JCST Journal of
Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7766284

GRAPH SIGNAL PROCESSING ANALYSIS BASED ON GRAPH DATA SCIENCE: INTERPRETABILITY, REPLICABILITY

Zareena Begum
Assistant Professor, Dept. of Computer science and engineering (DataScience)
Vaagdevi College of Engineering

M.Rama
Assistant Professor, Dept. of Computer science and engineering
Vaagdevi College of Engineering

Dr. Ayesha Banu
Associate Professor, Dept. of Computer science and engineering (DataScience)
Vaagdevi College of Engineering

ABSTRACT

There are numerous modern data science challenges that make use of graphs (networks) as a representation of data, including those dealing with social, biological, and communication networks. There has been a rise in the use of signal processing and ML techniques for graph-based data analysis in the past decade. The prevalence of graphs and graph-based learning challenges across a wide range of applications has increased the interest in exploring explainability in graph data science. Since identifying communities is the first order of business when mining graphs for insights, we'll utilise that as a lens through which to investigate the challenge of explaining graph data science. Communities are formed when people with shared interests get together, and they are dense subnetworks of the larger network. Though many approaches to community discovery work well with artificial networks that have a clear modular structure, the quality and impact of these algorithms' results when applied to real-world networks with a more nuanced modular structure are less certain. In this paper, motivated by recent advances in explainable AI and ML, we offer methods and metrics from network science to quantify three separate elements of explainability in the context of community detection: interpretability, replicability, and reproducibility.

1. INTRODUCTION

Model explainability, interpretability, and reproducibility

Data is fed into the model, and then the model "does its thing" and spits out a prediction. The lack of clarity is problematic in many ways, and is often encapsulated by the vaguely related concepts of explainability, interpretability, and reproducibility.

- Explainability: Providing the human-friendly explanation of how a Machine Learning (ML) model works
- Interpretability: Indicates the ability to:
  - Learn how model inputs, features, and outputs are connected.
  - Anticipate outputs based on inputs.

- Reproducibility: means that the model may reliably be used to get the same results given the same inputs

CML's end-to-end model governance and monitoring process attempts to eliminate the blackbox aspect of most machine learning models and provide greater insight into an organization's machine learning operations.

White-box or translucent models are another name for this type of model. Within the realm of Data Science, black-box models and white-box models are often seen as two opposite ends of the spectrum (see Figure 1).

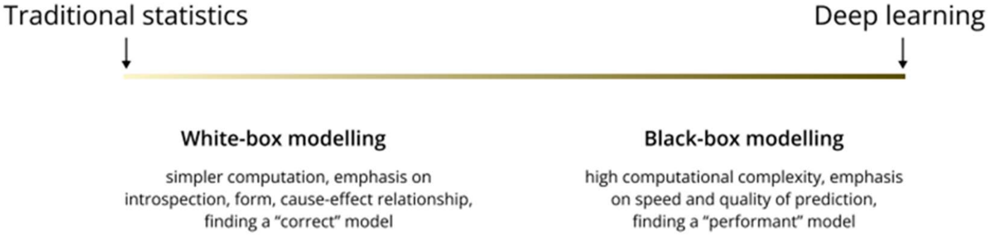
Traditional statistics
Deep learning
White-box modelling
simpler computation, emphasis on introspection, form, cause-effect relationship, finding a "correct" model
Black-box modelling
high computational complexity, emphasis on speed and quality of prediction, finding a "performant" model

Figure 1: Interpretability (left) vs performance (right) spectrum in Data Science.

Massive warehouses of structured data, where the structure itself reveals vital information about the data's nature, are the norm in modern data analysis. Many times, graphs (a mathematical tool) are used to show the structure of such data.

These days, everyone knows everyone else. Information gathered from complex networks of people, things, and money becomes widely available. The machine learning community has also shown a lot of enthusiasm for processing and analysing network data during the past few decades [1, 2, 3]. Interdependencies between data elements in the form of pairwise relationships are just one example of how the network structure inherently transmits information about the data. Graphs and other mathematical representations have long been used to convey these characteristics.

In this setting, novel tendencies and difficulties have been rapidly emerging. Take, for instance, the relationship between a network of protein-protein interactions and the temporal variation in gene expression. Discovery of important genes (through protein grouping) affected by the infection and prediction of how the host organism reacts (in terms of gene expression) to infections over time are two typical tasks in network biology involving such data that can advise the optimal intervention techniques.

