DeepGraph: Multi-Cluster Interactive Visualization of Complex Networks in a Learned Representation Space

Yidan Sun
Information Sciences Institute
University of Southern California
Marina del Rey, California, United States 90292
Email: yidans@isi.edu

Mayank Kejriwal
Information Sciences Institute
University of Southern California
Marina del Rey, California, United States 90292
Email: kejriwal@isi.edu

***Abstract*—Visualizing complex networks with many thousands of nodes and interesting community structure remains a challenging problem. This paper introduces *DeepGraph*, an off-the-shelf package that takes a network (encoded as an edge-list) as input, and uses open-source packages to interactively visualize nodes in the network in a learned representation space on a web browser. At its core, DeepGraph is powered by established unsupervised node embedding and clustering algorithms, allowing it to operate in an end-to-end fashion without requiring technical expertise or algorithmic parameters. More advanced users can 'swap' out the algorithms in a plug-and-play fashion. DeepGraph is especially designed for non-technical sociologists and subject matter experts looking to explore the data and its community structure before formulating research questions and follow-up studies. We demonstrate the utility and generality of DeepGraph on real-world network datasets spanning domains from digital communication to social media.**

I. Background

Recent years have witnessed enormous progress in deep representation learning of complex networks across multiple domains. While such learned representations or 'embeddings' have been extensively used in downstream machine learning applications, there has been comparatively little work in using them to visualize large networks, as well as *community structures* within these networks, in an interactive and dynamic manner. As networks become larger and more complex, there is a growing demand for easy-to-use visualization tools that can serve myriad purposes [1].

Unfortunately, network visualization remains a difficult and understudied problem, compared to other advances in network science. While hundreds (if not thousands) of papers have been published on problems such as community detection and link prediction [2]–[4], only a handful of open-source tools[1] exist for visualizing networks in a manner that is easy, interactive and scalable.

At the same time, it is well recognized in the computational sciences that visualization can play a crucial role in exploring and understanding any complex system or dataset [5]–[7], of which networks remain a paradigmatic example. Visualization offers users a 'feel' for the phenomenon, a critical step in formulating interesting research questions. Considering the wide range of domains that network science today can be applied to, from understanding social network patterns to decoding biological pathways [8], [9], the importance of an intuitive visualization approach cannot be overstated, especially for subject matter experts who may not have the specific computational expertise required to program their own visualization infrastructure.

We introduce a new open-source package called *DeepGraph* to address the challenge of scale and ease-of-use in a unified framework. DeepGraph operates by taking an edge-list as input, and using an established and efficient deep learning-based 'node embedding' algorithm like DeepWalk [10] to embed nodes in a learned represen-

ASONAM '23, November 6-9, 2023, Kusadasi, Turkey

ACM ISBN 979-8-4007-0409-3/23/11…$15.00
https://doi.org/10.1145/3625007.3627515

[1]We point the interested reader to [1] for a survey of the few tools that do exist.

tation space that can be subsequently rendered on a web browser. Once node embeddings are learned, we use an unsupervised clustering algorithm to automatically discover structures within the network. It is well known that most networks contain such structures e.g., in large-enough social networks, the natural formation of communities and groups has been extensively studied [3], [8], [11], [12]. To ensure that the embeddings (which can often be in the tens, or even a hundred, dimensions) can be visualized in a 2D or 3D space, we use the t-distributed stochastic neighbor embedding (t-SNE) algorithm, which itself uses a form of deep learning to do dimensionality reduction on vectors, with the explicit goal of visualizing them in a topologically accurate manner (relative to the high-dimensional space).

Users can explore these 'reduced' embeddings on a web browser using both bounding boxes (which allows them to zoom in or out, depending on which aspects of the network they are interested in), and a search box that allows them to hone in on nodes of interest. DeepGraph also allows the user to color the nodes using either the output of an unsupervised clustering algorithm, or a ground-truth file containing cluster labels. No technical knowledge is required to run DeepGraph, allowing non-technical sociologists and subject matter experts to explore complex networks in an off-the-shelf fashion. We demonstrate the utility of DeepGraph on several real-world network datasets, including the email communication network dataset [13] and the Facebook ego-network dataset [14] from SNAP.

