Erschienen in: Social Networks ; 34 (2012), 3. - S. 291-308
https://dx.doi.org/10.1016/j.socnet.2011.06.002

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

ARTICLE INFO

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

ABSTRACT

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).

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.

2.2. Stress minimization

We next describe the workhorse of our approach, stress minimization, in detail. Let $G = (V, E)$ be an undirected graph defined by a set V of n vertices, and a set E of m edges. Given a matrix D of vertex *dissimilarities* δ_{ij}, $i, j \in V$, the purpose of stress minimization is to determine positions $p_i = \langle x_i, y_i \rangle \in \mathbb{R}^2$ for every vertex $i \in V$ such that the Euclidean distances in the plane resemble the given dissimilarities as closely as possible, i.e.,

$$\delta_{ij} \approx \|p_i - p_j\|,$$

where $||\cdot||$ denotes the Euclidean norm. For any given layout $P = (p_1, \ldots, p_n)$ this is quantified using a parameterized *stress function* $\mathrm{stress}(P)$,

$$\mathrm{stress}(P) = \sum_{i<j} \omega_{ij} (\delta_{ij} - \|p_i - p_j\|)^2, \tag{1}$$

where $W = (\omega_{ij})_{i,j \in V}$ is a weight matrix whose entries determine the contribution of each pair $i, j \in V$. Since stress is defined as the weighted sum of squared distance-representation errors, the objective is to find a layout of minimum stress.

In graph drawing, graph-theoretic distances (i.e., lengths of shortest paths) are a plausible choice for dissimilarities (Kamada and Kawai, 1988; Gansner et al., 2004). The stress term of each dyad then corresponds to the squared error of representing a shortest path as a straight line with unit length edges. Because these distances are clearly not realizable for any non-trivial graph, weights $\omega_{ij} = \delta_{ij}^{-2}$ discount representation errors for distant pairs, thus emphasizing local accuracy.

Since no closed form is known to compute a layout with minimum stress directly, the approach appears to share some of the drawbacks of other force-directed methods. In particular, iterative stress reduction in general only yields a local minimum which may be far from an optimal layout. The experiments of Brandes and Pich (2009) suggest, however, that low-stress layouts can be obtained routinely and efficiently using a two-step process: In the first step, an initial layout is determined using *classical scaling* (Torgerson, 1952), the initial, spectral-decomposition variant of multidimensional scaling that has an essentially unique solution in which large distances are represented well, and that can be approximated very quickly (Brandes and Pich, 2007). In the second step, the representation of small distances is improved by iteratively and monotonically reducing the stress using an optimization technique called majorization (de Leeuw, 1977) that can be implemented by a simple, localized process moving one vertex at a time and considering only a sparse version of the stress function. See Gansner et al. (2004) for details.

The kind of results obtained is illustrated in Fig. 1, using four observations from our running example.

We will not elaborate on algorithmic issues any further, but would like to emphasize that stress minimization is reasonably easy to implement, and yields much better results than other force-directed methods (Brandes and Pich, 2009), including multilevel approaches (Walshaw, 2001; Gajer et al., 2004; Hachul and Jünger, 2004). At the same time the method is very flexible, because there are several degrees of freedom that allow for sophisticated layout modeling. We may alter the given network's structure resp. stress terms, admissible positions, dissimilarities, and weights to incorporate hard and weak constraints (see, e.g., Dwyer et al., 2008; Gansner and Hu, 2008; Brandes and Pich, 2011).

Indeed, this will be our approach for both, dynamic layout and visual model assessment, as described below.

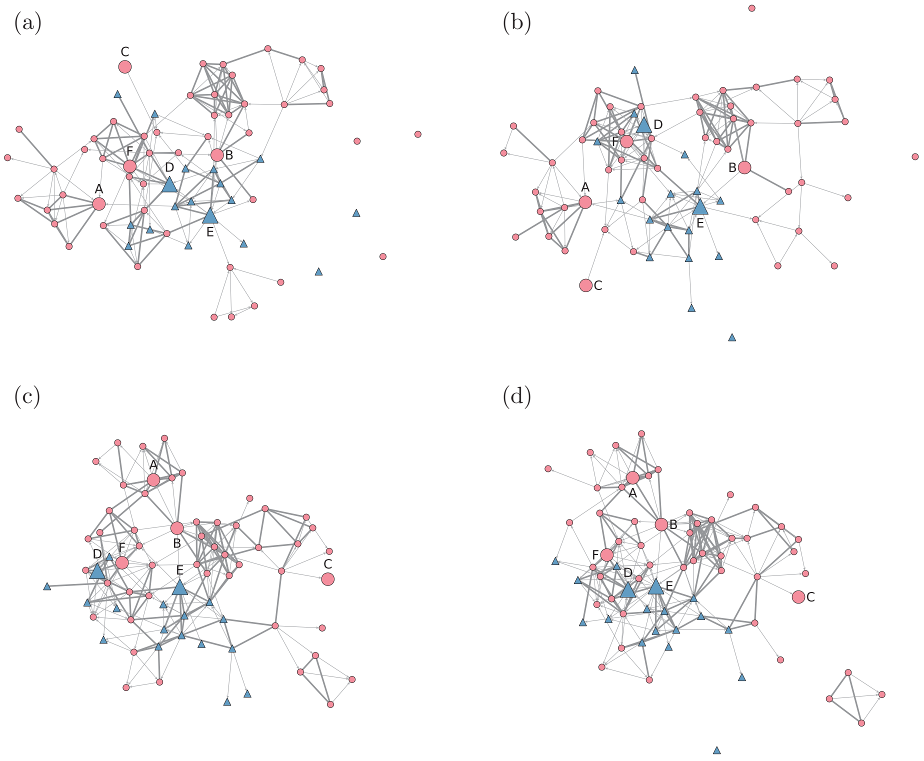


Fig. 1. Four observations of an acquaintanceship network laid out using stress minimization. Red circles represent female students, whereas blue triangles represent males. If an acquaintanceship tie is reciprocated, it is represented by a thicker line segment, otherwise an arrow indicates the direction of nomination. Some vertices are indexed for later reference. (a) Wave 6, (b) Wave 7, (c) Wave 13 and (d) Wave 14. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

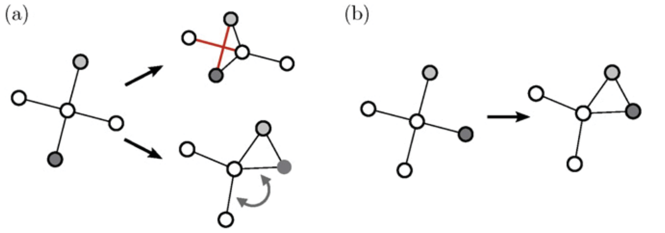


Fig. 2. Knowledge of future changes can inform the choice between otherwise equally good layouts. (a) Unfortunate and (b) fore-sighted.

3. Dynamic network visualization

While producing layouts of single networks can already be challenging, determining coherent layouts for several observations of a longitudinal network is even more difficult (Branke, 2001). This is because, in addition to the criterion of faithfully displaying structural properties, the sequence of layouts should convey the evolution of the network.

The latter is facilitated by maintaining an observer's *mental map* (Misue et al., 1995; Saffrey and Purchase, 2008; Purchase and Samra, 2008). The goal is to ease comparison between a state and its predecessor by largely retaining those parts of a drawing in which little structural change occurred, so that the time a viewer spends on familiarizing with the drawing is not wasted. Since layout algorithms for the static, single-graph case are based solely on the current graph's structure, they have to be modified to strike the right balance between structural changes and positional stability.

Two scenarios need to be distinguished, because they differ in the amount of information available when a layout is determined. Either, the entire sequence of graphs is known in advance, i.e., before any layout is required (generally referred to as *offline scenario*, graphs are given together), or future graphs are unknown at the time that an intermediate state is to be laid out (*online scenario*, graphs are presented to the algorithm one at a time). In an online scenario, it is more difficult to maintain dynamic stability, because future changes are not known at the time that vertices have to be placed.

Let us illustrate this phenomenon by the small example in Fig. 2 consisting of a 4-star in the first observation, and an additional tie between the two gray vertices in second observation. In an online scenario, there is no preference for any of the two layouts of the first observation, whereas in an offline scenario we can actually choose the one that will lead to better quality and less movement.

We will take advantage of the fact that the sequence of graphs that constitute a longitudinal social network is usually known in advance, and therefore assume to be in an offline scenario for the remainder of this paper.

The simplest (and most common) approaches to take stability into account are based on variants of the spring embedders of Fruchterman and Reingold (1991) or Kamada and Kawai (1988), in which the iterative computation for each graph in the sequence is initialized with the preceding layout (Bender-deMoll and McFarland, 2006; Huang et al., 1998; Geipel, 2007; Groh et al., 2009). An implicit assumption is that consecutive graphs are similar in general, so that the initial layout is not too far from a locally optimal one. The method is therefore easy to implement, more efficient than computing a layout from scratch, and applicable in both on- and offline scenarios. Note also that both algorithms repeatedly move one vertex at a time to improve the current layout, thereby creating trajectories from the initial to the final position. These may be useful for animation. In practice, however, linear or sinusoidal coordinate interpolation offer better control and are more commonly used to determine such trajectories. More sophisticated transition methods are described, for instance, in Friedrich and Eades (2002); Friedrich and Houle (2002); Nesbitt and Friedrich (2002).

Inspite of the convenient properties enumerated so far, the approach is rather problematic, because it does not address stability in a controlled way and may hence result in excessive and unnecessary movement of vertices. Moreover, by failing to make use of the existing knowledge about the future in offline scenarios, the approach is biased towards earlier configurations, and therefore prone to suffer from poor local minima precisely as already illustrated in Fig. 2. Hence, layout quality tends to degrade over the course of the sequence.

Among the first to address stability directly were Böhringer and Paulisch (1990), and North (1996) provides a generic problem statement. The trade-off between readability and stability is formalized in Brandes and Wagner (1997) and a similar principle for offline scenarios is proposed in Diehl and Görg (2002). Current proposals generally take one of the following three approaches.

Aggregation. All graphs in the sequence are aggregated into a single graph that has one vertex for each actor. The position of each individual vertex instance in the sequence is determined from a layout of the aggregated graph. This approach is used, e.g., in Brandes and Corman (2003), Dwyer and Gallagher (2004), and Moody et al. (2005).

Linking. All graphs in the sequence are combined into a single graph that has one vertex for each occurrence of an actor, and an edge is created between vertices representing the same actor in consecutive graphs. A layout of this graph directly yields positions for all vertex instances in the sequence. This approach is used, e.g., in Dwyer and Gallagher (2004), Erten et al. (2004), and Dwyer et al. (2006).

Anchoring. Using auxiliary edges, vertices are connected to immobile copies fixed to a desired location which may be, for instance, the previous position in an online scenario, or a reference position in an offline scenario. This approach is used, e.g., in Lyons et al. (1998), Brandes and Wagner (1997), and Frishman and Tal (2008).

Since the method advocated here is a combination of aggregation and anchoring, we describe in more detail how these can be instantiated in a stress-minimization framework.

3.1. *Aggregation*

Maximum stability is obtained when a vertex maintains its position throughout the entire sequence of diagrams. This is called the *flip book* approach in Moody et al. (2005). We argue that suitable positions can be obtained by applying stress minimization to an aggregation of the input sequence.

Given a sequence $G^{(1)}=(V, E^{(1)}), \ldots, G^{(T)}=(V, E^{(T)})$ of T graphs with corresponding dissimilarities $D^{(t)}$, $1 \le t \le T$, we are to determine layouts $P^{(1)}, \ldots, P^{(T)}$ such that a vertex is placed at the same position throughout. In other words, we are looking for one layout P for the vertices in V and let $P^{(t)}=P$ at all times $t=1, \ldots, T$.

We combine all shortest path information into a single stress function using the mean shortest path distance per dyad, and use the same position variables p_i for each instance $i^{(t)}$ of the same actor i:

$$\text{stress}(P) = \sum_{i<j} \omega_{ij}(\bar{\delta}_{ij} - \|p_i - p_j\|)^2, \qquad (2)$$

where $\bar{D} = (\bar{\delta}_{ij})_{i,j \in V}$, $\bar{\delta}_{ij} := \frac{1}{T}\sum_{t=1}^{T} \delta_{ij}^{(t)}$, contains the mean shortest-path distances, and

$$\omega_{ij} = \frac{1}{\bar{\delta}_{ij}^{2}} \cdot \frac{1}{1 + \mathrm{VAR}(\delta_{ij})},$$

where $\mathrm{VAR}(\delta_{ij}) := \frac{1}{T}\sum_{t=1}^{T}(\delta_{ij}^{(t)} - \bar{\delta}_{ij})^{2}$ is the variance of distances within a dyad across all observations. Thus, representation accuracy of dyads that are connected via short paths most of the time is emphasized. By additionally scaling with the variance, priority is given to structures that are relatively stable throughout the sequence.

