Versinus: an Animated Visualization Method for Evolving Networks⋆

Renato Fabbri[1][0000−0002−9699−629X] and Maria Cristina Ferreira de Oliveira[1][0000−0002−4729−5104]

University of São Paulo, São Carlos SP, BR
renato.fabbri@gmail.com, cristina@icmc.usp.br
http://conteudo.icmc.usp.br/pessoas/cristina/

Abstract. Most real world networks change over time with new nodes and links or their removal, or even with metadata modification. Increasing interest in modeling evolving networks has been reported and a number of methods have been proposed to visualize them. This article presents a novel visualization approach for (time-)evolving (or longitudinal) networks, specially useful for the observation of (in)stability of basic topological characteristics. The method has received a few software implementations and has been useful in research and publication, which motivated the writing of this present document. Named Versinus, the visualization consists essentially in placing the most connected nodes (hubs and intermediary) along a sine curve, and the peripheral nodes nodes along a separate segment, usually a line. Thus its name from Latin *versus* (line) and *sinus* (sine). The method has provided insights into network properties and for network visualizations, which are described here after a formal presentation of the visualization and yielded software.

Keywords: Network visualization · Evolving networks · Animated visualization · Complex networks · Data visualization.

1 Introduction

The visualization of evolving networks poses challenges related to data size and complexity which have merited various contributions [1, 8, 3]. In fact, the meaningful visualization of time-evolving networks is a well-known problem in information visualization (*infovis*) and dynamic network analysis (*DNA*). In this work, we present a novel approach in DNA infovis that consists in distributing the nodes along a fixed layout that consists in three segments for three sections (or partitions) of a scale-free network, namely for the hub, intermediary and peripheral nodes [2]. The development of the network is expressed by changes in node height, width, color, format, and by changes in link disposition, thickness and color. Also, auxiliary global and local information are be displayed in order to enable a sound analysis of the features of interest. The method has been called

Versinus since its early usages in 2013, because the layout stabilized, after trying many figures, in a sine period and a line, which in Latin is *sinus* (sine) and *versus* (line). Versinus has received four software implementations, two of them written in Python scripts that yield animations through stop-motion. The other two are written in JavaScript, one of them is an exploitation for static networks, the other is an ongoing Visual Analytics interface which extends the description in this document for 3D with additional analysis techniques such as principal component analysis (PCA). The writing of this document was motivated by its usefulness in scientific research [2, **?**] and the benefit of a careful mathematical and technological description of the procedures involved in applying the method to real networks.

The article is organized as follows. Next subsections hold remarks about related work and vocabulary. Section 2 describes the method in detail and sufficient formalism. Section 3 is depicts the software implementations of Versinus. Section 4 is dedicated to revealing further insights obtained through the use of the method. Finally, Section 5 convene concluding and further work commentaries.

1.1   Related work

Although animation is often criticized in the data visualization canonical literature [6, 10], advantages have also been reported [7, 4] e.g. making visuallizations more eye catching and due to the fact that we are accustumed to moving through the world and to moving objects. Evolving network visualization has been tackled both through animated and static (often timeline-based) visualizations [11, 5, 9], focusing on various global and local network features. The contribution reported in this document is singular in consisting basically of a network layout which is fit for scale-free characteristics, together with some visualization artifacts dependent of the implementation. The collection of procedures and conceptualizations involved in the achieving the final visualization were also considered worth documenting as were not found elsewhere in the literature, designed simple and genuine, and found of scientific relevance [2].

1.2   Nomenclature remarks

Evolving networks are also called e.g. logitudinal networks, time-evolving networks, dynamic networks. We chose the term "evolving network" herein because it is simple, is semantically reasonable, and has been very much in use. The other main issue in current vocabulary is in the phrase "section of a network" (e.g. of the hubs). In most cases, the sections are also a partitions (i.e. they are non-overlaping sets whose union is the complete superset), but it depends of the criteria for obtaining the sectors [2], thus the use of the more general vocable *section*. Moreover, network science is very multidisciplinary and context-dependent, and flourished from the ill-defined area of complex systems, favoring polysemy and synonymy. We strove to avoid such issues because they are usually not convenient in scientific literature.

2   A description of the method

Versinus is essentially obtained through placing the nodes in segments with respect to their overall topological features, and then representing instantaneous topological features though gliph and arrow attributes. The visualization may be divided thus in three stages, described in the following subsections.

2.1   Node placement

First, a network is constructed with all the activity of interest, i.e. all the transactions, such as all the email messages considered or all the links in a data file. The nodes are then ordered either by degree or strength (or in- or out- degree or strength), from greatest to smallest values. Also from such global network, the hubs, intermediary and peripheral sectors are obtained. Two methods have been proven useful for obtaining such sectors: 1) assuming fixed fractions of hubs (e.g. 5%), of intermediary (e.g. 15%) and peripheral nodes (e.g. 80%); 2) by comparison of the network against the Erdös-Rényi model, a not trivial procedure carefully described in [2].

With such structures at hand, the nodes are placed along three segments. The hubs are placed along the first half of a sine period. The intermediary nodes are placed along the second half of the sine period. The peripheral nodes are placed along an indepentent line. Such placement is depicted in Figure **??** with the basic parameters:

- the amplitude α of the sine oscillation.
- The displacement Δ of the sine in the vertical axis.
- the horizontal margin μ, i.e. the distance from the left border to the first hubs, assumed the same as the distance from the right border to last intermediary node.
- The positions (x_0, y_0) and (x_1, y_1), i.e. the endpoints of the segment for the peripheral sector.