At its core, we incorporate several engineering innovations into DeepGraph to make it a highly practical system, one that users can rely upon to not only explore complex networks more intuitively, but to formulate productive research questions suggested by the exploration. DeepGraph does not implement new algorithms for embeddings, dimensionality reduction, clustering, or rendering, although the defaults for some of these can be swapped out by computational researchers innovating in these areas. Hence, one secondary use case of DeepGraph is to allow such researchers to visualize the outputs of their algorithms to understand how these outputs differ compared to more standard baselines (although we do not intend to showcase this baselining capability of DeepGraph in an actual demonstration).

While conventional network visualization methods typically present data in static formats, there is a trend toward embracing deep learning-based rendering techniques. Examples of good network visualization tools include Cytoscape.js [15] and Gephi [16], which also offer dynamic visualizations, enhanced interactivity, and real-time data manipulation capabilities. However, direct support for embeddings and representation learning is not typically supported in these tools, despite enormous progress in graph representation learning over the last decade [10], [17]–[21]. Similar progress has been seen in dimensionality reduction, although even today, the t-SNE algorithm remains among the most widely used for reducing learned representations to two or three dimensions to visualize them more effectively. DeepGraph aims to complement the network visualization tools noted above by using deep representation learning to interactively visualize nodes and community structure. In conjunction with those tools, especially for large-scale network data, DeepGraph can provide an additional valuable perspective on the network.

II. DeepGraph Workflow

Figure 1 demonstrates the workflow and system architecture (core components) in DeepGraph. Below, we detail the different components, starting from the input.

Input: As illustrated in the workflow, DeepGraph requires a network dataset in the form of an edge-list text file as input. Within this format, each line, such as 0 10, denotes an edge originating from node 0 and pointing to node 10. Users are free to provide their edge-list or choose from our sample datasets. An example includes a subset of email communications sourced from a distinguished European research institution [13].

To visualize the nodes, we learn deep representations (or real-valued vector embeddings) for the nodes using the DeepWalk algorithm [10]. DeepWalk, a well-established algorithm in network science for representation learning, derives node representations via random walks traversing the graph. It is highly efficient, and although it has been outperformed in some domains by more advanced algorithms like node2vec [17], it remains widely used because of its efficiency and its visual utility. Besides, even algorithms like node2vec ultimately rely on a similar intuition (random walks). We make DeepGraph open-source; hence, the more advanced user can always swap out the DeepWalk module for another embedding algorithm.

Using node embeddings in a lower-dimensional space than the adjacency matrix itself (which has the same number of dimensions as the number of nodes in the graph, in contrast with DeepWalk, where the dimensionality is an input parameter and rarely exceeds 100), has numerous advantages. First, the embeddings are known to be capable of robustly capturing the topological relationships and other structural features of the nodes

within the network. Despite their low dimensionality, they have proven to be information-dense in difficult machine learning applications like link prediction. This is likely both due to the non-linearity in the underlying neural network used to learn the representation, and the use of random walks, which are known to (stochastically) encode non-local network structures [10].

We also give users the option to input any embeddings of their own interest, such as document embeddings, as long as they adhere to a documented format also used by other open-source embedding datasets, such as GLoVE [22]. Regardless of whether the embeddings are generated by our system or introduced by the user, we refer to them as the 'original embeddings' at this stage.

Embedding Processing: After obtaining the node embeddings, our system employs K-means clustering to identify community structures in the network. By default, there's a predetermined number of clusters, but advanced users have the flexibility to specify their desired number of clusters. To better delineate clusters within the high-dimensional setting, the system subsequently performs linear transformations on the grouped embeddings with the dual objective of minimizing intra-group node distances and maximizing inter-group distances. The embeddings derived from this procedure are referred to as 'transformed embeddings.'