Since, technically, the aggregate stress function consists of the same type of terms as before, we can also use the same algorithms for its minimization: layout computation is initialized by classical scaling of mean distances in the aggregated graph, i.e., a graph with all vertices and an edge between two vertices if this edge exists in any of the observations. Subsequently, $\mathrm{stress}(P)$ is reduced via majorization.

We require that the aggregated graph is connected, otherwise the data is split into connected components, and layouts are calculated separately for each of them. Note that it may still be the case that some individual networks of a sequence contain multiple disconnected components, resulting in infinite distances and undefined weights. Being in an offline scenario, other observations can be used to fill in gaps as follows. An infinite distance in a dyad is replaced by interpolating between the two finite distances observed previously and next for this dyad, and by adding a small constant, say 1, to emphasize temporary disconnectedness. Let t_{ij}^{-} be the most recent and t_{ij}^{+} be the next observation in which actors i and j have finite distance $\delta_{ij}^{(t_{ij}^{-})}$ and $\delta_{ij}^{(t_{ij}^{+})}$. Then the interpolated distance for dyad $\{i, j\}$ is

$$\delta_{ij}^{(t)} = (1 - \beta_{ij}^{(t)})\delta_{ij}^{(t_{ij}^{-})} + \beta_{ij}^{(t)}\delta_{ij}^{(t_{ij}^{+})} + 1,$$

where $\beta_{ij}^{(t)} = (t - t_{ij}^{-})/(t_{ij}^{+} - t_{ij}^{-})$. In the special cases that there is no last or next finite distance, we do not interpolate, but use the one existing finite distances distance plus the same constant. Although unlikely, it may happen that a dyad has infinite distances at all times, even if the aggregated graph is connected. In this case, a sufficiently large distance $\varDelta$ with a small weight $\varOmega$ is used. We suggest $\varDelta\sqrt{n}$ and $\varOmega = 1/n$ with n the number of vertices. This is in analogy to the height and width of an equally spaced grid with n points.

The resulting aggregate layout for our running example is given in Fig. 3, and Fig. 4 contains samples from the corresponding sequence of diagrams.

The relatively stable major groups, i.e., the female groups around Actor A, and above Actor B, the male group around Actor E, and the mixed group involving Actor D and Actor F, are discernible. Due to perfect stability, occurrences of the same actor are easily indentified in each of the subfigures of Fig. 4. However, a comparison with the layouts obtained by regular stress minimization shown in Fig. 1 reveals that this extreme stability comes at the price of less desirable individual layouts:

- Subconfigurations may appear to be placed oddly at times. An example is the group around Actor A in Waves 13 and 14, where it would be expected to lie closer to the group to which it connects through Actor B. Its position leads to unnecessary long links representing the strong ties to Actor B that have been established by then. These also create much clutter in the group belonging to Actor F. See Fig. 1(d) for comparison.

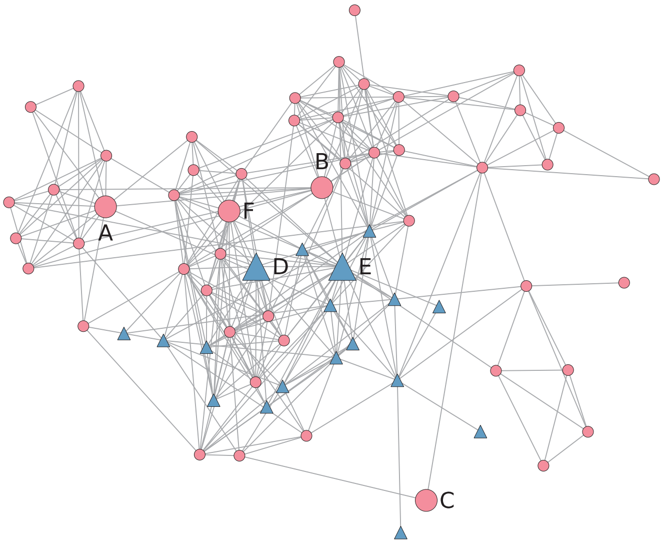


Fig. 3. Global layout obtained by aggregation with mean distances. An edge is drawn if two vertices are adjacent in any graph of the input sequence.

- The almost bipartite structure of the group around Actor A that could be observed in Fig. 1(d) is hardly recognizable.
- Actor C starts out being connected to an actor at the very bottom, but then links up with an actor further up. Since this new neighbor is far away from the first neighbor, any reasonable constant position can only result in either an awkward or a long link connecting Actor C.
- Actor D and Actor E exhibit a very volatile connection, resulting in corresponding changes of geometric distances between them in Fig. 1. Although this volatility can be observed in the aggregate layout, it might be easily missed if the actors were located in denser regions.

These qualitative examples are corroborated by quantitative measurement; for instances, stress of the layout of Wave 6 and Wave 7 as measured by Eq. (1) is almost twice as high as in the corresponding layouts of Fig. 1, and more than thrice as high for Wave 13 and Wave 14. Because of the complex dependencies between structural characteristics, change, and relative positions, a comprehensive quantitative evaluation is beyond the scope of this work, though.

The above examples nevertheless illustrate that fixed positions are helpful in building a mental map of the overall configuration, but that it is desirable to allow for at least small deviations to represent better the specific configurations at individual time points.

3.2. Anchoring

The main idea of the Bayesian approach to online dynamic graph drawing (Brandes and Wagner, 1997) is an explicit modeling of the trade-off between layout quality as measured by an objective function, and layout stability with respect to the previous drawing as measured by a difference metric (Bridgeman and Tamassia, 2000). A conceptually similar, though computationally more demanding approach using instability thresholds for offline scenarios is proposed in Diehl and Görg (2002). Our approach here is a direct translation of the former, and a computationally more efficient variant of the latter.

Concretely, we use the aggregation approach of Section 3.1 to obtain a reference layout that serves as a baseline for representing stable overall structures, and thereby facilitates the formation of a persistent mental map. For each individual graph in the sequence we do, however, allow deviations from reference positions if they

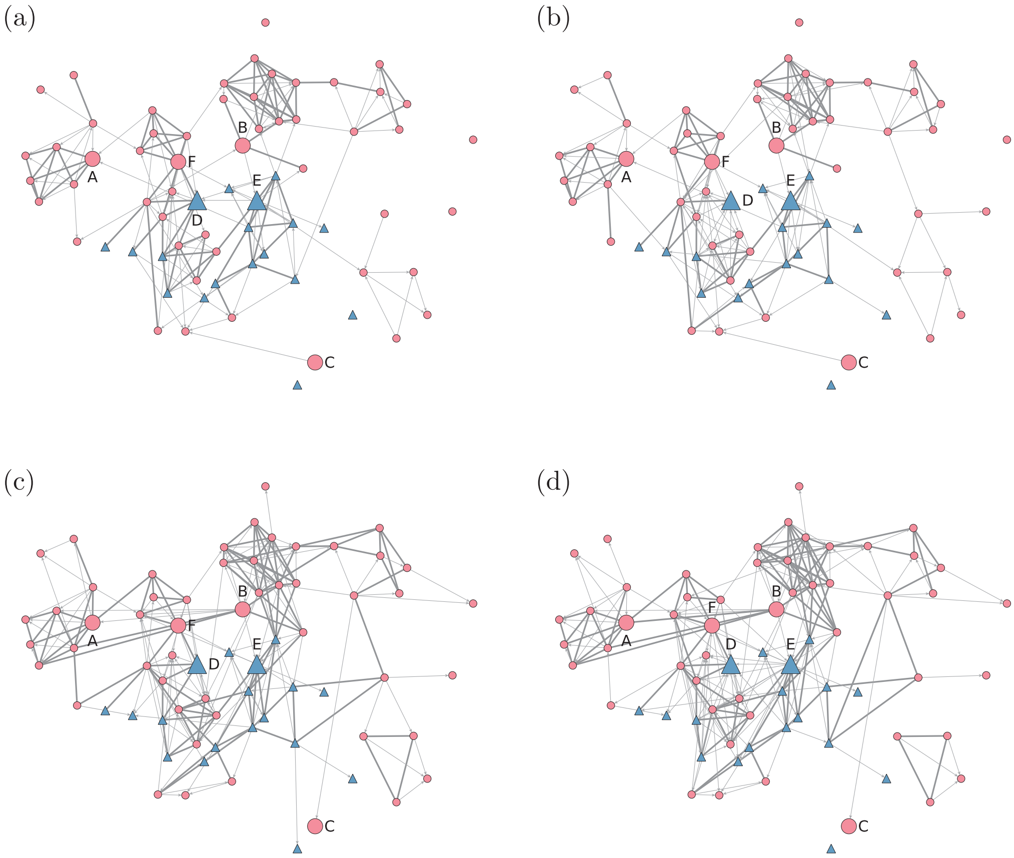


Fig. 4. Four observations of the acquaintanceship network using aggregation layout. While stability is perfect, there are inherent drawbacks as exemplified by labeled vertices (see main text). (a) Wave 6, (b) Wave 7, (c) Wave 13 and (d) Wave 14.

lead to improved representation of momentary structures. A stress function quantifying this compromise is

$$\text{stress}_\alpha(P^{(t)}) = (1-\alpha)\cdot \underbrace{\sum\nolimits_{i<j}\omega_{ij}^{(t)}\left(\delta_{ij}^{(t)} - \left\|p_i^{(t)}-p_j^{(t)}\right\|\right)^2}_{\text{individual layout quality}}$$

$$+\alpha\cdot\underbrace{\sum\nolimits_{i}\phi_i^{(t)}\left\|p_i^{(t)}-p_i\right\|^2}_{\text{stability}},$$

where $P=(p_i)_{i\in V}$ denotes the reference layout determined as described in the previous section, and weights $\phi_i^{(t)}$ allow for inter-vertex variation in deviation tolerance.

The stability term thus corresponds to a pointwise penalty for deviations from the reference layout, and the parameter $0 \le \alpha \le 1$ provides explicit control of the trade-off between quality (original stress) and stability. Note that minimizing stress_α for $\alpha=0$ corresponds to regular stress minimization without control for stability, and $\alpha=1$ corresponds to the aggregation method, since no deviation from the reference layout is tolerated.

Besides modeling stability explicitly as a layout objective, two additional measures are taken. First, we initialize individual layout computation with the reference layout for the first observation, and with the layout of the preceding observation for subsequent ones. Assuming that consecutive observations are structurally similar, stress minimization is expected to reach a similar local minimum as for the preceding observation. Hence, ambiguities will be resolved in favor of the preceding layout if deviations from the reference layout must occur due to structural changes. For the same reason we may also expect that only few iterations are needed to compute a layout. Second, we postprocess the layout thus obtained using *Procrustes rotation* (Sibson, 1978), i.e., by applying an affine transformation that minimizes the sum of squared deviations from reference positions without changing relative distances.

For now, we use constant stability weights $\phi_i^{(t)}{:=}1$ for all i and t. More sophisticated choices, however, may be useful to compensate for cases with highly varying degrees or localized structural change. Another potential use of stability weights is as normalizing factors, such that the quality and stability part in stress_α are on equal scales, and thus, parameter α can be interpreted more easily in between its extreme values. More technical and user-oriented experimentation is needed, though, to quantify dependencies on these parameters.

All 15 networks of our running example are shown in Fig. 5, now laid out using moderate anchoring with reference positions taken from the aggregate layout of Fig. 3. Although more effort is needed to trace vertices from one layout to the next, major substructures remain stable and the individual quality of layouts has improved notably.

This is confirmed when considering the issues we had with constant layout in the previous section. The groups of Actors A and B move towards each other when they start being connected (Waves 11–15). The near-bipartite structure of the group involving Actor A in Wave 14 is apparent. Movement highlights the changing affiliations of Actors C and the volatile connection between Actor D and Actor E. While it is more difficult to locate them in individual layouts, deviations from their reference position are actually meaningful and therefore considered desirable.

