in large social networks. We also intend to conduct more comprehensive studies to assess the effectiveness of our methodology in other domains, such as the previously mentioned business applications.

**Acknowledgments:** This research was supported by the project KDUS (PTDC/EIA-EIA/098355/2008), by ERDF through the COMPETE Programme and by National Funds through FCT within the project FCOMP-01-0124-FEDER-022701. First author was also funded by FCT, under the PhD grant SFRH/BD/81339/2011.

## References

- Acar, E. and Yener, B. (2009). Unsupervised multiway data analysis: A literature survey. *IEEE Transactions On Knowledge and Data Engineering*, 21:6–20.
- Ahn, J.-w., Plaisant, C., and Shneiderman, B. (2011). A task taxonomy for network evolution analysis. Human-Computer Interaction Lab Tech Report HCIL-2011-09, University of Maryland.
- Aigner, W., Miksch, S., Muller, W., Schumann, H., and Tominski, C. (2007). Visualizing time-oriented data - a systematic view. *Computers and Graphics*, 31(3):401–409.
- Blondel, V., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.
- Bro, R. and Smilde, A. K. (2003). Centering and scaling in component analysis. *Journal of Chemometrics*, 17(1):16–33.
- Carroll, J. D. and Chang, J. J. (1970). Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition. *Psychometrika*, 35:283–319.
- Csardi, G. and Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems*:1695.
- De Lathauwer, L. De Moor, B. V. J. (2000). A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278.
- Fortunato, S. (2010). Community detection in graphs. *Physics Report*, 486(3-5):75–174.

---

<sup>3</sup>Customer churn refers to the propensity of customers to abandon a company in a given time period.

- Ghoniem, M., Fekete, J. D., and Castagliola, P. (2005). On the readability of graphs using node-link and matrix-based representations: A controlled experiment and statistical analysis. *Information Visualization*, 4:114–135.
- Giordani, P., Kiers, H., and Del Ferraro, M. (2012). Three-way component analysis using the r package threeway.
- Greene, D., Doyle, D., and Cunningham, P. (2010). Tracking the evolution of communities in dynamic social networks. In *Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining, ASONAM’2010*, pages 176–183. IEEE Computer Society.
- Harshman, R. A. (1970). Foundations of the parafac procedure: Models and conditions for an explanatory multi-modal factor analysis. Working Paper in Phonetics, UCLA - University of California, Los Angeles.
- Keogh, E. and Ratanamahatana, C. A. (2005). Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7:358–386.
- Kiers, H. A. L. (2000). Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics*, 14(3):105–122.
- Kiers, H. A. L. and Van Mechelen, I. (2001). Three-way component analysis: Principles and illustrative application. *Psychological Methods*, 6(1):84–110.
- Kolda, T. G. and Bade, B. W. (2009). Tensor decompositions and applications. *SIAM Review*, 51(3):455–500.
- Kroonenberg, P. M. (1983). *Three-mode Principal Component Analysis: Theory and Applications*. DSWO Press.
- Kroonenberg, P. M. (2008). *Applied Multiway Data Analysis*. John Wiley and Sons.
- Lavit, C., Escoufier, Y., Sabatier, R., and Traissac, P. (1994). The act (statis method). *Computational Statistics and Data Analysis*, 18(1):97–119.
- Levin, J. (1963). *Three-mode factor analysis*. PhD thesis, University of Illinois, Urbana.
- Moody, J., McFarland, D., and Bender-deMoll, S. (2005). Dynamic network visualization. *American Journal of Sociology*, 110(4):1206–1241.
- Morup, M. and Hansen, L. K. (2009). Automatic relevance determination for multi-way models. *Journal of Chemometrics*, 23(7-8):352–363.
- Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, 45(23):167–228.
- Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113.
- Oliveira, M. and Gama, J. a. (2012). A framework to monitor clusters’ evolution applied to economy and finance problems. *Intelligent Data Analysis*, 16:93–111.
- Oseledets, I., Savostyanov, D., and Tyrtyshnikov, E. (2009). Linear algebra for tensor problems. *Computing*, 85(3):169–188.
- Palla, G., Barabasi, A.-L., and Vicsek, T. (2007). Quantifying social group evolution. *Nature*, 446:664–667.
- Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions On Acoustics, Speech and Signal Processing*, 26:43–49.
- Skillicorn, D. (2007). *Understanding Complex Datasets: Data Mining with Matrix Decompositions*. Chapman and Hall/CRC.
- Smilde, A. K. (1992). Three-way analyses problems and prospects. *Chemometrics and Intelligent Laboratory Systems*, 15:143–157.

- Sun, J., Papadimitriou, S., Lin, C., Cao, N., Liu, S., and Qian, W. (2009). Multivis: Content-based social network exploration through multi-way visual analysis. In *SDM'09: Proceedings of the 2009 SIAM International Conference on Data Mining*, pages 1063–1074. SIAM.
- Sun, J., Tao, D., Papadimitriou, S., Yu, P. S., and Faloutsos, C. (2008). Incremental tensor analysis: Theory and applications. *ACM Transactions on Knowledge Discovery from Data*, 2(3):11:1–11:37.
- Takaffoli, M., Sangi, F., Fagnan, J., and Zaïane, O. R. (2010). A framework for analyzing dynamic social networks. In *Proceedings of the 7th Conference on Applications of Social Network Analysis*, pages 1–14.
- Team, R. D. C. (2008). R: A language and environment for statistical computing. ISBN: 3-900051-07-0.
- Timmerman, M. E. and Kiers, H. A. L. (2000). Three mode principal component analysis: Indicating the numbers of components and sensitivity to local optima. *British Journal of Mathematical and Statistical Psychology*, 53(1):1–16.
- Tucker, L. (1963). Implications of factor analysis of three-way matrices for measurement of change. In Harris, C. W., editor, *Problems in Measuring Change*, pages 122–137. John Wiley.
- Tucker, L. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311.
- Van de Bunt, G., Van Duijn, M., and Snijders, T. (1999). Friendship networks through time: An actor-oriented statistical network model. *Computational and Mathematical Organization Theory*, 5(2):167–192.
- Yi, J. S., Elmqvist, N., and Lee, S. (2010). Timematrix: Analyzing temporal social networks using interactive matrix-based visualizations. *International Journal of Human-Computer Interaction*, 26:1031–1051.

## The authors

### Márcia Oliveira

Márcia Oliveira is a researcher at LIAAD/INESC TEC in Portugal. She received her bachelor’s degree in Business Administration from the University of Aveiro in 2008 and her master’s degree in Data Analysis and Decision Support Systems from FEP, School of Economics and Business, University of Porto in 2010. She is currently a PhD candidate at FEP, University of Porto. Her research focuses on modeling and understanding the evolution of social networks, with a special interest in the corresponding business applications.

### João Gama

João Gama is a researcher at LIAAD/INESC TEC, University of Porto, working at the Machine Learning group. His main research interest is in Learning from Data Streams and evolving data. He published more than 100 articles. He served as Co-chair of ECML 2005, DS09, ADMA09 and a series of Workshops on KDDs and Knowledge Discovery from Sensor Data with ACM SIGKDD. He is author of a recent book on Knowledge Discovery from Data Streams.