EasyChair Preprint

№ 9922

Demystifying the Applications of Deep Learning Using Graph Learning Models and Algorithms

A S Sasipriya

March 31, 2023

Demystifying the applications of deep learning using graph learning models and algorithms

Sasipriya A.S, Faculty of Engineering and Technology, Datta Meghe Institute of Higher Education and Research, Wardha, Maharashtra, India.

Abstract: Machine learning algorithms that handle a variety of data kinds are typically the backbone of real-world intelligent systems. Due to its inherent complexity, graph data have presented machine learning with hitherto unheard-of hurdles despite their widespread use. Since graph data are embedded in an irregular domain, unlike text, audio, and image data, some fundamental operations of current machine learning methods cannot be used. To address these issues, numerous graph learning models and algorithms have been created. In this article, the most recent graph learning techniques are reviewed in detail, along with some of their possible uses. The article has several functions. For researchers and practitioners in a variety of fields, including social computing, information retrieval, computer vision, bioinformatics, economics, and e-commerce, it serves as a rapid reference to graph learning. Second, it offers details about active fields of study in the area. Thirdly, it hopes to pique interest in graph learning and inspire fresh research ideas.

Keywords: Deep learning, network representation learning, graph signal processing, graph topology

I. Introduction

GRAPHS, often known as networks, can be derived from a variety of real-world relationships among numerous entities. Social networks, biological networks, patent networks, traffic networks, citation networks, and communication networks are examples of typical graphs that have been widely utilised to define various relationships [1]–[3]. A graph is frequently defined by two sets: vertex and edge. In a network, vertices represent entities, while edges represent relationships between them. Because of its many real-world applications, such as data mining and knowledge discovery, graph learning has gotten a lot of attention. Because graphs exploit crucial and meaningful interactions among vertices, graph learning methods have grown in favour for capturing complicated relationships [4], [5]. The spread trajectory of rumours can be traced in microblog networks, for example, by recognising information cascades. By inferring protein interactions in biological networks, new therapeutics for complex diseases can be discovered. Human mobility patterns in transportation networks can be predicted by looking at the co-occurrence phenomenon with different timestamps [6]. The way networks are represented has a huge impact on the efficiency of network analysis.

A. Graph Learning: Graph learning refers to machine learning on graphs in general. Graph learning methods link a graph's features to feature vectors in the embedding space with the same dimensions. Without projecting the network into a low-dimensional space, a graph learning model or algorithm translates the graph data straight into the output of the graph learning architecture. Because deep learning approaches can encode and display graph data as vectors, most graph learning methods are based on or generalised from them. The graph learning output vectors are in continuous space. The goal of graph learning is to extract desired graph features. We attempt to study machine learning algorithms on graphs that fall into the category, as indicated in Fig. 1: graph signal processing (GSP)-based methods. GSP is

concerned with graph sampling and recovery, as well as learning topology structure from data.