Some crafty notes: fine-tunning of the size of each sector (number of nodes) may come handy due to the density of nodes in each segment and due to their activity. Also, the node positioned at the middle of the sine period may be a hub or an intermediary node, and this choice often impacts final visualization clarity more than predicted by intuition. Finally, avoiding clutter with the hubs motivated both the positioning of the peripheral sector in the upper-right corner and the placement of the most active peripheral nodes next to the intermediary sector, i.e. the ordering along the peripheral line.

2.2   Drawing nodes, links and auxiliary information

For visualizing the evolving network, further concepts and quantities need to be defined. A *snapshot S* consists of a number σ time-contiguous transactions, e.g. of 100 email messages, starting at an arbitrary message. The separation η
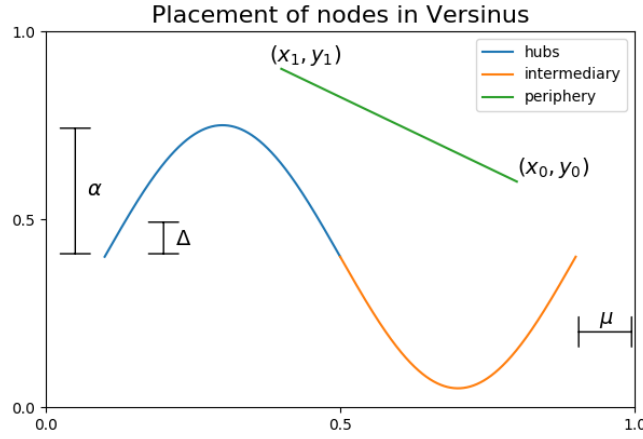
Fig. 1. The underlying layout of Versinus. Most connected nodes are positioned from left to right on the sine segment (for the hubs and intermediary sectors), and from right to left on the upper line (for the periphery sector). The parameters are: amplitude α of the oscillation; vertical displacement Δ of the oscillation; horizontal margin μ; and line endpoints (x_0, y_0) and (x_1, y_1).

between each snapshot is the number of transactions skipped from one snapshot to the next. In summary, the evolving network is visualized through a sliding window of σ transactions, updated each η transactions. To achieve the final animation, one need also to define a transaction rate ω. Thus, these parameters are specified:

- the window σ captured at each snapshot, expressed in transactions.
- The step size η between each snapshot, expressed in transactions.
- The transaction rate ω, expressed in transactions per second.

The sequence of snapshots may be expressed as $\{S_i^{\sigma,\eta}\}_0^L$, where L is the smallest integer such that $L \times \eta + \sigma > T$ and T is the total number of transactions. Notice that the last snapshot will not necessarily have σ transactions, but may have less than σ. Also, the rate ω is only applied when rendering the animation. E.g. if each snapshot S_i yields an image, the stop motion animation will use ω/η images per second to achieve the desired rate of ω transactions per second.

Finally, auxiliary information may be given by a persistent legend that displays essential information such as σ, η, ω and T, and keeps track of current snapshot along e.g. the number of nodes and edges of the snapshot. Also, node-specific information may be displayed by blinking the information onver the nodes periodically or displaying them with a vertical displacement. Figure **??** examplifies two flavours of Versinus settings.

2.3   Observation of the evolving network

3   Software implementations

4   Insights gathered

5   Conclusions and further work

References

1. Albert, R., Barabási, A.L.: Topology of evolving networks: local events and universality. Physical review letters **85**(24), 5234 (2000)
2. Fabbri, R., Fabbri, R., Antunes, D.C., Pisani, M.M., de Oliveira Junior, O.N.: Temporal stability in human interaction networks. Physica A: Statistical Mechanics and its Applications **486**, 92–105 (2017)
3. Fenu, C., Higham, D.J.: Block matrix formulations for evolving networks. SIAM Journal on Matrix Analysis and Applications **38**(2), 343–360 (2017)
4. Fisher, D.: Animation for visualization: opportunities and drawbacks. Ch **19**, 329–352 (2010)
5. Ma, C., Kenyon, R.V., Forbes, A.G., Berger-Wolf, T., Slater, B.J., Llano, D.A.: Visualizing dynamic brain networks using an animated dual-representation. In: Proceedings of the Eurographics conference on visualization (EuroVis). pp. 73–77 (2015)
6. Munzner, T.: Visualization analysis and design. AK Peters/CRC Press (2014)
7. Nakakoji, K., Takashima, A., Yamamoto, Y.: Cognitive effects of animated visualization in exploratory visual data analysis. In: Proceedings Fifth International Conference on Information Visualisation. pp. 77–84. IEEE (2001)
8. Peel, L., Clauset, A.: Detecting change points in the large-scale structure of evolving networks. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
9. Schneider, B., Acevedo, C., Buchmüller, J., Fischer, F., Keim, D.A.: Visual analytics for inspecting the evolution of a graph over time: pattern discovery in a communication network. In: 2015 IEEE Conference on Visual Analytics Science and Technology (VAST). pp. 169–170. IEEE (2015)
10. Ware, C.: Information visualization: perception for design. Elsevier (2012)
11. Wu, Y., Pitipornvivat, N., Zhao, J., Yang, S., Huang, G., Qu, H.: egoslider: Visual analysis of egocentric network evolution. IEEE transactions on visualization and computer graphics **22**(1), 260–269 (2016)