Software for Agent-based Network Simulation and Visualization

Patrick Shepherd, Isaac Batts, Judy Goldsmith, Emory Hufbauer, Mia Weaver, and Angela Zhang
University of Kentucky
patrick.shepherd@uky.edu, IsaacBatts@uky.edu, goldsmit@cs.uky.edu, emory.hufbauer@uky.edu, Mia.Weaver@uky.edu, angelalzhang@gmail.com

Abstract

We present a network software suite that can model contagions or opinion manipulation in social networks, that combines features from the standard packages and extends them to allow complex, interacting, dynamic topologies, and dynamic heterogeneous agent types, with individual interaction policies. The framework allows for the easy implementation of new agent types, and provides flexible visualization tools to elucidate network behavior over time.

Introduction

At its core, our software is for network simulations — specifically, processes wherein the topology of a network and/or a k-dimensional attribute space associated with it evolve over time (e.g., epidemics, political opinions, etc.). Although designed specifically for opinion revelation and updates in online social networks, the software is designed to be a general network simulator that can easily adapt to other settings. Our demonstration illustrates two applications for our software, and shows some of the types of investigation it can support. In the next section, we describe the technical details of our implementation; next, we discuss similar software packages and highlight both their capabilities and their shortcomings; finally, we situate our own work against the backdrop of the previous section, highlighting both similarities and differences.

The Software

Like most social network software, ours begins with a representation of a graph $G = \langle V, E \rangle$, where V is the set of entities (e.g. users on a social network), and E is the set of pairs $(i, j) : i, j \in V$ representing connections between them. Edges in our software can be weighted or unweighted, depending on the context. For example, when simulating opinion change over time, edges may represent personal relationships with weights corresponding to more or less valuable relationships, such as family as opposed to loose acquaintances. Additionally, the software accounts for either directed or undirected edges, allowing for influence to be either symmetric or not.

We have built our platform using the well-known package NetworkX for the underlying structure and basic mechanisms such as creating and destroying links, and gathering neighbors of a node.[1] Since a NetworkX data structure underlies our own, any facilities available to those data structures is also available natively within our platform.

Our framework allows for easy user-defined topology update models. Since we employ an agent-based approach, individual nodes can decide, for each of their neighbors, whether to sever their connection based on some criteria (see below). We also provide a replaceable mechanism for creating new links. By default, the software probabilistically adds new links to create new triadic closures at each discrete time step. However, our software allows for the quick substitution of another edge creation routine to fit the context. Edge creation and deletion can be made either deterministic or probabilistic with the alteration of one parameter each to allow either predictability or dynamism in simulations.

In addition to topological evolution, our platform facilitates changing multidimensional attribute spaces associated with the network. For instance, our main work using the software involves opinion revelation and revision over time in a social network. First, attribute spaces can have as many dimensions as desired — in terms of opinion experiments, we allow each agent (node) to have an opinion on each of k topics. The dimensions are binary by default, but can be modified to consider continuous spaces. We also provide the ability for each agent to decide when and if each of its neighbors is allowed to know its state in each of those dimensions.

Our software's key feature is the ability to define *archetypes*. One archetype can be assigned to each node during a simulation, which influences the node's behavior. For example, one archetype may gravitate socially towards others who are similar in attribute space (homophily), while another tends to move toward those who are different instead (heterophily) (Albi, Pareschi, and Zanella 2014; Chen, Tang, and Sun 2018; Flache 2018; Friedkin et al. 2016; Motsch and Tadmor 2014; Rossetti et al. 2017; Toscani, Tosin, and Zanella 2018). Agent-specific attribute update schemes can easily be incorporated as well. Using an opinion space on the network, this is analogous to each agent changing their opinions based on current visible evidence; some agents will

[1] networkx.org

change their opinion to conform to the majority, while others change in the opposite direction to avoid conforming (Do Yi et al. 2013; Kurmyshev, Juárez, and González-Silva 2011; Sîrbu et al. 2013). In an epidemic model, this could correspond to different types of entities: businesses, hospitals, and individuals, for example. Our framework also provides an interface for reinforcement learning, allowing for easy definition of archetype reward functions and coded policies, as well as feature spaces for state representations.