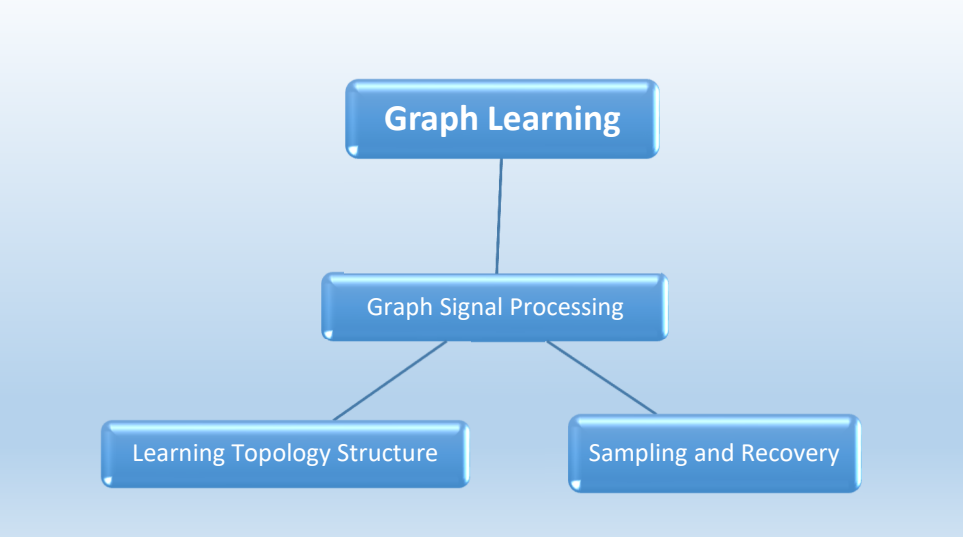

Fig. 1. Category of graph learning

The model architectures of the methods/techniques are fundamentally different. This article provides a comprehensive overview of current graph learning approaches. To represent a network, researchers have traditionally used an adjacency matrix, which can only record the relationship between two neighbouring vertices. This simple depiction, however, cannot capture many complicated and irregular structures. Traditional approaches for analysing large-scale networks are computationally expensive and difficult to deploy in real-world applications. As a result, proper representation of these networks is a critical issue to address [4]. Recently developed network representation learning (NRL) may learn latent network vertices properties with low dimensional representation [7]–[9]. Following the learning of the new representation, earlier machine learning methods can be used to analyse the graph data and identify hidden relationships. Communication patterns, community organisation, and information diffusion are all investigated using graphical network representation [12], [13]. Graph learning tasks can be split into three groups depending on the qualities of vertices, edges, and subgraphs: vertices-based, edges-based, and subgraph-based. Designing appropriate graph sampling algorithms to preserve the properties of the original graph is critical for GSP, which attempts to recover the original graph efficiently [16]. In the event of partial data, graph recovery algorithms can be employed to reconstruct the original graph [17]. The topology structure can then be learned from graph data via graph learning. In conclusion, graph learning can be used to solve the problems listed below, which are difficult to answer using typical graph analysis approaches [18].

1) Irregular domains: Traditional sensors collect data in a grid layout. Graphs, on the other hand, exist in an irregular domain (i.e., non-Euclidean space). Data in non- Euclidean space are not sorted in the same way that data in regular domains are. As a result, defining distance is challenging. As a result, typical machine learning and signal processing approaches cannot be readily applied to graphs.
2) Heterogeneous networks: Traditional graph analysis algorithms frequently include homogeneous networks. The suitable modelling approaches simply consider the network's direct connections and ignore all other useless data, greatly simplifying the processing. It

is, nevertheless, prone to causing data loss. The edges between vertices and the sorts of vertices are frequently different in the real world, as demonstrated in the academic network in Fig. 2. As a result, discovering potential value from heterogeneous information networks with many vertices and edges is difficult.

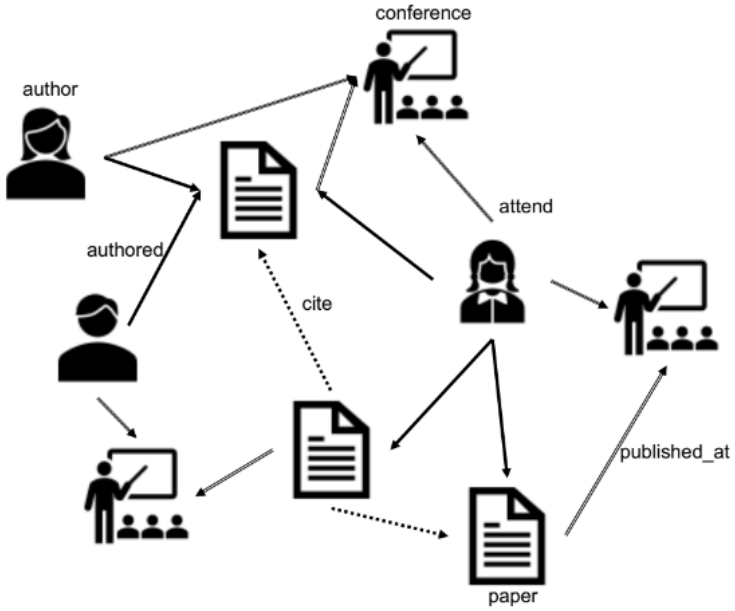

Fig. 2. Heterogeneous academic network [28]

3) Distributed algorithms: In large social networks, millions of vertices and edges are common [19]. Centralized methods can't manage this since the computing complexity of these algorithms increases dramatically as the number of vertex increases. A significant difficulty that has yet to be solved is the creation of distributed algorithms for coping with large networks [20]. Distributed algorithms have the advantage of being able to operate on numerous CPUs or GPUs at the same time, reducing the overall running time dramatically.

