Social Networks 34 (2012) 291–308

Contents lists available at ScienceDirect

Social Networks

journal homepage: www.elsevier.com/locate/socnet

Visualization methods for longitudinal social networks and stochastic actor-oriented modeling☆

Ulrik Brandes*, Natalie Indlekofer, Martin Mader

Department of Computer & Information Science, University of Konstanz, Germany

A R T I C L E I N F O

Keywords:
Longitudinal social networks
Dynamic network visualization
Stochastic actor-oriented models
Model fit

A B S T R A C T

As a consequence of the rising interest in longitudinal social networks and their analysis, there is also an increasing demand for tools to visualize them. We argue that similar adaptations of state-of-the-art graph-drawing methods can be used to visualize both, longitudinal networks and predictions of stochastic actor-oriented models (SAOMs), the most prominent approach for analyzing such networks. The proposed methods are illustrated on a longitudinal network of acquaintanceship among university freshmen.

1. Introduction

With the ever-increasing availability of time-varying data and the diffusion of advanced modeling methods, research on longitudinal network analysis is widening tremendously. This is especially true for the interest in dependencies among tie dynamics and actor attributes, and more concretely the co-evolution of networks and behavior. Since the temporal dimension constitutes an additional, qualitatively different level of complexity, the demands on visualization tools are even higher than they are anyway in static network analysis (Bender-deMoll and McFarland, 2006).

Social network visualization is a field of growing interest in itself (Klovdahl, 1981; Freeman, 2000; Brandes et al., 2006), and partly so because very different approaches are suitable for specific use cases. For the present case, we assume to have longitudinal network data given in the form of panel data, i.e., as a time-ordered sequence of interrelated network observations that possibly differ in actor composition, structure, and attributes. In social sciences, this is the most common form of longitudinal network data today, and often due to data collection in waves or aggregation of dyadic events over time intervals. The latter is frequently done to allow for the application of the same methods that are common for static networks, and various forms of aggregation are described in Bender-deMoll and McFarland (2006).

We here define our problem area as that of visualizing a given sequence of snapshots of an evolving social network (rather than, say, an unordered collection of networks, an event stream, or a process taking place on a network). The task is further limited to producing a corresponding sequence of diagrams which may or may not serve as the basis of an animation (rather than, say, a merged view of the entire evolution). The characterizing trade-off in this situation is between the individual quality of each snapshot and the persistence of features over the sequence (Brandes and Wagner, 1997). In other words, each diagram should be a good representation of the corresponding cross-sectional network, and at the same time, a mental map of the structure should be preserved as much as possible to relate the individual frames with less cognitive effort (Misue et al., 1995).

The motivation behind this task is to facilitate visual exploration of longitudinal network data in a generic way. By using a specific methodology, however, analysts take a specific perspective that is generally in need of targeted visualization designs. As a concrete example, we here focus on the most prominent approach to longitudinal social network analysis, stochastic actor-oriented modeling (Snijders, 2005; Snijders et al., 2010b), and show that with little adaptation, the same visualization techniques can be applied to reveal such a model's predictions and interrelate them with the actual observation. Our approach is likely to generalize to other models as well.

The remainder of this article is organized into three main parts. Since the crucial technical challenge in network visualization

☆ Research partially supported by DFG under grants Br 2158/3-2 (as part of 05-ECRP-026) and GRK 1042.

* Corresponding author. Tel.: +49 7531 88 4433; fax: +49 7531 88 3577.
E-mail address: Ulrik.Brandes@uni-konstanz.de (U. Brandes).

0378-8733/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.socnet.2011.06.002

is to find a suitable layout for the underlying graph structure, we start by providing background on layout algorithms for static graphs in Section 2, and outline a method known as *stress minimization* that is central to our approaches. In Section 3, we review the dynamic graph drawing problem, and propose specific instantiations of stress minimization designed for visual exploration of dynamic graphs. In the third part, we introduce two targeted visualization approaches for stochastic actor-oriented models (SAOMs) in Section 5, after recalling the formal basics of SAOMs in Section 4. The first of these approaches shall help assess congruence of simulations and observations w.r.t. their underlying graph structure, whereas the second one is to point to inhomogeneities across actors, if any. We conclude with a brief discussion that includes directions for future work.

1.1. Running example

We use a longitudinal network of acquaintanceship among university students as a running example. The data is courtesy of Britta Renner and Manja Vollmann (Department of Psychology, University of Konstanz) and was collected in 15 waves between October 2008 and February 2009.