Finally, we provide visualization capabilities for simulations. The ability to watch a network evolve over time as the nodes make individual decisions helps elucidate phenomena occurring that are not apparent immediately, if at all, in raw data. We provide an easily extensible API that allows individual or classes of nodes to be color-coded and/or resized at each step based on user-defined criteria. In the first example of our demonstration, we emulate a classic SIR model for epidemics and color-code nodes based on their status as Susceptible, Infected, or Removed, and provide white or black borders for nodes depending on whether or not they are wearing face masks. In its current form, our software can handle dynamic visualizations with thousands of nodes.

Related Work

The most similar work to our own is NetSim (Stadtfeld 2015). This R package provides a flexible and extensible framework in which the user can define arbitrary timing schemes, attribute spaces, and behavior models for the network. The behavior model is achieved by defining a rule to compute the state of the network at time $t + 1$ given the state at time t. The software also provides basic visualization tools to accompany these simulations. One of our main goals was to not only meet and exceed this functionality, but to flatten the learning curve associated with its use. Another tool allowing for dynamic network attribute updates is Tulip (Auber et al. 2017). Similar to NetSim, Tulip provides the underlying software framework for implementing a dynamic attribute update scheme, but it involves a heavy programming load. It does, however, provide the functionality to implement an organically dynamic network, and produce highly customized visualizations to accompany it.

To our knowledge, almost all other software is focused on network analysis, or strictly on visualization. Here we briefly discuss some work from each category.

Network Analysis: Bogdanov *et al.* (Bogdanov, Mongiovì, and Singh 2011) investigate methods of determining subgraphs of a network with high importance over time. The authors develop algorithms for discovering such subgraphs, and provide experimental testing on, e.g., transport and communications networks. Lee *et al.* (Lee, Xue, and Hunter 2020) also present work on discovering groups within evolving networks with certain properties such as stability. Both studies make mention of their approaches' scalability. Meerkat (Jiyang et al. 2010) is another tool that focuses not only on community detection using novel algorithms for temporal networks, but heavily on visualization as well. IncNSA (Su et al. 2020) is also focused on community detection within a dynamic network.

Maduaco *et al.* (Maduako, Wachowicz, and Hanson 2019) present a novel framework that is specifically designed to address the computational overhead involved in storing and visualizing large and rapidly changing networks. Their work provides mechanisms for statistical analysis on the network. Ho and Xing (Ho and Xing 2014) investigate the impacts of different roles for nodes in a time-evolving network, mostly with respect to topology. Ohsaka *et al.* investigate influence evolution in a dynamic network using a novel data structure designed for efficiently handling dynamic network updates. Another well-known tool for social network analysis is UCINET, although it is meant for use on static networks.

Visualization: Commetrix (Trier 2008) is a network visualization tool designed with dynamic networks in mind. It offers tools for 2D and 3D plotting, versatile controls for aspects like node coloring, and facilities for searching and sorting. Commetrix can access a wide range of data, making it applicable to virtually any domain. Cuttlefish (Geipel 2007) is a technology for visualizing complex and dynamic networks from a broad range of sectors. SocNetV is a program that combines expansive visualization tools with a full set of network analysis features, but only allows for static networks. Similarly, Gephi (Bastian, Heymann, and Jacomy 2009) and GraphViz (Ellson et al. 2003) are two software packages that include extensive functionality for graph visualization on static networks.

Significance

Our software provides several advantages. First, it is a necessary addition to the set of software for modeling network evolution through agent-based decision making. Prior tools are scarce and require considerable effort and study before they can be implemented effectively. We designed our framework to reduce barriers to use, and to allow the user to design flexible agent-driven network evolution schemes.

The second advantage is the set of tools available for designing such schemes. With the ability to define archetypes, the user can control many aspects of the experience of each agent: they can be given different sets of potential actions, they can process information differently from others, they can alter the topology of the network around them according to their preferences, and they can react to external influences differently. Not only does this act as the engine that drives network evolution, but it also provides all the necessary mechanisms to serve as a platform for reinforcement learning. Finally, our software combines a scalable set of visualization features. This aspect of the project is a continuing effort, and moving forward we will equip the software with greater capabilities, as well as providing the user with more fine-grained control over the animations.

As networks of all kinds shape our lives to an ever-increasing degree, the need for more flexible and powerful network analysis tools will only grow. While the ability to run simulations on a simple set of rules — or analyze a network's evolution after the fact — are necessary and have an enormous breadth of application, novel conditions brought about by social networks and the rising Internet of Things, among others, have necessitated more thorough investigation into deliberate actors within networks.