Finally, we want to demonstrate the benefits of using offline information. Compare the results of anchoring at reference

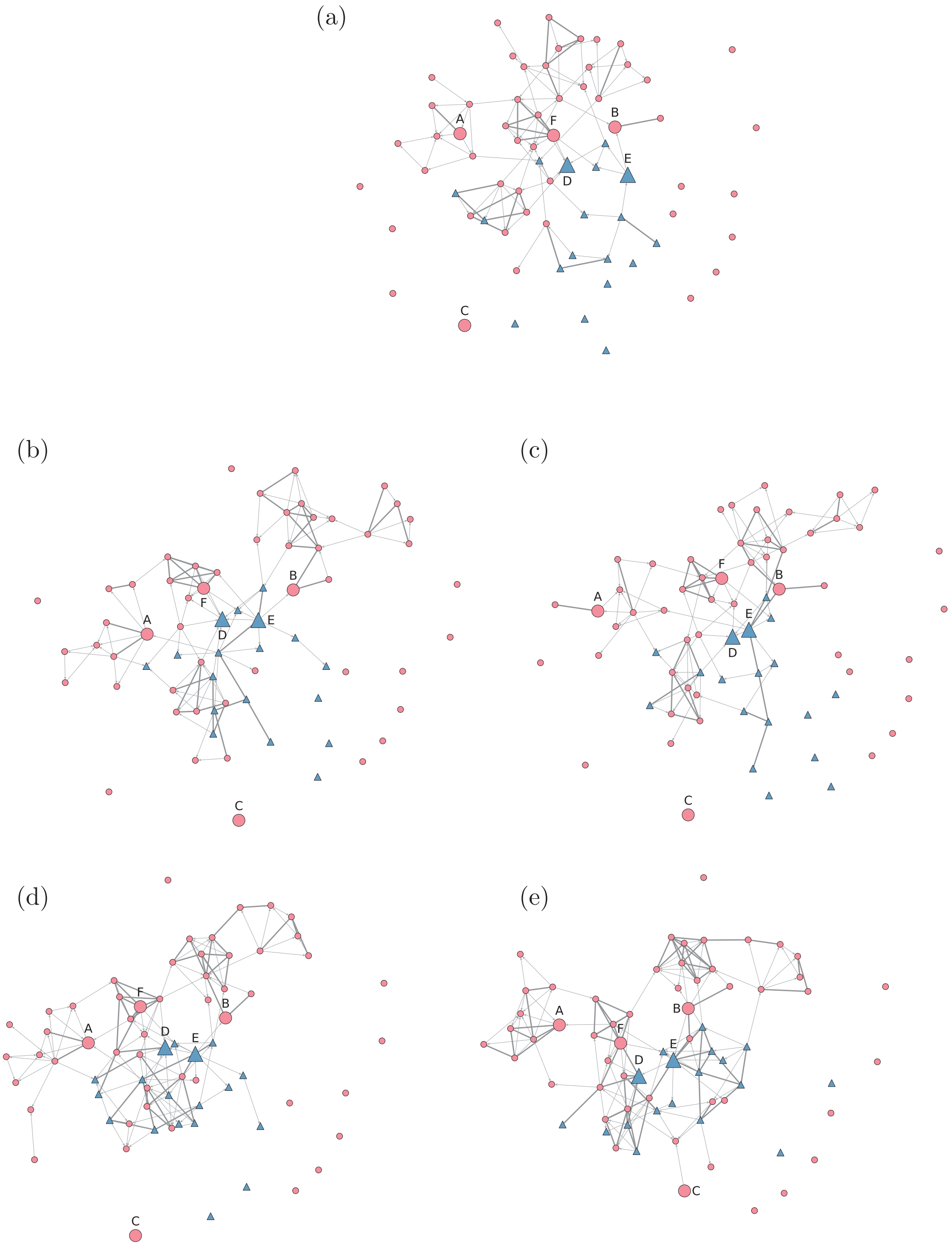


Fig. 5. All waves of the acquaintanceship network drawn by anchoring with aggregate layout as a reference (stability at $\alpha = 0.15$). (a) Wave 1, (b) Wave 3, (c) Wave 2, (d) Wave 4, (e) Wave 5, (f) Wave 7, (g) Wave 6, (h) Wave 8, (i) Wave 9, (j) Wave 10, (k) Wave 11, (l) Wave 13, (m) Wave 12, (n) Wave 14 and (o) Wave 15.

positions from an aggregation layout (Fig. 5) to an online approach based on initialization with the preceding layout and no control for stability (Fig. 6). Note that the latter is similar to the online approach provided by the SoNIA[1] software tool (Bender-deMoll and McFarland, 2006), since the method employed there is conceptually the same, as is the basic objective function, stress, when using the Kamada-Kawai option (MultiCompKK).

Both approaches result in a similar configuration for Wave 10. In Wave 11, the group of Actor A establishes lasting connections to the groups of Actor B and momentary connections to the group of Actor E, forcing the groups to move towards each other. The online approach results in a positioning of the group of Actor A below the group of Actor F in Waves 11 and 12. However, with the dissolution of the momentary connections in Wave 13, those groups are forced to flip around each other, creating excessive movement between observations.

Anchoring with reference positions from an aggregation layout exploits offline information to avoid the need for such flips, and thus eliminates some movement that indicates structural changes inappropriately. Observe that there is much less difference in the configuration of substructures than suggested by the large movements in the online scenario.

[1] http://www.stanford.edu/group/sonia/.

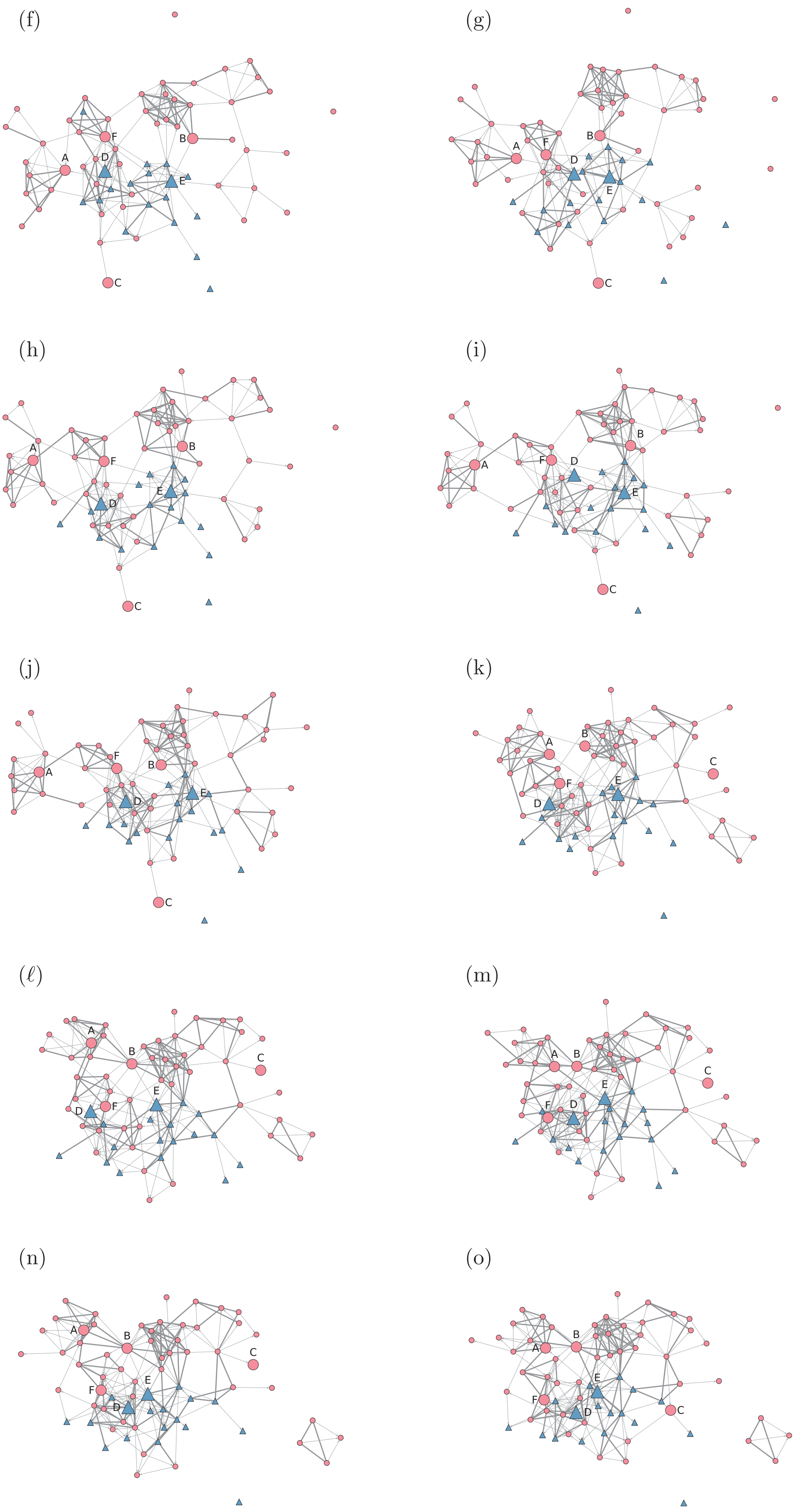


Fig. 5. (*Continued*).

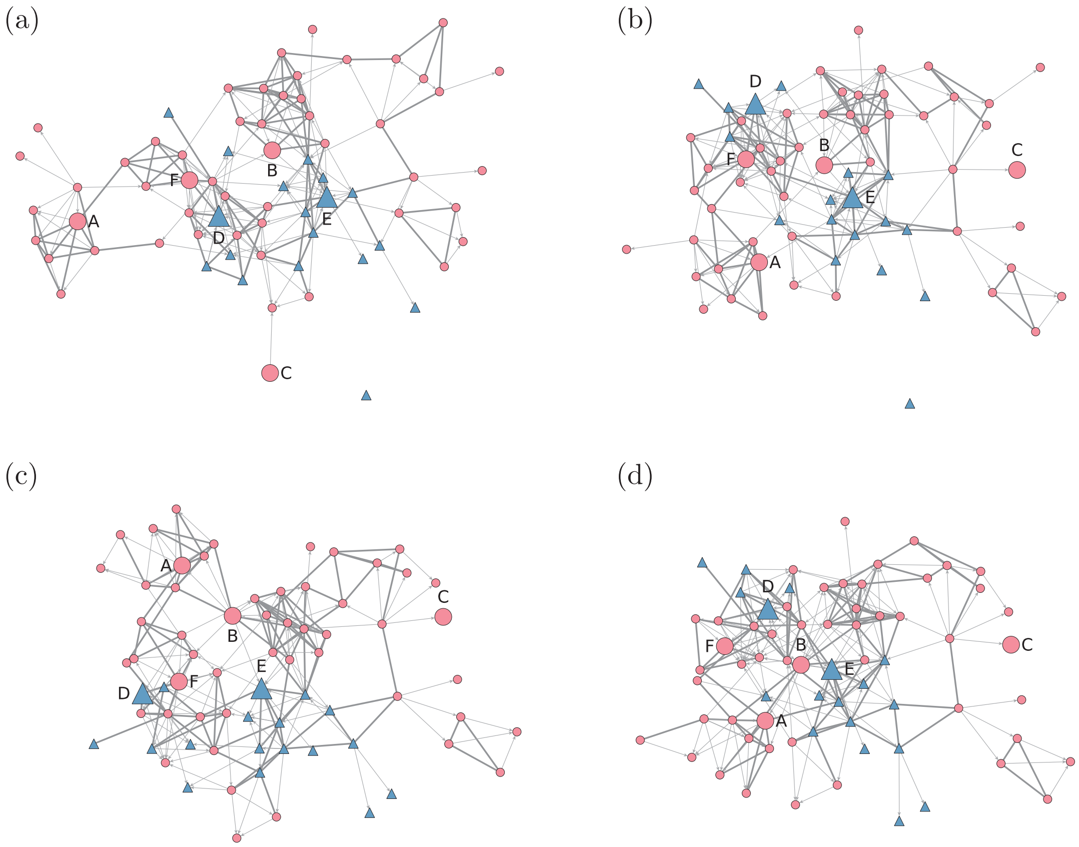


Fig. 6. Waves 10–13 of our running example drawn by a common online approach. Substructures are similar, but the offline method (Fig. 5) avoids the need for flipping subgroups such as the ones associated with Actors A and F from Wave 12 to 13. (a) Wave 10, (b) Wave 11, (c) Wave 13 and (d) Wave 12.

4. Stochastic actor-oriented models

Today, stochastic actor-oriented models (SAOMs) (see, e.g., Snijders, 2005, 2001) are the primary tool for the analysis of longitudinal social networks. Given panel data, i.e., sequences of network observations at discrete time points, as input, it is supposed that an unobserved evolution process took place inbetween consecutive states. The process model is composed of network-specific rules such as reciprocity, triadic closure, or homophily, and these rules are referred to as *network effects*. The aim of an analysis is to identify network effects that explain the observed evolution.

We briefly introduce the model in its simplest form, with only two consecutive network observations. For a detailed introduction and extended actor-oriented models see Snijders et al. (2010b, 2007) and Steglich et al. (2010).

Let $V = \{1, \ldots, n\}$ denote the fixed set of actors, and $\mathcal{A}_V$ the set of all adjacency matrices over V, i.e., for a matrix $A \in \mathcal{A}_V$ we have entry $a_{ij} = 1$ if there is a tie from i to j, and $a_{ij} = 0$ otherwise. Note that we are dealing with directed graphs from now on. The two observed networks are denoted by $A^{(pre)}, A^{(post)} \in \mathcal{A}_V$.

The evolution from $A^{(pre)}$ to $A^{(post)}$ is regarded as a stochastic sequence of elementary changes, called *micro-steps*, that are performed consecutively by randomly chosen actors. An actor is allowed to either create a single new outgoing tie, delete a single existing outgoing tie, or not change any tie at all. When an actor is designated to perform the next micro-step, probabilities of all feasible micro-steps are determined by the current network structure $A^{(cur)}$ and do not depend on previous states. These model assumptions are realized in a continuous-time Markov process on state space $\mathcal{A}_V$ that is starting from the first observation, $A^{(pre)}$. The transition probability from the current state $A^{(cur)}$ to a possible next state A is positive only if the two networks differ by at most one tie a_{ij} for which i is the actor designated to perfom the next micro-step. Since actors are assumed to strive for improving their position in the network, the transition probability from $A^{(cur)}$ to A depends on the position enhancement achieved by changing to a_{ij} rather than any other feasible option.