2. LITERATURE REVIEW

Graphs are a special kind of database that may be used to depict the relationships between a collection of "nodes" (edges). Researchers have been putting a lot of effort into finding ways to use machine learning to analyse graphs because of their flexibility. Graphs are used extensively in fields as diverse as social science (social networks; Wu et al., 2020), physics (Sanchez et al., 2018; Battaglia et al., 2016), and biology due to their flexibility as a representational tool (protein-protein interaction networks; Fout et al., 2017). That was the conclusion reached by researchers (Khalil et al., 2017).

In the field of machine learning, graph analysis stands out as a distinct subset of non-Euclidean data structures due to its major focus on node categorization, link prediction, and clustering. Graph neural networks (GNNs) are a powerful tool in the field of deep learning. GNN is widely used because of its effective outcomes in graph analysis. Following is an example-filled breakdown of the fundamental concepts underlying graph neural networks.

The long tradition of graph neural networks is a major inspiration for GNNs. In the 1990s, researchers first began implementing recursive neural networks onto directed acyclic graphs (Sperduti and Starita, 1997; Frasconi et al., 1998). Then, in an effort to deal with cyclical patterns, two types of neural networks—Recurrent Neural Networks and Feedforward Neural Networks—are introduced to the area (Scarselli et al., 2009; (Micheli, 2009). Both approaches aim to construct state transition systems on graphs and iterate until convergence, which severely limits both their scalability and their describability.

GNNs have been rediscovered thanks to advancements in deep neural networks, most notably convolutional neural networks (CNNs) (LeCun et al., 1998). Improvements in nearly every domain of machine learning led to the development of CNNs, which can extract multi-scale localised spatial data and integrate it to produce highly expressive representations (LeCun et al., 2015). Typical characteristics of CNNs include the use of numerous layers, shared weights, and local connections (LeCun et al., 2015). In addition, these are essential in addressing problems that can be represented as graphs. Even while data structures like pictures (2D grids) and texts (1D sequences) can be thought of as examples of graphs, CNNs can only handle data that conforms to the standard Euclidean format. This makes it simple to generalise CNNs on graphs.

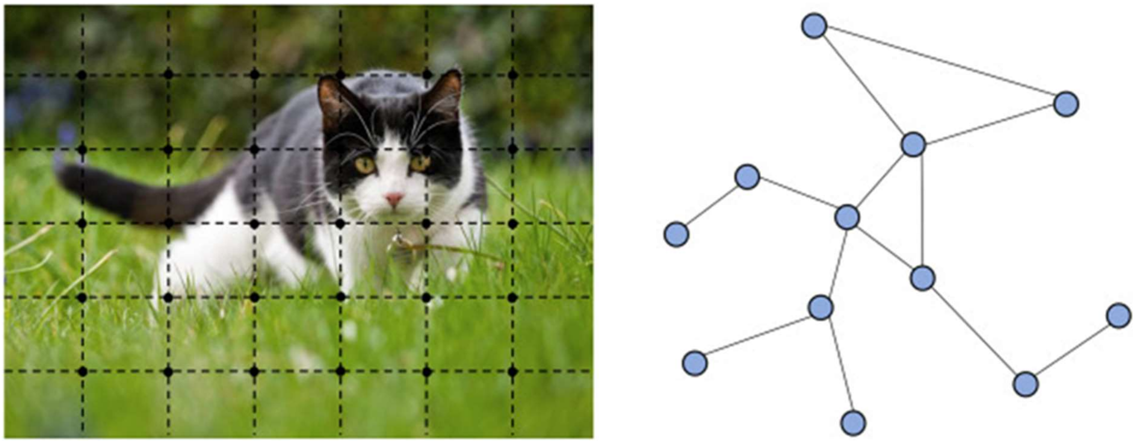

Fig. 2. Images in Euclidean space are shown on the left. Exactly: a graph in a space other than Euclidean space.

The challenge of creating localised convolutional filters and pooling operators makes it hard to transfer CNN from the Euclidean domain to the non-Euclidean domain (see Fig. 2). Bringing the benefits of deep neural models to non-Euclidean environments is the focus of geometric deep learning, a burgeoning area of research (Bronstein et al., 2017). To be more specific, this term is being used to describe the widespread interest in using deep learning on graphs.

GNN models are the primary topic of the most recent GNN survey articles, such as those by Zhang et al. (2018b), Wu et al. (2019a), and Chami et al. (2020). In their classification of GNNs, Wu et al. identify four distinct subtypes: recurrent graph neural networks, convolutional graph neural networks, graph autoencoders, and spatial-temporal graph neural networks (2019a). Two recent studies, Zhang et al. (2018b) and Chami et al. (2020), present thorough overviews of graph deep learning approaches, with the latter proposing a Graph Encoder Decoder Model to integrate network embedding and graph neural network models. In this research, we present a novel categorization system for these techniques, with an emphasis on the classic versions of GNNs. In addition, we provide a detailed account of the many fields to which GNNs have been applied and an overview of GNN variants for the many different kinds of graphs.