B. Related Surveys

There are various surveys that are linked to the topic of this article in some way. GSP is specifically mentioned as a method for graph learning that is not covered by previous studies. Zhang et al. [4] reviewed NRL approaches and addressed their applicability based on two categories: unsupervised NRL and semisupervised NRL. Nickel and colleagues [23] introduced two types of knowledge extraction methods: latent feature models and graph-based models. Akoglu et al. [24] examined state-of-the-art approaches for event identification in graphed data and their real-world applications. Wu et al. [25] evaluated and described current GNN algorithms and their applications in many sectors. GSP strategies for representation, sampling, and learning were introduced and described by Ortega et al. [26]. Huang et al. [27] looked at how GSP may be used in functional brain imaging and how to accomplish brain network analysis from a signal processing standpoint. In conclusion, none of the currently available surveys provide a comprehensive picture of graph learning. They barely go on a few aspects of graph learning, namely network embedding and deep learning-based network representation. The GSP techniques

are not covered by the NRL and/or GNN-based surveys. Graph learning is a crucial strategy for GNNs, thus we cover GSP techniques in the context of graph learning. This comprehensive article, in particular, integrates cutting-edge machine learning techniques for graph data, provides a general overview of graph learning, and examines its applications in numerous fields.

C. Contributions and Organization

This article's contributions can be summarised as follows.

1. A comprehensive overview of current graph learning methods: We provide an essential introduction to graph learning methods, which includes technical sketches, application scenarios, and prospective research areas.
2. Graph learning taxonomy: We present a technical classification of popular graph learning methods based on theoretical models. Wherever possible, technical descriptions are supplied to aid understanding of the taxonomy.
3. Insights regarding graph learning's future directions: We summarise many outstanding issues and pertinent challenges to throw light on prospective research directions in the field of graph learning, in addition to qualitative examination of existing methods.

The rest of this article is organized as follows. An overview of graph learning approaches containing GSP-based methods is provided in Section II. The applications of graph learning are examined in Section III. Some future directions as well as challenges are discussed in Section IV. Finally, Section V concludes the article.

II. GRAPH LEARNING MODELS AND ALGORITHMS

In prior machine learning methods, the feature vectors that represent distinct categorical attributes were used as input. Graph learning [21] must, however, handle the mapping from the input feature vectors to the output prediction results. Deep learning is considered one of the most successful artificial intelligence approaches [29], [30]. In domains like pattern recognition and image processing, extracting complex patterns using deep learning from a large amount of irregular data has shown to be quite effective. As a result, there has been a lot of interest in how to use deep learning techniques to extract patterns from complex graphs. GNNs, GCNs, and GAEs are examples of deep learning on graphs. In addition, GSP has been proposed for graph analysis [26]. A set of values is stored on a set of vertices, which are connected by edges in one of the most common instances [31]. Graph signals can be used to model a variety of real-world occurrences. Users on Facebook, for example, can be described as vertices and their friendships as edges in social networks. In this social network, each vertex's number of followers is indicated.

Many standard signal processing techniques (e.g., convolution, filter, wavelet, etc.) can be used for GSP with appropriate changes based on this assumption [26]. We examine graph learning models and algorithms of GSP-based methods in this section. The abbreviations used in this article are listed in Table I.