The assessment of an actor i's position in a network A is according to an objective function $f_i : \mathcal{A}_V \rightarrow \mathbb{R}$ that is assumed to be of the form:

$$f_i(\theta; A) = \sum_{k=1}^{K} \theta_k \cdot s_{ki}(A), \tag{3}$$

where each $(s_{k1}, \ldots, s_{kn})^T$, $1 \leq k \leq K$, is a vector of statistics representing network effects, and $\theta = (\theta_1, \ldots, \theta_K) \in \mathbb{R}^K$ are associated model parameters. The latter are fitted to the observed data, assuming that actors strive to maximize their objective function. Uncertainty stemming from the potential influence of chance and other unknown factors is usually represented by a random term added to the objective function, but is not important for the purpose of this paper.

Specifically, statistics $s_{ki}(A)$ count the number of occurrences of configurations in A that actor i is part of, and thus represent the k th network effect. The selection of network effects is referred to as the *model specification*. Some common statistics are the number of

outgoing ties:	$s_{1i}(A) = \sum_{j=1}^{n} a_{ij}$
reciprocated ties:	$s_{2i}(A) = \sum_{j=1}^{n} a_{ij} a_{ji}$
transitive ties:	$s_{3i}(A) = \sum_{j=1}^{n} a_{ij} \cdot \max_{l \neq j,i}(a_{il} a_{lj})$
actors at distance two:	$s_{4i}(A) = \sum_{j=1}^{n} (1 - a_{ij}) \cdot \max_{l \neq j,i}(a_{il} a_{lj})$
transitive triplets:	$s_{5i}(A) = \sum_{j=1}^{n} \sum_{l=1}^{n} a_{ij} a_{il} a_{lj}$
3-cycles:	$s_{6i}(A) = \sum_{j=1}^{n} \sum_{l=1}^{n} a_{ij} a_{jl} a_{li}$

Such statistics indicate local rules of network evolution. Note that the entire formulation implies a rather strong assumption, namely that actors are homogenous in the sense that the same rules are working for them in the same way.

For illustration, consider three actors i, j_1, j_2 with i intending to connect with either j_1 or j_2. Assume that the network positions of j_1 and j_2 are equivalent, except that there exists a tie from j_1 to i, but not from j_2. In this setting, a positive reciprocity parameter (i.e., a positive weight θ_2 for the above s_{2i}) implies that i is more likely to connect with j_1 than with j_2. A negative parameter would imply the opposite. In the following, we identify a model with its parameters θ.

Given a model θ and an initial network $A^{(pre)}$, the evolution predicted by θ can be simulated via a sequence of micro-steps conforming to the above rules. We denote such a sequence by $\mathcal{S}(\theta, A^{(pre)})$. One possible condition for terminating the simulation is that the current network state $A^{(cur)}$ differs from $A^{(pre)}$ in the same number of dyads as $A^{(post)}$. The probability that a network $A \in \mathcal{A}_V$ is the final state of a simulation $\mathcal{S}(\theta, A^{(pre)})$ depends on θ, so that θ induces a probability distribution over all networks in $\mathcal{A}_V$ conditional on $A^{(pre)}$.

Among several methods for fitting θ to observed data (Snijders, 2001; Snijders et al., 2010a; Koskinen and Snijders, 2007), the one most commonly used is an instance of the *Method of Moments* (Snijders, 2001).[2] Parameters are determined to match statistics:

$$s_k(A^{(post)}) := \sum_{i=1}^{n} s_{ki}(A^{(post)}), \qquad 1 \le k \le K, \tag{4}$$

of the second observation $A^{(post)}$ as closely as possible to their expected values in predicted networks. Since the model is too complicated to determine expected values precisely, they are estimated from simulations. Therefore, the aim is finding parameters θ that satisfy the *moment equations*

$$E_\theta[S_k] = s_k(A^{(post)}), \qquad 1 \le k \le K, \tag{5}$$

where S_k is a random variable for the value of s_k in a network resulting from a simulation $\mathcal{S}(\theta, A^{(pre)})$.

5. Diagnostic visualization

Based on the view that a SAOM θ induces a probability distribution over $\mathcal{A}_V$, we say that θ's predictions fit the observed data if $A^{(post)}$ is highly probable under the distribution implied by θ, i.e., if it is likely that $A^{(post)}$ is the result of simulations $\mathcal{S}(\theta, A^{(pre)})$.

There are several potential causes for poor representation of data. The main issue is model mis-specification, i.e., the inclusion of effects that are not present in the data or exclusion of effects that actually drive the evolution process. Inappropriate specification is often evidenced by parameter estimates that exhibit large standard errors or, in extreme cases, by non-convergence of the iterative estimation process. A problem that is even more difficult to cope with is inhomogeneity across actors, because it is hidden in the aggregated statistics (4). The fact that parameters θ do not depend on individual actors, but are estimated in order to fit the average behavior of all actors by satisfying the moment equation (5), basically suggests that all actors follow the same rules, except for differences implied by covariates. In many cases, this restrictive assumption is unjustified and leads to poor models. However, poor fit caused by inhomogeneity is often not apparent from estimates and their standard errors because parameters inferred from the aggregate quantities in (5) do not necessarily exhibit a high level of uncertainty, even if occurrences of statistics are distributed very inhomogeneously across actors.

In this section, we introduce methods that allow for visual exploration of model predictions and comparison with observed data. The first method reveals a global impression of model predictions that can be compared with the observed data in order to detect regions or structural patterns that are poorly represented. The second method is especially designed for assessing actor-inhomogeneity. Contrary to the first method, where the focus is on the global structure of connections, the second method considers local statistics in the neighborhood of individual actors. These graphical methods support model interpretation and can provide guidance in proper effect selection in addition to the significance tests (Schweinberger, submitted for publication) already integrated into the Siena program (Ripley and Snijders, 2011).

5.1. Overall fit

Our first goal is a graphical representation of the predictions made by a model. To assess how well such predictions and the data that was actually observed align, the representation shall be comparable to related visualizations of the observed data. What, however, is an appropriate graphical representation for a *distribution* of networks? And how could it be displayed whether a network is typical for a given distribution or not?

Since the actual distribution is computationally intractable, we start by sampling from it. Let $A^{(1)}, \ldots, A^{(R)} \in \mathcal{A}_V$ be a set of network samples resulting from R simulations $\mathcal{S}(\theta, A^{(pre)})$. In order to reveal shared characteristics of these samples, such as clustering, we follow the idea of the stress minimization approach described in Section 3.1 where, too, information on structural features of several networks is aggregated into a single layout.

Even in the context of networks arising from uncoordinated myopic decisions of independently acting actors, the following arguments substantiate why an aggregate layout approach may be appropriate. Local actor decisions indirectly depend on each other, since previous dynamics formed the structure that subsequent changes depend on. Therefore, actor-oriented models determine not only local dynamics, but also the global evolution of network characteristics. Purely local effects such as, for instance, reciprocation and transitive closure, thus reinforce existing global features such as segmentations into cohesive substructures and structural holes. Since the given observation $A^{(pre)}$ seeds an initial global structure, it is plausible that a model can not only predict occurrences of structural patterns but also their topological distribution.

In analogy to the stress in Eq. (2), we therefore define a stress function stress_{sim} based on shortest-path distances between each pair of vertices in all simulated networks simultaneously. Observe that it does not matter that simulation runs do not have a meaningful ordering as the observations do. Because they are determined by the same principles, comparison of such drawings with a layout of $A^{(post)}$ based on the regular stress function yields insight into structural similarities or differences.

Let $\delta_{ij}^{(1)}, \ldots, \delta_{ij}^{(R)}$ denote the lengths of undirected shortest paths between vertices i and j in the graphs of $A^{(1)}, \ldots, A^{(R)}$. If i and j are in different connected components of $A^{(r)}$, we replace the conventional infinite distance by $\delta_{ij}^{(r)} = n$, so that all distance between distinct vertices are in the range $\{1, \ldots, n\}$. Similarly to the aggregated stress function (2), we define a stress function based on samples $A^{(1)}, \ldots, A^{(R)}$ by

$$\text{stress}_{sim}(P) = \sum_{i<j} \omega_{ij} (\bar{\delta}_{ij} - \| p_i - p_j \|)^2, \tag{6}$$

[2] An implementation is available in the R package RSiena, see http://cran.r-project.org/web/packages/RSiena/.

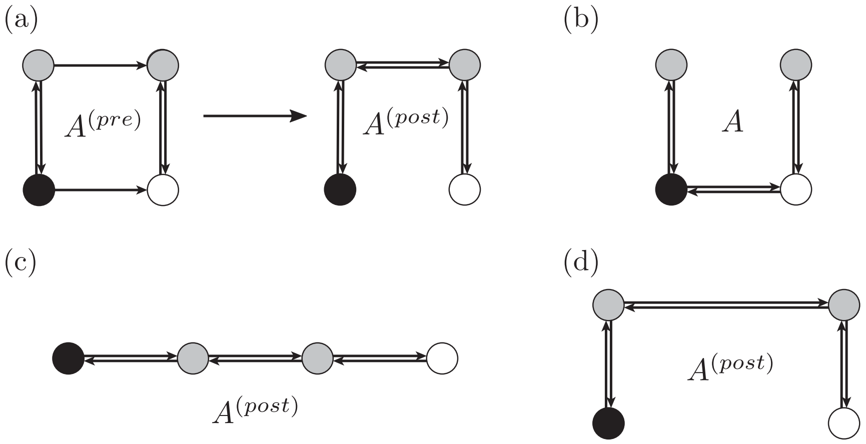


Fig. 7. How to read layouts based on model predictions (see text). (a) Longitudinal network data, (b) equally probable outcome, (c) layout using observed distances and (d) layout using expected distances.

where $\bar{\delta}_{ij} := \frac{1}{R}\sum_{r=1}^{R}\delta_{ij}^{(r)}$ denotes the sample mean of simulated distances between vertices i and j. As in Section 2.2, local accuracy should be emphasized by reducing the impact of distant dyads $\{i, j\}$ identified by high sample means $\bar{\delta}_{ij}$. Our application also requires additional emphasis on the influence of highly confirmed distances indicated by small sample variances $s^2(\delta_{ij}) := \frac{1}{R-1}\sum_{r=1}^{R}(\delta_{ij}^{(r)} - \bar{\delta}_{ij})^2$. We therefore propose to use weights

$$\omega_{ij} := \frac{1}{(1 + s^2(\delta_{ij}))\bar{\delta}_{ij}^2},$$

where the newly introduced factor $1/(1 + s^2(\delta_{ij}))$ increases the bias towards faithful representation of local structures, since dyads with higher average distances usually exhibit larger standard deviations.

Minimization of stress$_{sim}$ yields a layout in which vertices that are likely (unlikely) to be connected by a short path are close together (far apart). Note that (6) does not include any data of the second observation $A^{(post)}$, but only information resulting from model-based simulations starting from $A^{(pre)}$. However, if all sampled networks $A^{(r)}$ were identical to $A^{(post)}$, stress$_{sim}$ would reduce to the basic stress function of $A^{(post)}$ for static stress minimization as introduced in Section 2.2. The layout of $A^{(post)}$ based on stress$_{sim}$ would then be equal to the layout based on graph-theoretic distances. Therefore, in case of excellent and detailed model predictions, both methods would yield essentially the same layout of $A^{(post)}$. In particular, a drawing of $A^{(post)}$ based on stress$_{sim}$ would exhibit short edges, and large distances between pairs of unconnected vertices. Long edges, on the other hand, appear where graph-theoretic distances are small in the observation $A^{(post)}$, but not in the samples $A^{(1)}, \ldots, A^{(R)}$.

We stress that the assessment of model predictions should be in line with the inferential goal of SAOMs. This is not the prediction of individual ties, but the characterization of local processes by a limited number of network effects. Consider, for instance, the longitudinal network of Fig. 7(a) together with a model $\bar{\theta}$ consisting of only two effects, `outgoing ties` and `reciprocated ties`. In this setting, it is equally probable that a simulation $\mathcal{S}(\bar{\theta}, A^{(pre)})$ ends up in $A^{(post)}$ or in network A depicted in Fig. 7(b). Therefore, simulations resulting in A do not necessarily invalidate the model even if A differs from $A^{(post)}$ in one third of all possible ties. Knowledge about poorly predicted ties may well be important for model evaluation, though, because it may inform the selection of effects.