There have also been some surveys that have looked at specific areas of the graph learning environment in depth. Two excellent surveys of adversarial learning on graphs, covering both attacks and defences against graph data, may be found in Sun et al. (2018) and Chen et al. (2018). (2020a). The attention models for graphs are thoroughly reviewed by Lee et al (2018a). In a paper due out in 2020, Yang et al. demonstrate a technique for learning representations of networks with diverse nodes and links. Huang et al. provide a thorough analysis of the state-of-the-art in GNN models for dynamic graphs (2020). Peng et al. provide a concise summary of methods for combinatorial optimization using graph embeddings (2020). Conclusions about GNNs for heterogeneous graphs, dynamic graphs, and combinatorial optimization are drawn in Sections 4.2, 4.3, and 8.1.6, respectively.

3. GSP FOR EXPLOITING DATA STRUCTURE

Modeling, analysing, and manipulating graph-based data are all possible with GSP. For the sake of argument, let us assume that there are N nodes in set V and N less edges in set E. Our discussion will centre around the acyclic, weighted graph G V =,E. With the help of the function x:V " R, a scalar value is assigned to each node in a graph. So, GSP is primarily concerned with incorporating graph signals within the purview of more conventional signal processing techniques. The so-called graph Fourier transform, defined in terms of the eigenvectors of the graph Laplacian matrix or a graph shift operator (a generic matrix representation of the graph), is a significant recent development in the field (GFT). Laplacian L D = - W is a well-known example, where W is the weighted adjacency matrix of the graph and D is the degree matrix.

This insight can be applied in two different contexts. To start, the idea of frequency, which helps establish a measure of smoothness for signals on graphs, allows for a fresh perspective on the theory of kernels and regularisation on graphs [8]. The second benefit is that it allows for the generalisation of techniques such as convolution and frequency-domain filtering to graph signals. Therefore, with the aid of cutting-edge instruments made feasible by GSP-based principles, we may make use of established understanding of data structures and relationships. Since their inception, they have been put to use in numerous machine learning problems. Using examples from both supervised and unsupervised learning scenarios, we explain how to apply GSP when the underlying graph is already known or can be inferred from observations of different variables (for more on this topic, read the "Inference of Hidden Relational Structure for Interpretability" section).

3.1 Regression for multiple response variables

Learning a mapping from a set of inputs (or features) to a real-valued output is what regression aims to do; it is one of the simplest forms of supervised learning (or response). Learning such a mapping for a large number of response variables is crucial in the larger problem of multivariate regression. Modeling the relationships between response variables is a common concept used in modern methods. Both the coefficients of a multivariate autoregressive process in a parametric model and the kernel function of a multioutput Gaussian process in a nonparametric model can be thought of as representations of these sorts of relationships (GP). For modelling interactions between response variables, separable kernels for multioutput GP use a positive semidefinite (PSD) matrix as part of the kernel function.

$$\mathbf{y} = \mathbf{B}\mathbf{x} = (\mathbf{I} + \alpha\mathbf{L})^{-1}\mathbf{x} = \mathbf{\Phi}(\mathbf{I} + \alpha\mathbf{\Lambda})^{-1}\mathbf{\Phi}^T\mathbf{x},$$

to which a hyperparameter an is added, where I is an identity matrix. Keep in mind that the filter matrix B, a function of the network Laplacian L, encodes the relationships between the data points acquired by each node. The authors of [10] suggest a GP model on graphs with a kernel function similar to that of a multi-output GP by building the PSD matrix between observations of the response variables using B.

3.2 Classification using graph theory

Classification is an essential type of supervised learning that is quite similar to regression except that it uses a categorical response variable. Most notably, recent developments in deep learning have enabled significant gains in applications like image classification thanks in large part to the use of convolutional neural networks (CNNs). However, due to the absence of the notions of shift and convolution in the irregular graph domain, standard convolutional neural networks (CNNs) cannot be utilised directly to categorise signals that rest on a graph structure. To combat such an issue, GSP is an effective technique. To be more specific, the graph spectral domain, when supplemented with a concept of frequency and a GFT, can be used to explain convolution implicitly. Here we consider a graph with a signal x and a convolution kernel gt () m defined on the Laplacian eigenvalues of the graph. Multiple definitions exist for the xg convolution.