TABLE I DEFINITIONS OF ABBREVIATIONS

Abbreviation	Definition
PCA	Principal component analysis
NRL	Network representation learning
LSTM	Long short-term memory (networks)
GSP	Graph signal processing
GNN	Graph neural network
GMRF	Gauss markov random field
GCN	Graph convolutional network
GAT	Graph attention network
GAN	Generative adversarial network
GAE	Graph auto-encoder
ASP	Algebraic signal processing
RNN	Recurrent neural network
CNN	Convolutional neural network

Graph Signal Processing : Signal processing is a well-known topic that deals with signals in a conventional data domain. Researchers have recently extended classic signal processing techniques into graphs [22]. Graph analysis can be done using traditional signal processing techniques and tools like the Fourier transform and filtering. Graphs, in general, are a type of irregular data that is difficult to handle directly. GSP presents a unique perspective on spectral analysis of graphs as a complement to learning methods based on structures and models. GSP, which is derived from signal processing, can explain graph properties such as connectedness, similarity, and so on.

1)*Representation on Graphs:* The rapid rise of graph learning has been aided by a meaningful representation of graphs. GSP is divided into two types: adjacency matrix-based GSP [31] and Laplacian-based GSP [32]. Algebraic signal processing (ASP) [33], which interprets linear signal processing from algebraic theory, provides an adjacency matrix-based GSP. Signals, filters, and signal modification are all part of linear signal processing. It works in both discrete and continuous time domains. In ASP, the fundamental postulate of linear algebra is extended to algebra space. ASP can get distinct examples in linear signal processing by choosing the right signal model. The signal model in adjacency matrix-based GSP is created from a shift. A shift in GSP is a graph domain filter, similar to classical signal processing [31], [34], [35]. Adjacency matrices are commonly used in GSP to create graph signal models. A graph's signals are generally defined at its vertices. Spectral graph theory gives rise to Laplacian-based GSP. A portion of the Laplacian basis is used to convert high-dimensional data into a low-dimensional space [36]. To achieve distributed processing of graph signals, some researchers used sensor networks [37]. Other scholars solved the problem at a global level, assuming a smooth graph.

The Laplacian matrix, unlike adjacency matrix-based GSP, is symmetric with real and nonnegative edge weights and is used to index undirected graphs. Although several matrices are used as basic shifts in the models, most of the concepts in GSP are drawn from signal processing. In these frameworks, concepts with distinct definitions may have comparable meanings. They all correspond to signal processing ideas. Signals in GSP are graphed values that are commonly expressed as a vector, $s = [s_0, s_1, \dots, s_{N-1}] \in \mathbb{C}^N$. Each element in the vector represents the value on a vertex, and N is the number of vertices. Even

while most applications are based on real-value signals, certain studies [26] enable complex-value signals.

A graph can be represented as a triple G(V, E, W) in adjacency matrix-based GSP, where V is the vertex set, E is the edge set, and W is the adjacency matrix. We can define degree matrix $D_{ii} = d_i$ with the definition of graphs, where D is a diagonal matrix and d_i is the degree of vertex i. L = D-W is the graph Laplacian formula, while $L_{norm} = D^{-1/2}LD^{-1/2}$ is the normalised Laplacian formula. Filters in signal processing can be thought of as a function that amplifies or decreases relevant frequencies while removing others that aren't. In linear space, matrix multiplication equals scale shifting, which is the same as filtering in the frequency domain.

The GSP formula $s_{out} = H\, s_{in}$ where H stands for a filter, makes it clear that matrix multiplication can be used as a filter. Shift is a key concept for describing signal variation, and time-invariant signals are widely utilised [31]. In fact, there are several options for GSP shifts. The shift in an adjacency matrix-based GSP is A. L is used in Laplacian-based GSP [32], and other matrices are also used by some studies [38]. Shift invariance is defined in GSP by following time invariance in traditional signal processing. Shift-invariant filters are commutative with shift, which can be represented as AH = HA. It is shown that the shift can represent a shift-invariant filter. The features of shift are crucial, as they impact the popularity of other terms like Fourier transform and frequency. Shift A's eigenvalue decomposition in adjacency matrix-based GSP is $A = V\Lambda V^{-1}$ where V is the eigenvector matrix $[v_0, v, \dots, v_{N-1}]$ and $\Lambda = \begin{bmatrix} \lambda_0 & & \\ & \ddots & \\ & & \lambda_{N-1} \end{bmatrix}$ is a diagonal matrix of eigen values. The Fourier transform matrix, or $F = V^{-1}$ is the inverse of V. Total variation, or the difference following a shift, is the definition of frequency of shift $TV_G = \left\| v_k - \frac{1}{\lambda_{\max}} A v_k \right\|_1$ here the normalized factor of the matrix is $\frac{1}{\lambda_{\max}}$. This indicates that the frequencies of eigenvalues on the complex plane that are far from the biggest eigenvalues are high. After shift filtering, a large frequency indicates that signals are modified on a broad scale. Fig.3 depicts the variations between minimum and maximum.