To illustrate how our approach can be used for this kind of assessment, let us compare a layout of $A^{(post)}$ based on observed shortest-path distances in Fig. 7(c) with a layout of expected shortest-path distances in Fig. 7(d). Comparing both layouts, we recognize that the reciprocated edges connecting the two gray vertices in Fig. 7(d) are longer than other reciprocated edges in the same graph as well as corresponding edges in Fig. 7(c). Moreover, compared to their positions in Fig. 7(c), the white vertex and the black vertex are closer in Fig. 7(d). The fact that both pairs are equidistant in Fig. 7(d) implies that, although only one pair is connected in $A^{(post)}$, both connections are equally probable under the model $\bar{\theta}$. As already stated, this does not necessarily translate into poor model-fit, but that the model fails to predict specific connections between these actors reliably. If context calls for more detailed explanations of observed relations, this may be seen as a hint at the inclusion of other effects, possibly based on actor covariates.

To facilitate comparison between layouts for $A^{(post)}$ according the observed (stress) and expected (stress$_{sim}$) distances, both computations are initialized with positions from a layout of $A^{(pre)}$ according to stress. While this ensures a certain degree of similarity, we refrain from adding stability via anchoring, because this would interfere with our reading of the simulation result. Instead, layouts are aligned using Procrustes rotation, which does not alter relative distances.

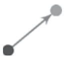
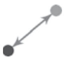
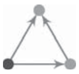
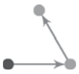
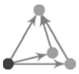
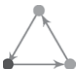
We apply the method to the sixth and thirteenth observation of our running example. Note however that in case $A^{(pre)}$ is disconnected, connections between different components are hardly predictable by a SAOM. As a consequence, the positioning of separate components can be quite arbitrary and therefore misleading. Since this is particularly true for singleton components, we omit the five isolated actors of Wave 6, but continue to refer to the reduced data as Waves 6 and 13.

The data was subjected to the three models listed in Table 1, with Wave 6 as the initial and Wave 13 as the target observation. The three models are increasingly complex. While Model 1 is elementary and dyadic, the additional effects of Model 2 generate triadic closure and thus the formation and reinforcement of clusters. Model 3 is a further refinement representing the formation of local hierarchies (if signs are as in our example).

The layout in Fig. 8(a) serves as initialization for visualizing model predictions by minimizing the modified stress function stress$_{sim}$. The resulting layout of Wave 13 based on simulations of Model 2 is depicted in Fig. 9(a). To allow direct comparison, Wave 13 laid out according to stress is shown once more in Fig. 9(b). Layouts have been aligned by Procrustes rotation.

Informally speaking, the clustering in the layouts of Fig. 9 is quite similar. The cohesive group of female students including Actor Y is easily identified in both. Other groups such as the males around Actor Q and Actor S, the females at the top left, or the mixed

Table 1
Analyzed models with parameters θ estimated using the `RSiena` package (Ripley and Snijders, 2011) for $A^{(pre)}$ = Wave 6 and $A^{(post)}$ = Wave 13 (ignoring isolates of Wave 6).

	Network effect	Model 1		Model 2		Model 3	
		θ	(S.E.)	θ	(S.E.)	θ	(S.E.)
	outgoing ties	−1.61	(0.10)	−1.95	(0.23)	−2.16	(0.23)
	reciprocated	2.64	(0.20)	1.96	(0.22)	2.60	(0.29)
	transitive ties			1.37	(0.24)	1.05	(0.26)
	actors at distance 2			−0.26	(0.06)	−0.22	(0.06)
	transitive triplets					0.36	(0.09)
	3-cycles					−0.65	(0.16)

group next to Actor U can be matched as well. However, from their poor representation in Fig. 9(a) we can conclude that several ties between groups as well as ties connecting marginal vertices like V or T with the center of the network are not predicted well by the model.

Another salient disparity is the location of Actor W who has a bridging position between two groups in Fig. 9(b), but is close to only one group in Fig. 9(a) This indicates that the model performs badly in predicting the creation of ties between W and the group at the top left. Note that there were no such connections in Wave 6. The position of Q can be explained in a similar way.

A weakness of this approach is that it is not always obvious whether a poor layout, indicated by long edges and close unconnected vertices, is caused by poor model predictions or by desired distances that are just difficult to realize in a two-dimensional drawing. Therefore, Fig. 10 shows two complementary diagrams that are meant to improve interpretability of the layout from Fig. 9(a).

Fig. 10(a) displays sample means $\bar{\delta}_{ij}$ of simulated graph-theoretic distances between all dyads. E.g., the position of R in Fig. 9(a) close to the above female group suggests that connections between this group and R are likely. But in Fig. 10(a), the two more suspicious edges incident to R reveal that its position is rather caused by the realization of desired distances to vertices below it. Note that peripheral vertices such as V or T, exhibit large average graph-theoretic distances to all other vertices.

Additional information on model predictions is obtained by considering for each pair of vertices i and j the probability π_{ij} that edge a_{ij} exists in a simulated network A. This probability can be estimated by counting the number of simulated networks in which i and j are adjacent, i.e.,

$$\pi_{ij} := \frac{1}{R}\sum_{r=1}^{R} a_{ij}^{(r)}.$$

Even though our previous considerations implied that assessment of model predictions must not be based solely on individual dyads, the elementary comparison of entries $a_{ij}^{(post)}$ with the predicted probabilities π_{ij} can still be indicative of the plausibility of a model, where especially the coincidence of high values of π_{ij} with absent ties $a_{ij}^{(post)}$ points at poor predictions. Moreover, correspondence of tie probabilities with distances between vertices in the layout implied by stress_{sim} is another indicator for assumptions on model predictions.

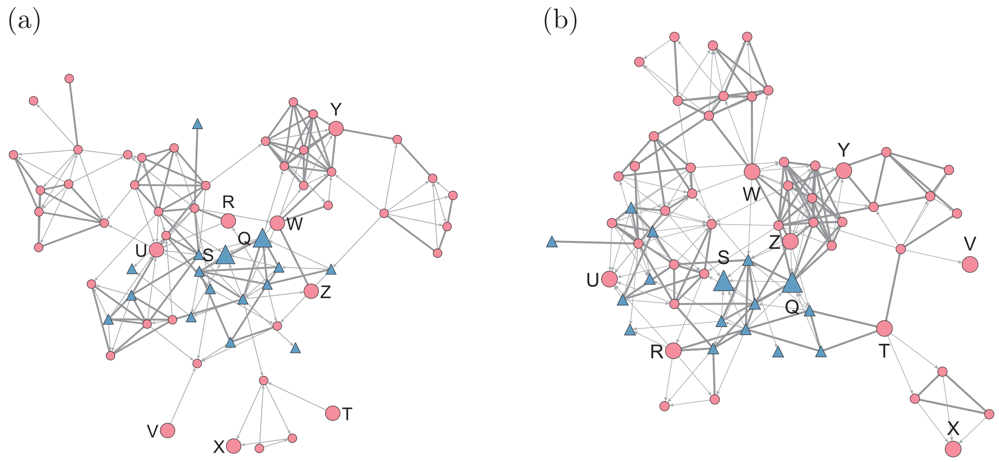


Fig. 8. Stress minimization layouts of Wave 6 and Wave 13 based on graph-theoretic distances. Vertices that are isolated in Wave 6 are omitted in both networks. (a) Wave 6 and (b) Wave 13.

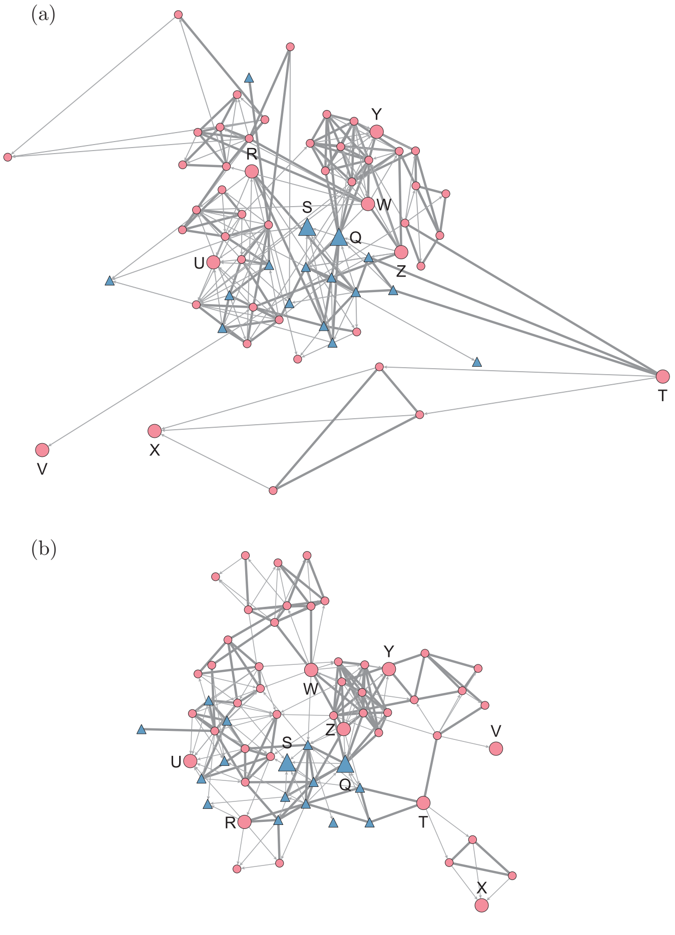


Fig. 9. Comparison between layouts of Wave 13 based on simulated and observed distances. The layout in (a) results from modifying Fig. 8(a) by minimizing stress$_{sim}$ based on Model 2. The layout in (b) is determined by using regular stress minimization based on graph-theoretic distances. (a) Layout based on simulations of Model 2 and (b) layout based on graph-theoretic distances.

Fig. 10(b) displays tie probabilities π_{ij} implied by Model 2. In general, it indicates that most edges are very improbable, and that most of the highly probable edges are within groups that existed previously while edges between groups are rather unlikely with few exceptions of moderate probability. This is indeed what we would expect from a model in which existing groups are reinforced by positive triadic-closure and reciprocity parameters, and a negative density and distance-2 parameter.

The basic structures in Fig. 10(b) are, as expected, very similar to those in Fig. 10(a). But there are also differences such as the moderate probabilities of connections between Actor X and two others, which are visible in Fig. 10(b) but not in Fig. 10(a). Apparently, X is isolated in many simulated networks so that the average graph-theoretic distances to the two other actors, and so the Euclidean distances in the layout, are large, even though direct connections are fairly probable. Due to this shortcoming, our method may not be suitable for extremely sparse networks or for longitudinal data with great differences between consecutive observations. Our experiences so far suggest that a Jaccard index (see Snijders et al., 2010b) higher than 0.5 is a reasonable threshold.

Finally, we use our method to compare the three models listed in Table 1. Fig. 11 shows for each model the layout determined in the same way as in Fig. 9(a). Compared to Fig. 11(b) and (c), where the layouts reveal several clusters of vertices, vertices in Fig. 11(a) are homogeneously distributed. This indicates that the effects included in Model 1 are not sufficient to predict the observed clustering that, on the other hand, is clearly identifiable in the predictions of Model 2 and Model 3. The additional effects included in Model 3 reinforce the basic structure already implied by Model 2, but achieve no remarkable improvement. In conclusion, Model 2 appears to strike the best balance between fit and parsimony.

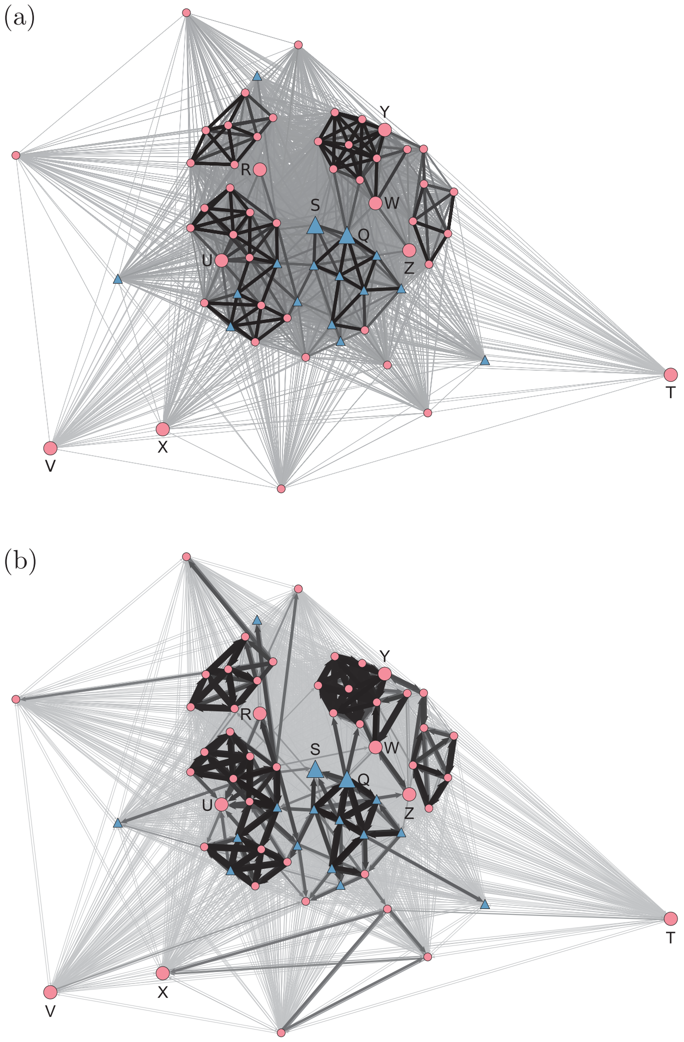


Fig. 10. Sample means of shortest paths and tie probabilities implied by simulations of Model 2. Color intensities and widths of edges increase with their probability of existence π_{ij} in (b). In (a) they increase with the inverse of the expected shortest path $1/\bar{\delta}_{ij}$ between connected vertices.