$$\mathbf{x} * g = \mathbf{\Phi}\hat{g}(\mathbf{\Lambda})\mathbf{\Phi}^T\mathbf{x} = \hat{g}(\mathbf{L})\mathbf{x}.$$

3.3 Graph-based clustering and dimensionality reduction

Clustering and dimensionality reduction are two of the most studied topics in unsupervised learning, another fundamental paradigm in machine learning. Clustering algorithms that rely on graph theory often use a similarity graph to partition a large set of nodes into smaller, more manageable clusters. Many network science and machine learning experts have studied this phenomenon. Among the many methods developed during the past two decades, spectral clustering has seen explosive growth in popularity.

The notions supplied by GSP have been useful in a number of similar contexts, despite the fact that graph-based clustering does not often contain node properties (i.e., graph signals). Multiscale clustering is a technique where many clustering results are sought, each of which represents a separate grouping of nodes at a different scale, as the name suggests. Transforms that operate on several scales in signal processing, such as the wavelet transform, offer a practical alternative to traditional methods that require adapting concepts like modularity to the multiscale setting.

3.4 GSP for improving efficiency and robustness

The original goal of Graph Structured Programming (GSP) was to provide a framework for using graphs in data analysis; nevertheless, many traditional machine learning methods have found success by adapting some of GSP's core methodology and tools. For tasks like (semi-)supervised learning, few-shot learning, zero-shot learning, and multitask learning, improved resistance against sparse and noisy training data or hostile instances has been shown. Computing complexity and time required for graph-based learning can be reduced by using simple GSP operations like sampling and filtering in big data processing and statistical learning.

CONCLUSION

In recent years, graph neural networks have evolved into potent and practical tools for use in graph-based machine learning applications. These innovations are the result of progress made in the areas of expressive capability, model flexibility, and training methods. Graph neural networks are the focus of this survey. In our novel GNN models, we introduce variants that vary in the computing modules they use, the kinds of graphs they employ, and the ways in which they are trained. We also offer a number of theoretical evaluations and overviews of several overarching frameworks. Utilizing a taxonomy-based framework, we evaluate GNN applications in both structural and non-structural settings, as well as a brief overview of other settings. Finally, we propose four open topics—resilience, interpretability, pretraining, and modelling complex structures—that point to the most pressing challenges and interesting future study fields for graph neural networks.

REFERENCES

[1] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” IEEE Signal Process. Mag., vol. 30, no. 3, pp. 83–98, 2013. doi: 10.1109/MSP.2012.2235192.

[2] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs,” IEEE Trans. Signal Process., vol. 61, no. 7, pp. 1644–1656, 2013. doi: 10.1109/ TSP.2013.2238935.

[3] A. Ortega, P. Frossard, J. Kovacˇevic´, J. M. F. Moura, and P. Vandergheynst, “Graph signal processing: Overview, challenges, and applications,” Proc. IEEE, vol. 106, no. 5, pp. 808–828, 2018. doi: 10.1109/JPROC.2018.2820126.

[4] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, “Geometric deep learning: Going beyond Euclidean data,” IEEE Signal Process. Mag., vol. 34, no. 4, pp. 18–42, 2017. doi: 10.1109/MSP.2017.2693418.

[5] N. Tremblay and A. Loukas, “Approximating spectral clustering via sampling: A review,” in Sampling Techniques for Supervised or Unsupervised Tasks, F. Ros and S. Guillaume, Eds. New York: Springer-Verlag, pp. 129–183, 2020.

[6] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro, “Connecting the dots: Identifying network structure via graph signal processing,” IEEE Signal Process. Mag., vol. 36, no. 3, pp. 16–43, 2019. doi: 10.1109/MSP.2018.2890143.

[7] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, “Learning graphs from data: A signal representation perspective,” IEEE Signal Process. Mag., vol. 36, no. 3, pp. 44–63, 2019. doi: 10.1109/MSP.2018.2887284.
[8] A. Smola and R. Kondor, “Kernels and regularization on graphs,” in Proc. Annu. Conf. Computational Learning Theory, 2003, pp. 144–158. doi: 10.1007/978-3-540-45167-9_12.
[9] M. A. Álvarez, L. Rosasco, and N. D. Lawrence, “Kernels for vector-valued functions: A review,” Found. Trends Mach. Learn., vol. 4, no. 3, pp. 195–266, 2012. doi: 10.1561/2200000036.
[10] A. Venkitaraman, S. Chatterjee, and P. Handel, “Gaussian processes over graphs,” 2018, arXiv:1803.05776