Interactive Visualization: Our system employs the t-Distributed Stochastic Neighbor Embedding (t-SNE) method for dimensionality reduction and visualization. We offer several visualization options, each highlighting different facets of the original input and transformed embeddings:

- **Original Embedding Visualization**: This visualization utilizes the 'original embeddings' directly obtained from DeepWalk and projects the embeddings to a lower-dimensional space using t-SNE, without implementing any clustering or transformations.
- **Transformed Embedding Visualization (Colored by Clusters)**: In this visualization, t-SNE is employed on the 'transformed embeddings'. The 'reduced' node embeddings are colored according to clustering outcomes from previous steps.
- **Transformed Embedding Visualization (Colored by Labels)**: Similar to the previous visualization, t-SNE is employed on the 'transformed embeddings', but the 'reduced' node embeddings are colored based on user-provided labels. Users have the option to upload a label file, where an entry like 1 12 indicates that node 1 is labeled as 12. Note: This visualization option is only available when a label file is provided; otherwise, it will not appear in the selection choices.

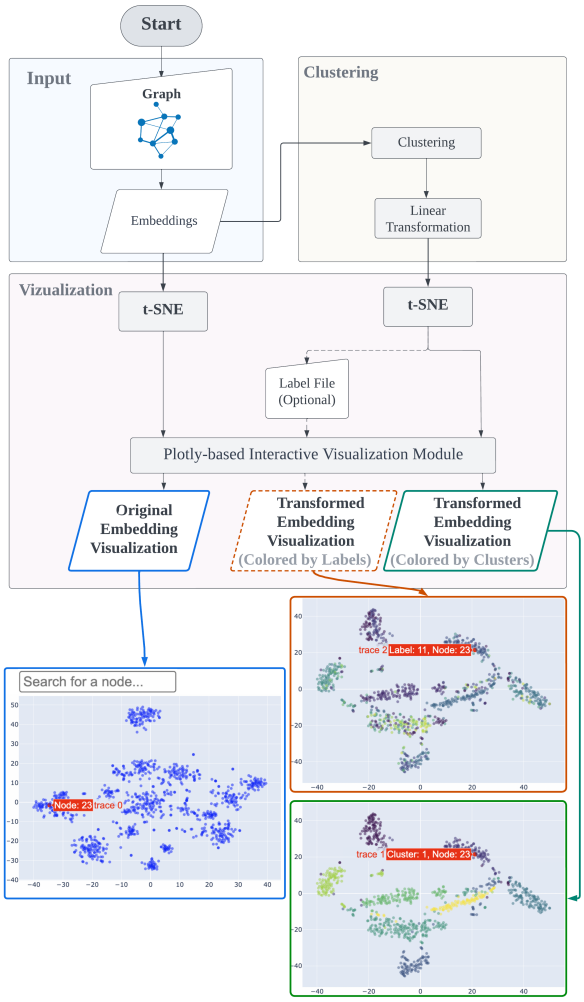

Fig. 1. A workflow-based demonstration of DeepGraph, starting from a user-provided network edge-file as an input. The three facets shown in the figure (two on the right, and one on the left) can all be rendered, and interacted with, on a browser. All algorithms (including for clustering and embeddings) have defaults that operate in an unsupervised fashion, not requiring any technical expertise from users.

The system also integrates a node search feature implemented using the Plotly Open Source Graphing Library for Python [23]. This feature allows users to pinpoint nodes, recognize affiliated clusters or communities, and compare node positions across three dynamic visualizations. In addition to the 2D graphics, a 3D version of the above three visualizations is also available. Both versions support zooming, panning, and hovering over nodes to display details like node ID and its affiliated

cluster or label. The final output is optimized for viewing and exploration directly within a web browser.