5.2. *Inhomogeneity*

From analyzing the overall configuration predicted by a model, we now shift focus to individual actors. The methods below facilitate model diagnostics in general, and validation of actor homogeneity assumptions in particular. Thereby, they may support improvement of a model specification.

Stochastic actor-oriented models are estimated in order to obtain simulated networks that resemble second observation $A^{(post)}$ in terms of aggregated statistics s_k defined in Eq. (4). Considering, e.g., the `outgoing ties` effect, simulated networks should contain the same number of ties as $A^{(post)}$. Whether these ties are homogeneously distributed over the entire network or clustered around some hubs, however, is not differentiated by the summarizing moment Eq. (5). One way to deal with inhomogeneities is the inclusion of additional network effects such as popularity or similarity effects. As a prerequisite, however, the analyst must first become aware of them.

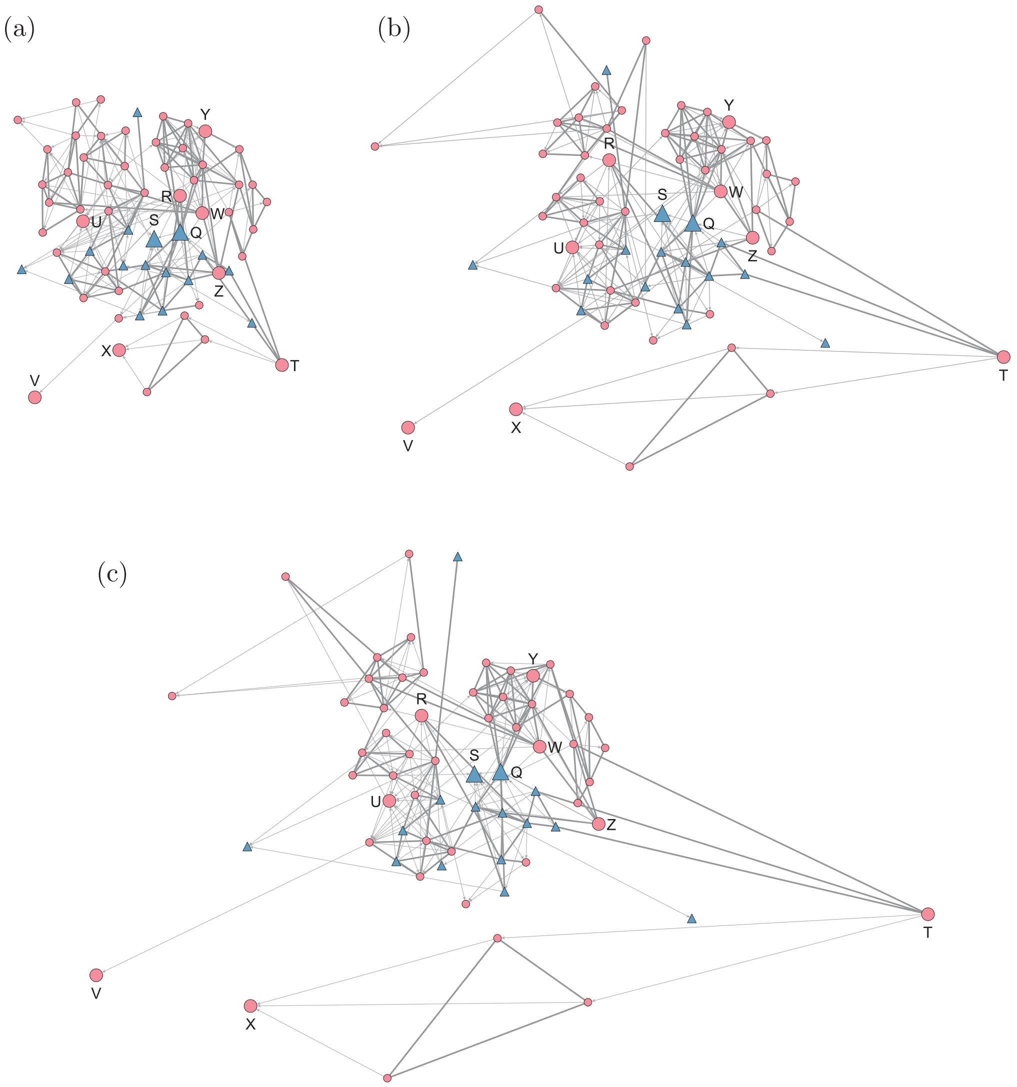


Fig. 11. Layouts of Wave 13 based on simulated networks predicted by (a) Model 1, (b) Model 2, and (c) Model 3.

In the context of exponential-family random graph models (ERGMs), Hunter et al. (2008) addresses the problem of potential inhomogeneity by comparing distributions of statistics (e.g., the degree distribution) of simulated and observed networks. In contrast to ERGMs, however, where simulations start from a random network and are made to converge to a stationary distribution, SAOMs start from a fixed initial network $A^{(pre)}$, from which a finite number of micro-steps leads to the resulting simulated network. Consequently, actors are distinguishable by their position and the structure of their environment in $A^{(pre)}$. This justifies, in fact, the assessment of model predictions on the local level of individuals, as an alternative to the global analysis of distributions of statistics. To identify actors or groups of actors deviating from model predictions, we turn to the fit of statistics in the neighborhoods of individual actors.

Let θ denote a model based on objective functions $f_i = \sum_{k=1}^{K} \theta_k s_{ki}$ and let $A^{(1)}, \ldots, A^{(R)} \in \mathcal{A}_V$ be network samples resulting from R simulations $\mathcal{S}(\theta, A^{(pre)})$. As in Section 5.1, these sampled networks are regarded as representative for model predictions. For each actor, we determine sample means

$$\bar{s}_{ki} = \frac{1}{R} \sum_{r=1}^{R} s_{ki}(A^{(r)})$$

of statistics associated with the effects included in the model. To assess model predictions for individual actors, we are interested in deviations of these sample means from observed values. Differences

$$\Delta_{ki} := s_{ki}(A^{(post)}) - \bar{s}_{ki}$$

indicate deficiencies of model predictions for actor i in terms of the network effect s_k. While differences are comparable across actors, they are not comparable across effects, because the latter operate on variable scales. Therefore, relative deviations $\tilde{\Delta}_{ki}$ are determined by dividing Δ_{ki} by estimated standard deviations of corresponding statistics in the entire network, i.e.,

$$\tilde{\Delta}_{ki} := \frac{\Delta_{ki}}{\sqrt{\mathrm{VAR}_k}}$$

with

$$\mathrm{VAR}_k = \frac{1}{R} \sum_{r=1}^{R} \left(\sum_{i=1}^{n} (s_{ki}(A^{(r)}) - \bar{s}_{ki}) \right)^2 .$$

This normalization is chosen in analogy to the t-statistics used for checking the convergence of the parameter estimation algorithm

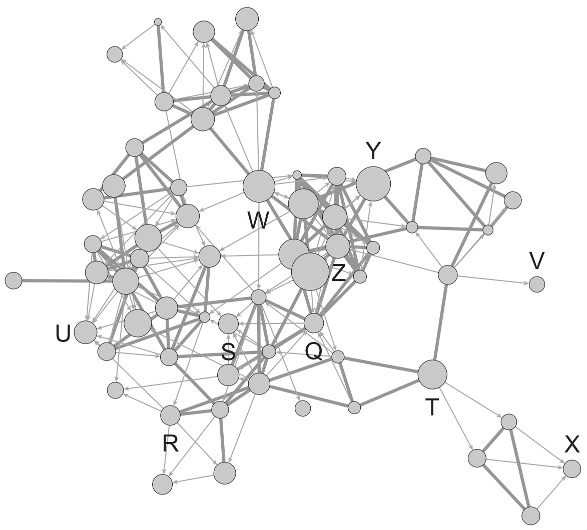


Fig. 12. Wave 13 of the acquaintanceship network with vertex sizes indicating sums of absolute deviations of simulated and observed statistics associated with the effects included in Model 2. The larger a vertex, the larger its lack of fit.

(see Ripley and Snijders, 2011), which equal the sums $\sum_{i=1}^{n}\tilde{\Delta}_{ki}$ of respective relative deviations.

As a heuristic summary measure for the quality of model predictions for actor i, we here use the sum of absolute values of relative deviations $|\tilde{\Delta}_{ki}|$ of all considered statistics,

$$\|\tilde{\Delta}_i\|_1 := \sum_{k=1}^{K}|\tilde{\Delta}_{ki}|,$$

although other definitions such as a generalized Mahalanobis distance incorporating covariances between effects are conceivable as well.

Fig. 12 shows Wave 13 of the acquaintanceship network with vertex areas proportional to the associated values $\|\tilde{\Delta}_i\|_1$ resulting from Model 2 (see Table 1). Recall that these represent differences of four effects, outgoing ties (OT), reciprocated ties (RT), transitive ties (TT), and actors at distance 2 (AD2), between the actual statistics in Wave 13 and those in simulations starting from Wave 6.

This helps with comparing the quality of predictions across actors and detecting individuals or groups whose behavior is particularly deviant from model predictions. In Fig. 12, Z is most salient. But also the bridging vertices Y, W, and T are subject to a mismatch between simulated and observed statistics. From a global viewpoint, vertices of the cohesive group around Z and, in attenuated form, vertices of the group next to U exhibit particularly high deviations. However, the causes for the observed inhomogeneities and therewith possible model improvements are not apparent from this visualization since it does not point to the effects that are responsible for poor prediction. This motivates the extension of the above idea to separate visualizations for different effects with sizes of vertices proportional to their relative deviation $\tilde{\Delta}_{ki}$ and colors of vertices indicating the direction of deviation, i.e., whether effects are over- or underestimated by model predictions.

Applied to effects not yet included in the model, such diagrams may indicate whether they are sufficiently represented by already included effects or whether they should be included themselves. For effects considered in the model, it is expected that the sum of deviations over all actors satisfies

$$\sum_{i=0}^{n}\tilde{\Delta}_{ki} \approx 0, \qquad (7)$$

since parameters θ are estimated to satisfy Eq. (5). But deviations of single actors or subgroups can compensate each other mutually, such that (7) can be true even though individual deviations are large. Inhomogeneities thus require detailed inspection as provided in our visualizations.

Moreover, there may be actors whose behavior with respect to some effect is hardly determined by the model such that deviations from observed values are caused by uncertainty rather than poor prediction. For other actors the implied probability distribution may admit only slight variations of statistics such that deviations from observed values indeed contradict the model predictions. The degree of uncertainty is generally indicated by sample standard deviations of respective statistics. The ratio $\Delta_{ki}/\text{SSD}_{ki}$, where

$$\text{SSD}_{ki} = \sqrt{\frac{1}{R-1}\sum_{i=1}^{R}(s_{ki}(A^{(r)}) - \bar{s}_{ki})^2}$$

is the sample standard deviation of local statistics, indicates whether it is likely that observed statistics are sampled from the model-implied probability distribution. High values indicate contradictions with model predictions, whereas low values can either result from fitting predictions that yield low values of Δ_{ki}, or from indefinite predictions that yield high values of SSD_{ki}.

Figs. 13 and 14 show Wave 13 with graphical vertex attributes representing the quality of model predictions for individual actors regarding the four effects OT, RT, TT, and AD2. Visualizations in left columns illustrate results obtained from Model 1. Right columns contain results of Model 2. Areas of vertices are proportional to absolute values of relative deviations $\tilde{\Delta}_{ki}$. Color indicates the sign of $\tilde{\Delta}_{ki}$, i.e., whether occurrences of effect-related configurations are under- (blue) or overestimated (red) by model predictions. Color intensity depicts the ratio $\Delta_{ki}/\text{SSD}_{ki}$ of deviation divided by standard deviation, so that intensely red- or blue-colored vertices hint at contradictions with model predictions. Light colors are either caused by small deviations Δ_{ki} or by large standard deviations SSD_{ki}, where the two causes can be distinguished by the size of vertices. Vertices are white if the absolute value of deviation is lower than the standard deviation. Note that a light-colored vertex does not necessarily indicate a correct prediction but, depending on its size, rather indefinite predictions about the associated actor.