References

Albi, G.; Pareschi, L.; and Zanella, M. 2014. Boltzmann-type control of opinion consensus through leaders. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 372(2028): 20140138.

Auber, D.; Archambault, D.; Bourqui, R.; Delest, M.; Dubois, J.; Lambert, A.; Mary, P.; Mathiaut, M.; Melançon, G.; Pinaud, B.; et al. 2017. TULIP 5.

Bastian, M.; Heymann, S.; and Jacomy, M. 2009. Gephi: An Open Source Software for Exploring and Manipulating Networks. http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154.

Bogdanov, P.; Mongiovì, M.; and Singh, A. K. 2011. Mining heavy subgraphs in time-evolving networks. In *2011 IEEE 11th International Conference on Data Mining*, 81–90. IEEE.

Chen, X.; Tang, J.; and Sun, Y. 2018. Modeling Personalized Dynamics of Social Network and Opinion at Individual Level. *arXiv preprint arXiv:1811.02791* .

Do Yi, S.; Baek, S. K.; Zhu, C.-P.; and Kim, B. J. 2013. Phase transition in a coevolving network of conformist and contrarian voters. *Physical Review E* 87(1): 012806.

Ellson, J.; Gansner, E. R.; Koutsofios, E.; North, S. C.; and Woodhull, G. 2003. Graphviz and dynagraph – static and dynamic graph drawing tools. In *GRAPH DRAWING SOFTWARE*, 127–148. Springer-Verlag. URL https://graphviz.org/.

Flache, A. 2018. Between monoculture and cultural polarization: agent-based models of the interplay of social influence and cultural diversity. *Journal of Archaeological Method and Theory* 25(4): 996–1023.

Friedkin, N. E.; Proskurnikov, A. V.; Tempo, R.; and Parsegov, S. E. 2016. Network science on belief system dynamics under logic constraints. *Science* 354(6310): 321–326.

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

Ho, Q.; and Xing, E. P. 2014. Analyzing time-evolving networks using an evolving cluster mixed membership blockmodel. *Handbook of Mixed Membership Models and Their Applications* 489–525.

Jiyang, C.; Fagnan, J.; Goebel, R.; Rabbany, R.; Sangi, F.; Takaffoli, M.; and Zaïane, O. 2010. Meerkat: Community mining with dynamic social networks. In *Proceeding of the IEEE International Conference*.

Kurmyshev, E.; Juárez, H. A.; and González-Silva, R. A. 2011. Dynamics of bounded confidence opinion in heterogeneous social networks: Concord against partial antagonism. *Physica A: Statistical Mechanics and its Applications* 390(16): 2945–2955.

Lee, K. H.; Xue, L.; and Hunter, D. R. 2020. Model-based clustering of time-evolving networks through temporal exponential-family random graph models. *Journal of Multivariate Analysis* 175: 104540.

Maduako, I.; Wachowicz, M.; and Hanson, T. 2019. STVG: an evolutionary graph framework for analyzing fast-evolving networks. *Journal of Big Data* 6(1): 55.

Motsch, S.; and Tadmor, E. 2014. Heterophilious dynamics enhances consensus. *SIAM Review* 56(4): 577–621.

Rossetti, G.; Milli, L.; Rinzivillo, S.; Sirbu, A.; Pedreschi, D.; and Giannotti, F. 2017. Ndlib: Studying network diffusion dynamics. In *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 155–164. IEEE.

Sîrbu, A.; Loreto, V.; Servedio, V. D.; and Tria, F. 2013. Opinion dynamics with disagreement and modulated information. *Journal of Statistical Physics* 151(1-2): 218–237.

Stadtfeld, C. 2015. NetSim: A social networks simulation tool in R. *R package vignette http://www. social-networks. ethz. ch/research/research-projects. html* .

Su, X.; Cheng, J.; Yang, H.; Leng, M.; Zhang, W.; Chen, X.; et al. 2020. IncNSA: Detecting communities incrementally from time-evolving networks based on node similarity. *International Journal of Modern Physics C (IJMPC)* 31(07): 1–19.

Toscani, G.; Tosin, A.; and Zanella, M. 2018. Opinion modeling on social media and marketing aspects. *Physical Review E* 98(2): 022315.

Trier, M. 2008. Research note—towards dynamic visualization for understanding evolution of digital communication networks. *Information Systems Research* 19(3): 335–350.