Students provided, among many other data, their current perceived level of acquaintanceship with each other on a scale from 1 (lowest) to 7 (highest). We dichotomized each observation using 5 as a threshold. Of the 78 freshmen majoring in Psychology, only nine did not participate in an initial screening, never answered any questionnaire, or never made a nomination resp. were never nominated at a level above the threshold.

The example networks thus consist of acquaintanceship nominations among 69 students (18 male, 51 female) that form a connected component when aggregated over all waves.

The data constitutes a realistic scenario in which our methods may be applied, but is used here solely for illustrative purposes. No attempt at justifying models or drawing conclusions will be made.

2. Graph drawing methods for static general graphs

Social network visualization can draw on two major streams of research, information visualization of networks (Herman et al., 2000) and graph drawing (Di Battista et al., 1999; Kaufmann and Wagner, 2001). Roughly speaking, the focus in information visualization is on visualization design, navigation, and interactivity, whereas properties and construction of geometric representations are more central to graph drawing.

We here restrict our scope to the most common graphical representation for social networks, node-link diagrams (referred to as *sociograms* in Moreno, 1953), in which actor-representing vertices are depicted as points (or, more precisely, graphical elements described by a single position), and tie-representing edges are depicted as lines linking their endpoints. We will not, in general, make the distinction between actors, nodes, vertices, and points, and between ties, links, edges, and lines.

The central task in creating node-link diagrams is to determine positions for its elements, referred to as the diagram's *layout* in the following. This is because positional differences are the most accurately perceived graphical attributes (Cleveland and McGill, 1984), and layout with complex dependencies is the most challenging problem algorithmically. If the layout is of low quality, even the best graphical design (in terms of using other graphical attributes such as shape, color, size, etc.) or interaction mechanisms can only attenuate the problems of poor legibility and interpretation artifacts.

While graph structure is represented completely in plain node-link diagrams, the other attributes of a network can be incorporated by varying graphical attributes as mentioned above. Clearly, these choices are more dependent on the data and context, and in general easier to implement.

2.1. Graph layout

In addition to distinct vertex positions to avoid ambiguity, the following objectives are commonly considered relevant for application-independent layout (Bertin, 1983; Purchase et al., 1997).

- Edges should be of more or less the same length.
- Vertices should be distributed well over the drawing area.
- The number of meaningless edge crossings should be kept small.
- Symmetries in graph structure should be visible in geometric symmetries.

For specific applications and purposes, there may be many more criteria to observe. For most of them, optimization is computationally intractable even in isolation, at least for general graphs. Since, in addition, the various criteria are frequently contradictory, general-purpose graph-drawing algorithms are usually heuristic in nature.

Even though social networks exhibit some general tendencies such as sparseness and local clustering, they do not constitute a formally boundable class of graphs that allows for specific optimization algorithms. Due to their general applicability, conceptual simplicity, wide availability, and ability to produce satisfactory results in general, the most popular class of methods used for social network layout are *force-directed* or *energy-based* methods (Brandes, 2001), colloquially known as *spring embedders* (Eades, 1984).

The most widely available, and often only, layout algorithm in common software tools for social network analysis is the spring embedder variant of Fruchterman and Reingold (1991). It is a force-directed method in which a graph is likened to a physical system of repelling objects (the vertices) and springs of a given length (the edges) binding adjacent vertices together. Vertices are iteratively repositioned based on the forces exerted on them, so that the system moves toward a force equilibrium. The approach is easy to implement and yields acceptable results for small graphs, and it can be tuned for specific purposes by introducing additional or alternative forces.

There is, however, clear experimental evidence (Brandes and Pich, 2009) that this and related force-directed methods do not scale well to larger graphs, both in terms of quality and efficiency. It is almost ironic that a current variant of the earliest computer-implemented method for drawing social networks (Kruskal and Seery, 1980, already applied in the late 60s), turns out to be far superior.

This favorable approach, known as *stress minimization*, is an instance of a family of dimension-reduction methods referred to as *multidimensional scaling* (see, e.g., Cox and Cox, 2001). It is based on an objective function called *stress* (Kruskal and Wish, 1978) and was re-popularized in graph drawing by Gansner et al. (2004). Details are given next, but it should be noted that the same objective function was also used in the spring embedder of Kamada and Kawai (1988), although with an inferior minimization method.