Figs. 13b and d, and 14b and d, which represent Model 2, allow to distinguish effects responsible for remarkable deviations in Fig. 12. While all effects contribute to the deviations of strikingly prominent Actors Z and T by underestimating their activity, W is only underestimated in terms of effects OT, TT, and AD2, but not in terms of RT. Likewise, effect AD2 does not contribute to the overestimation of Y.

The balanced coloring in the graphs representing Model 2 implies that (7) approximately holds for all effects.[3] Regarding Model 1, the same applies to Fig. 13(a) and (c) representing effects OT and RT. For these effects, which are included in both models, differences between the two models are hardly perceivable (see Fig. 13). For TT and AD2, in contrast, the two models yield widely differing results (see Fig. 14). Visualizations obtained from Model 1, which includes neither of the two effects, are highly imbalanced. Compared to visualizations of Model 2, shifts to the extreme range of the color scale are noted. According to the algebraic signs of estimates in Model 2, the dominant blue in Fig. 14(a) indicates that the observed data contains considerably more transitive ties than Model 1 predicts, whereas the dominant red in Fig. 14(c) indi-

[3] Actually, the graphs exhibit a few more red than blue vertices but most blue vertices are slightly larger. Hence, absolute values of positive deviations are larger so that the higher number of negative deviations is still balanced.

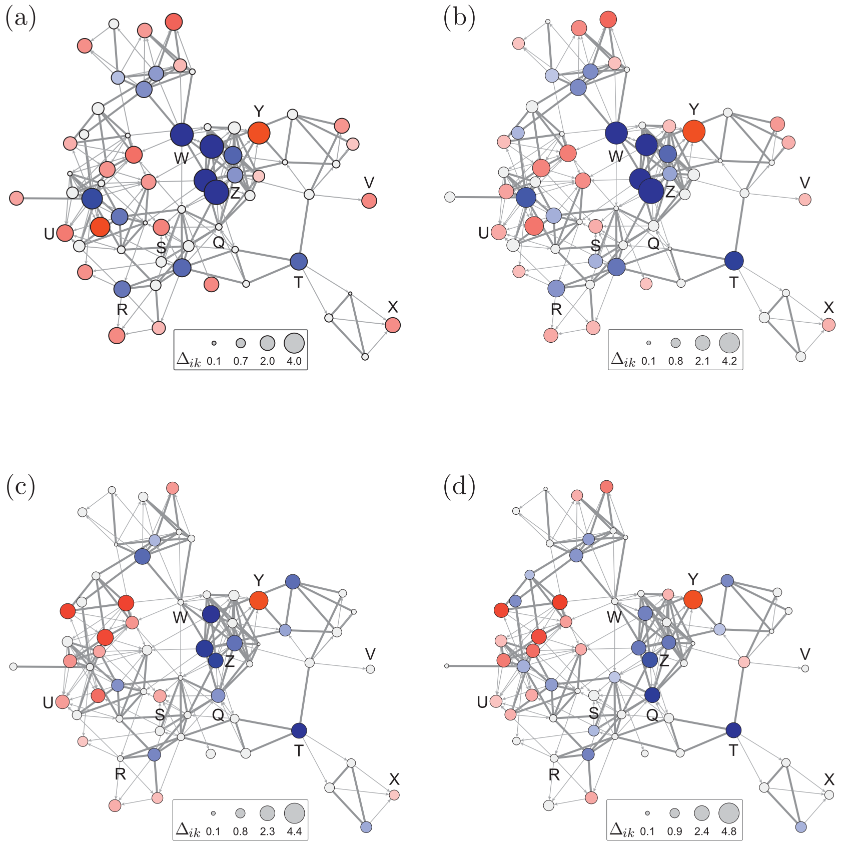

(a)
Y
W
V
Z
U
S
Q
T
R
X
Δ_{ik} 0.1 0.7 2.0 4.0
(b)
Y
W
V
Z
U
S
Q
T
R
X
Δ_{ik} 0.1 0.8 2.1 4.2
(c)
Y
W
V
Z
U
S
Q
T
R
X
Δ_{ik} 0.1 0.8 2.3 4.4
(d)
Y
W
V
Z
U
S
Q
T
R
X
Δ_{ik} 0.1 0.9 2.4 4.8

Fig. 13. Goodness-of-fit of Model 1 and Model 2 in terms of inhomogeneity analyzed for individual network effects. The framed boxes in each diagram indicate the relation between vertex sizes and deviations Δ_{ki}. The color-scale -3 -1 1 3 represents values $\Delta_{ki}/\text{SSD}_{ki}$. (a) Model 1: outgoing ties (OT), $\sum_i \Delta_{ki} = 0.30$, $\sqrt{\text{VAR}_k} = 10.41$; (b) Model 2: outgoing ties (OT), $\sum_i \Delta_{ki} = 1.45$, $\sqrt{\text{VAR}_k} = 10.85$; (c) Model 1: reciprocated ties (RT), $\sum_i \Delta_{ki} = 0.49$, $\sqrt{\text{VAR}_k} = 11.53$; (d) Model 2: reciprocated ties (RT), $\sum_i \Delta_{ki} = 0.60$, $\sqrt{\text{VAR}_k} = 12.41$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

cates an overestimation of the number of indirect neighbors. It is remarkable that many vertices indeed have an intense coloring but their moderate sizes indicate at the same time only small relative deviations $\tilde{\Delta}_{ki}$. Thus, model predictions for these actors must be quite definite which results in high values $\Delta_{ki}/\text{SSD}_{ki}$ due to minor standard deviations.

A closer look at the visualizations reveals that Model 2 outperforms Model 1 not only in terms of TT and AD2 but also in terms of OT and RT. Especially, predictions for actors with considerably more incoming than outgoing ties in both observations such as U, S, or X are improved. This is in line with the parameter estimates of the two models. Actor S, for example, has six incoming but no outgoing ties in Wave 6 and Wave 13, which results in an objective function value of zero. In Model 1, this value is increased by 1.03 for S, if an incoming tie is reciprocated. Hence, reciprocation is likely, and this leads to an overestimation of OT and RT. In Model 2, any new tie that S builds decreases his objective function because of the negative parameter estimate for AD2. Since no newly created tie would be a transitive tie but most would increase the number of indirect neighbors, the observed behavior of S gets more likely under Model 2, as conveyed in the visualizations.

The apparent improvement of model predictions after inclusion of TT and AD2 implies that these effects are not represented indirectly in Model 1. This is further substantiated by the fact that for OT and RT model predictions are hardly affected by the addition of the two other effects. Obviously, correlations between the effects are only moderate, so that it seems advisable to include TT and AD2 in the model. A similar comparison between Model 2 and Model 3 reveals that the additional inclusion of the effects `transitive triplets` and `3-cycles` yields only slightly improved predictions for a few actors.

6. Discussion

We have presented methods for explorative and diagnostic visualization of longitudinal social networks.

Explorative visualization of longitudinal social networks was treated as an offline dynamic graph-drawing problem, and a corresponding instantiation of the generic stress-minimization framework for graph layout was proposed. Using anchoring at reference positions from a layout of the aggregate graph, intermediate layouts produced by our method represent a compromise between individual layout quality and persistence of an overall organization.

In summary, this method serves to visually explore the evolution of a dynamic network. Aggregation facilitates getting an overview of global structure, whereas anchoring provides explicit control over the balance of readability and stability, and therefore allows to put emphasis on the specifics of individual networks in the sequence. It should be noted, though, that the approach is based on the implicit assumption that there actually exists a relatively constant global organization.

Directions for future work therefore include the development of variant approaches for other scenarios by taking advantage of the flexibility of stress minimization. Also, more experimentation is needed to guide tuning of involved parameters by considering the type of structural change present in the longitudinal network at hand.

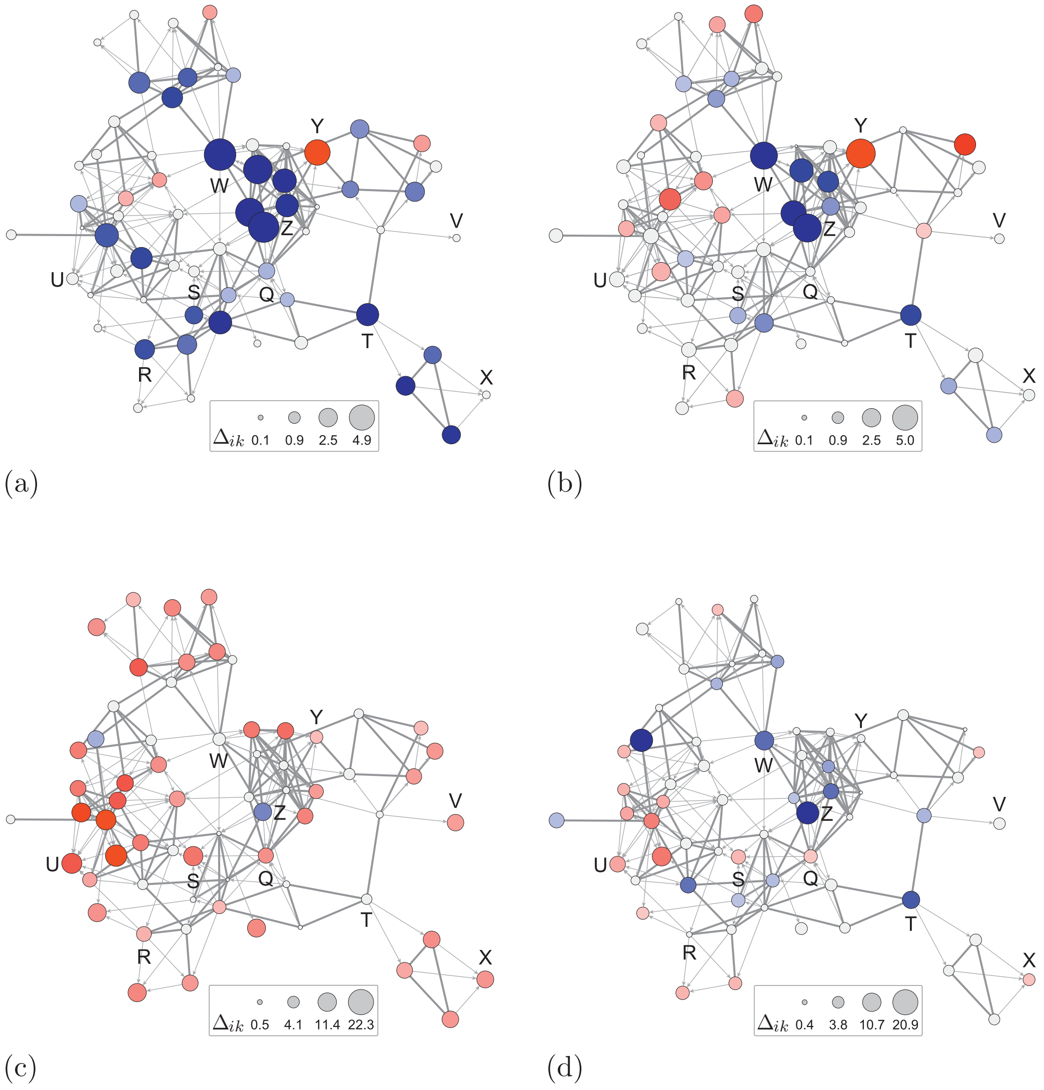


Fig. 14. Goodness-of-fit of Model 1 and Model 2 in terms of inhomogeneity analyzed for individual network effects. The framed boxes in each diagram indicate the relation between vertex sizes and deviations Δ_{ki}. The color-scale -3 -1 1 3 represents values $\Delta_{ki}/\text{SSD}_{ki}$. (a) Model 1: transitive ties (TT), $\sum_i \Delta_{ki} = -79.80$, $\sqrt{\text{VAR}_k} = 12.85$; (b) Model 2: transitive ties (TT), $\sum_i \Delta_{ki} = 1.48$, $\sqrt{\text{VAR}_k} = 13.03$; (c) Model 1: actors at distance 2 (AD2), $\sum_i \Delta_{ki} = 362.87$, $\sqrt{\text{VAR}_k} = 58.36$; (d) Model 2: actors at distance 2 (AD2), $\sum_i \Delta_{ki} = 1.40$, $\sqrt{\text{VAR}_k} = 54.61$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

In the modeling-related part of the paper, we have shown that layouts representing predictions from stochastic actor-oriented models can be obtained by aggregating over a set of simulated networks rather than a sequence of observations. By combining all information on dyadic predictions into one layout, an impression of the predicted global structure is conveyed. Our approach does assume that simulation results exhibit a certain degree of similarity in terms of their global structure, although this is not directly ensured by the model's inferential goal of identifying local processes. We argued, however, that conditioned on the initial structure in the first observation and because of dependencies of dynamics on previous changes, local processes often do yield globally similar patterns. This may no longer hold if observations differ in, say, more than half of the dyads, but this would also be a problem for the models themselves.

Our second diagnostic method focuses on the correspondence of predictions and observations on the actor level by visualizing discrepancies between simulated and observed local statistics. The aim is to either evaluate homogeneity assumptions or to detect outliers or groups of actors with deviant behavior in order to analyze their shared characteristics. The latter may indirectly suggest model alterations, and one particular use-case is to experiment with additional effects not yet included in a model. Note that, because of the Markov property of SAOMs, both diagnostic methods extend to more than two observations.