III. Anticipated User Experience

In the demonstration, we will use DeepGraph to visualize three different networks spanning domains, and one of which will be a social network with 'ground-truth' community structures. At least one network will contain 10,000+ nodes, a size that (although not large for network analysis) current visualization tools have trouble rendering efficiently. Users will be allowed to play with these visualizations in a browser, and to search for nodes of interest. Users will be able to draw bounding boxes and toggle between the three different facets described earlier. We will also assist interested users with setting up the package on their laptops by releasing it as open-source prior to the conference.

IV. Summary

Visualizing large networks and their inherent community structures continues to pose a persistent challenge. This challenge becomes even more pronounced as networks expand, particularly in fields such as social science and various other academic disciplines. Our proposed system DeepGraph provides a solution to this problem by utilizing deep learning-based node embedding algorithm, notably DeepWalk, to transform network nodes into embeddings. These embeddings are then clustered to identify inherent structures, and reduced to a visually interpretable 2D or 3D form using the t-SNE algorithm. Users can explore the visualizations on a web browser, with features like zooming, node search, and customizable color-coding based on clustering or provided labels. Tailored for user-friendliness, DeepGraph allows even non-technical experts to intuitively analyze complex networks.

References

[1] S. Majeed, M. Uzair, U. Qamar, and A. Farooq, "Social network analysis visualization tools: A comparative review," in *2020 IEEE 23rd International Multitopic Conference (INMIC)*. Bahawalpur, Pakistan: IEEE, 2020, pp. 1–6.

[2] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[3] M. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, p. 026113, 2004.

[4] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[5] A. Comai, "Decision-making support: The role of data visualization in analyzing complex systems," *World Futures Review*, vol. 6, no. 4, pp. 477–484, 2014.

[6] G. A. Pavlopoulos, D. Paez-Espino, N. C. Kyrpides, and I. Iliopoulos, "Empirical comparison of visualization tools for larger-scale network analysis," *Advances in Bioinformatics*, vol. 2017, p. 1278932, 2017, epub 2017 Jul 18.

[7] M. A. M. Faysal and S. Arifuzzaman, "A comparative analysis of large-scale network visualization tools," in *2018 IEEE International Conference on Big Data (Big Data)*. Seattle, WA, USA: IEEE, 2018, pp. 4837–4843.

[8] M. E. Newman, "The structure and function of complex networks," *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.

[9] A.-L. Barabási and Z. N. Oltvai, "Network biology: understanding the cell's functional organization," *Nature reviews genetics*, vol. 5, no. 2, pp. 101–113, 2004.

[10] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014.

[11] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group formation in large social networks: membership, growth, and evolution," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '06. ACM, 2006, pp. 44–54. [Online]. Available: http://dx.doi.org/10.1145/1150402.1150412

[12] B. Yang, D. Liu, and J. Liu, *Discovering Communities from Social Networks: Methodologies and Applications*. New York, NY: Springer, 2010, ch. 16.

[13] J. Leskovec and A. Krevl, "Email-euall dataset," http://snap.stanford.edu/data/email-EuAll.html, n.d., stanford Network Analysis Project (SNAP).

[14] J. McAuley and J. Leskovec, "Ego-facebook dataset," http://snap.stanford.edu/data/ego-Facebook.html, n.d., stanford Network Analysis Project (SNAP).

[15] F. M., L. C.T., H. M., and C. contributors. (2023) Cytoscape.js: Graph theory library for visualisation and analysis. Cytoscape Consortium. [Online]. Available: https://js.cytoscape.org/

[16] "Gephi: The open graph viz platform," https://gephi.org, accessed: [Insert Date].

[17] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016.

[18] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems*, 2017.

[19] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2017.

[20] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2018.

[21] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in neural information processing systems*, 2013.

[22] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[23] P. T. Inc., "Plotly open source graphing library for python," 2021, accessed: insert-date-you-accessed-the-resource. [Online]. Available: https://plotly.com/python/