The visualization methods presented here are to facilitate visual exploration of data and models. Clearly, experimentation and further refinement will be necessary to better assess their utility. The inherent flexibility of the basic layout engine, stress minimization, may carry over to related visualization problems, and we plan to look at exponential-family random graph models (ERGM; see, e.g., Snijders et al., 2006) in particular.

Most of the methods described here are already available within visone,[4] a software tool for network analysis and visualization, and more will be added.

[4] http://www.visone.info/.

References

Bender-deMoll, S., McFarland, D., 2006. The art and science of dynamic network visualization. Journal of Social Structure 7 (2).

Bertin, J., 1983. Semiology of Graphics: Diagrams, Networks, Maps. University of Wisconsin Press.

Böhringer, K.-F., Paulisch, F.N., 1990. Using constraints to achieve stability in automatic graph layout algorithms. In: Proceedings of the ACM Human Factors in Computing Systems Conference (CHI'90). ACM, The Association for Computing Machinery, pp. 43–51.

Brandes, U., 2001. Drawing on physical analogies. In: Kaufmann, M., Wagner, D. (Eds.), Drawing Graphs: Methods and Models. Vol. 2025 of Lecture Notes in Computer Science. Springer-Verlag, pp. 71–86.

Brandes, U., Corman, S.R., 2003. Visual unrolling of network evolution and the analysis of dynamic discourse. Information Visualization 2 (1), 40–50.

Brandes, U., Kenis, P., Raab, J., 2006. Explanation through network visualization. Methodology 2 (1), 16–23, Spanish translation in REDES 9(6), 2005.

Brandes, U., Pich, C., 2007. Eigensolver methods for progressive multidimensional scaling of large data. In: Kaufmann, M., Wagner, D. (Eds.), Proceedings of the 14th International Symposium on Graph Drawing (GD'06). Vol. 4372 of Lecture Notes in Computer Science. Springer-Verlag, pp. 42–53.

Brandes, U., Pich, C., 2009. An experimental study on distance-based graph drawing. In: Proceedings of the 16th International Symposium on Graph Drawing (GD'08). Vol. 5417 of Lecture Notes in Computer Science. Springer-Verlag, pp. 218–229.

Brandes, U., Pich, C., 2011. More flexible radial layout. Journal of Graph Algorithms and Applications 15 (1), 157–173.

Brandes, U., Wagner, D., 1997. A Bayesian paradigm for dynamic graph layout. In: Di Battista, G. (Ed.), Proceedings of the 5th International Symposium on Graph Drawing (GD'97). Vol. 1353 of Lecture Notes in Computer Science. Springer-Verlag, pp. 236–247.

Branke, J., 2001. Dynamic graph drawing. In: Kaufmann and Wagner (2001), pp. 228–246.

Bridgeman, S.S., Tamassia, R., 2000. Difference metrics for interactive orthogonal graph drawing algorithms. Journal of Graph Algorithms and Applications 4 (3), 47–74.

Cleveland, W.S., McGill, R., 1984. Graphical perception: theory, experimentation, and application to the development of graphical methods. Journal of the American Statistical Association 79 (September (387)), 531–554.

Cox, T.F., Cox, M.A.A., 2001. Multidimensional Scaling, 2nd edition. CRC/Chapman and Hall.

de Leeuw, J., 1977. Applications of convex analysis to multidimensional scaling. In: Barra, J.R., Brodeau, F., Romier, G., van Cutsem, B. (Eds.), Recent Developments in Statistics. North-Holland, Amsterdam, pp. 133–145.

Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G., 1999. Graph Drawing: Algorithms for the Visualization of Graphs. Prentice Hall.

Diehl, S., Görg, C., 2002. Graphs, they are changing. In: Proceedings of the 10th International Symposium on Graph Drawing (GD'02). Vol. 2528 of Lecture Notes in Computer Science. Springer-Verlag, pp. 23–30.

Dwyer, T., Gallagher, D.R., 2004. Visualising changes in fund manager holdings in two and a half-dimensions. Information Visualization 3 (4), 227–244.

Dwyer, T., Hong, S.-H., Koschützki, D., Schreiber, F., Xu, K., 2006. Visual analysis of network centralities. In: Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation (APVis'06), pp. 189–197.

Dwyer, T., Marriott, K., Wybrow, M., 2008. Topology preserving constrained graph layout. In: Proceedings of the 16th International Symposium on Graph Drawing (GD'08). Vol. 5417 of Lecture Notes in Computer Science. Springer-Verlag, pp. 230–241.

Eades, P., 1984. A heuristic for graph drawing. Congressus Numerantium 42, 149–160.

Erten, C., Kobourov, S., Le, V., Navabi, A., 2004. Simultaneous graph drawing: layout algorithms and visualization schemes. In: Proceedings of the 11th International Symposium on Graph Drawing (GD'03). Vol. 2912 of Lecture Notes in Computer Science. Springer-Verlag, pp. 437–449.

Freeman, L.C., 2000. Visualizing social networks. Journal of Social Structure 1 (1). http://www.heinz.cmu.edu/project/INSNA/joss/vsn.html.

Friedrich, C., Eades, P., 2002. Graph drawing in motion. Journal of Graph Algorithms and Applications 6 (3), 353–370.

Friedrich, C., Houle, M.E., 2002. Graph drawing in motion II. In: Proceedings of the 9 International Symposium on Graph Drawing (GD'01). Vol. 2265 of Lecture Notes in Computer Science. Springer-Verlag, pp. 122–125.

Frishman, Y., Tal, A., 2008. Online dynamic graph drawing. IEEE Transactions on Visualization and Computer Graphics 14 (4), 727–740.

Fruchterman, T.M., Reingold, E.M., 1991. Graph-drawing by force-directed placement. Software—Practice and Experience 21 (11), 1129–1164.

Gajer, P., Goodrich, M.T., Kobourov, S.G., 2004. A multi-dimensional approach to force-directed layouts of large graphs. Computational Geometry 29 (1), 3–18.

Gansner, E., Hu, Y., 2008. Efficient node overlap removal using a proximity stress model. In: Proceedings of the 16th International Symposium on Graph Drawing (GD'08). Vol. 5417 of Lecture Notes in Computer Science. Springer-Verlag, pp. 206–217.

Gansner, E., Koren, Y., North, S., 2004. Graph drawing by stress majorization. In: Proceedings of the 12th International Symposium on Graph Drawing (GD'04). Vol. 3383 of Lecture Notes in Computer Science. Springer-Verlag, pp. 239–250.

Geipel, M.M., 2007. Self-organization applied to dynamic network layout. International Journal of Modern Physics C 18 (10), 1537–1549.

Groh, G., Hanstein, H., Wörndl, W., 2009. Interactively visualizing dynamic social networks with DySoN. In: Proceedings of the IUI'09 Workshop on Visual Interfaces to the Social and Semantic Web (VISSW'09). Vol. 443. CEUR Workshop Proceedings.

Hachul, S., Jünger, M., 2004. Drawing large graphs with a potential-field-based multilevel algorithm. In: Proceedings of the 12th International Symposium on Graph Drawing (GD'04). Vol. 3383 of Lecture Notes in Computer Science. Springer-Verlag, pp. 285–295.

Herman, I., Melançon, G., Marshall, M.S., 2000. Graph visualization and navigation in information visualization: a survey. IEEE Transactions on Visualization and Computer Graphics 6 (1), 24–43.

Huang, M.L., Eades, P., Wang, J., 1998. On-line animated visualization of huge graphs using a modified spring algorithm. Journal of Visual Languages and Computing 9 (6), 623–645.

Hunter, D.R., Goodreau, S.M., Handcock, M.S., 2008. Goodness of fit of social network models. Journal of the American Statistical Association 103 (481), 248–258.

Kamada, T., Kawai, S., 1988. A simple method for computing general positions in displaying three-dimensional objects. Computer Vision, Graphics and Image Processing 41, 43–56.

Kaufmann, M., Wagner, D. (Eds.), 2001. Drawing Graphs: Methods and Models. Vol. 2025 of Lecture Notes in Computer Science. Springer.

Klovdahl, A.S., 1981. A note on images of networks. Social Networks 3, 197–214.

Koskinen, J.H., Snijders, T.A.B., 2007. Bayesian inference for dynamic social network data. Journal of Statistical Planning and Inference 137 (12), 3930–3938.

Kruskal, J., Seery, J., 1980. Designing network diagrams. In: Proceedings of the First General Conference on Social Graphics, pp. 22–50.

Kruskal, J.B., Wish, M., 1978. Multidimensional Scaling. Vol. 07–011 of Sage University Paper series on Quantitative Applications in the Social Sciences. Sage Publications.

Lyons, K.A., Meijer, H., Rappaport, D., 1998. Algorithms for cluster busting in anchored graph drawing. Journal of Graph Algorithms and Applications 2 (1), 1–24.

Misue, K., Eades, P., Lai, W., Sugiyama, K., 1995. Layout adjustment and the mental map. Journal on Visual Languages and Computing 6 (2), 183–210.

Moody, J., McFarland, D., Bender-deMoll, S., 2005. Dynamic Network Visualization. American Journal of Sociology 110 (4), 1206–1241.

Moreno, J.L., 1953. Who Shall Survive: Foundations of Sociometry, Group Psychotherapy, and Sociodrama. Beacon House.

Nesbitt, K.V., Friedrich, C., 2002. Applying gestalt principles to animated visualizations of network data. In: Proceedings of the 6th International Conference on Information Visualisation (IV'02). IEEE Press, pp. 337–343.

North, S.C., 1996. Incremental layout with DynaDag. In: Brandenburg, F.J. (Ed.), Proceedings of the 3rd International Symposium on Graph Drawing (GD'95). Vol. 1027 of Lecture Notes in Computer Science. Springer, pp. 409–418.

Purchase, H.C., Cohen, R.F., James, M., 1997. An experimental study of the basis for graph drawing algorithms. ACM Journal of Experimental Algorithmics 2 (4).

Purchase, H.C., Samra, A., 2008. Extremes are better: investigating mental map preservation in dynamic graphs. In: Proceedings of the 5th International Conference on Diagrammatic Representation and Inference (Diagrams 2008). Vol. 5223 of Lecture Notes in Computer Science. Springer-Verlag, pp. 60–73.

Ripley, R., Snijders, T.A., 2011. Manual for Siena Version 4.0 (Version February 6, 2011). University of Oxford, Department of Statistics; Nuffield College, Oxford, http://www.stats.ox.ac.uk/siena/.

Saffrey, P., Purchase, H., 2008. The "mental map" versus "static aesthetic" compromise in dynamic graphs: a user study. In: Proceedings of the 9th Australasian User Interface Conference (AUIC 2008), pp. 85–93.

Schweinberger, M. (2011). Statistical modelling of network panel data: goodness of fit. British Journal of Mathematical and Statistical Psychology, 64, doi:10.1111/j.2044-8317.2011.02022.x.

Sibson, R., 1978. Studies in the robustness of multidimensional scaling: Procrustes statistics. Journal of the Royal Statistical Society. Series B (Methodological) 40 (2), 234–238.

Snijders, T.A.B., 2001. The statistical evaluation of social network dynamics. Sociological Methodology 31, 361–395.

Snijders, T.A.B., 2005. Models for longitudinal network data. In: Carrington, P.J., Scott, J., Wasserman, S. (Eds.), Models and Methods in Social Network Analysis. Cambridge University Press, New York, pp. 215–247.

Snijders, T.A.B., Koskinen, J., Schweinberger, M., 2010a. Maximum likelihood estimation for social network dynamics. Annals of Applied Statistics 4 (2), 567–588.

Snijders, T.A.B., Pattison, P.E., Robins, G.L., Handcock, M.S., 2006. New specification for exponential random graph models. Sociological Methodology 36, 99–153.

Snijders, T.A.B., Steglich, C.E.G., Schweinberger, M., 2007. Modeling the co-evolution of networks and behavior. In: van Montfort, K., Oud, H., Satorra, A. (Eds.), Longitudinal Models in Behavioral and Related Sciences. Lawrence Erlenbaum, pp. 41–71.

Snijders, T.A.B., van de Bunt, G.G., Steglich, C.E.G., 2010b. Introduction to stochastic actor-based models for network dynamics. Social Networks 32 (1), 44–60.

Steglich, C.E.G., Snijders, T.A.B., Pearson, M., 2010. Dynamic networks and behavior: separating selection from influence. Sociological Methodology 40, 329–393.

Torgerson, W.S., 1952. Multidimensional scaling. I. Theory and method. Psychometrika 17, 401–419.

Walshaw, C., 2001. A multilevel algorithm for force-directed graph drawing. In: Marks, J. (Ed.), Proceedings of the 8th International Symposium on Graph Drawing (GD 2000). Vol. 1984 of Lecture Notes in Computer Science. Springer, pp. 171–182.