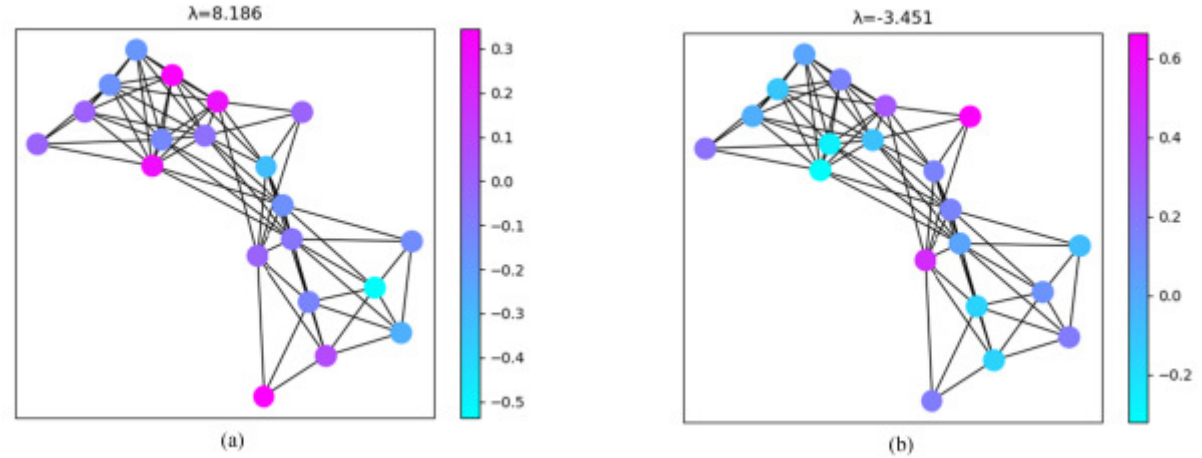

Fig. 3. Illustration of difference between minimum and maximum frequencies. (a) The maximum frequency and (b) The minimum frequency.

In general, when the overall variation is lower than the frequency, and vice versa, the frequency tends to be larger. While larger eigenvectors can be used to create low frequency filters that capture fundamental features, smaller eigenvectors can be utilised to capture variance between neighbouring nodes. We can distinguish the correct answers for topology learning tasks based on known information. When topology information is only partially known, we can infer the entire graph using the known data. If the topology information is unknown but the signals on the graph can still be seen, the topology structure must be derived from the signals. Blind topological inference is also known as graph topology (or structure) learning, and the former is frequently solved as a sampling and recovery issue.

2) *Sampling and Recovery*: In conventional signal processing, we frequently need to replicate original signals with the fewest samples possible while preserving all of the original signals' information. Data is lacking when there are few samples, whereas more samples necessitate greater storage space. The prerequisites for full time-domain signal recovery are specified by the well-known Nyquist Shannon sampling theorem. Researchers have moved sampling ideas into GSP in order to study the problem of sampling on graphs. Because some real-world applications, including sensor networks and social networks, contain massive data volumes, sample less and recover better are crucial for GSP. The majority of techniques and frameworks for resolving sampling problems really require the network to represent the correlations within the signals that were captured on it [39]. The sampling problem is defined as reconstructing signals from samples on a subset of vertices, and the signals in this problem are typically band-limited. The graph signals were added to the Nyquist-Shannon sampling theorem in [40]. For GSP, the normalised Laplacian matrix is used to define the sampling theorem and cutoff frequency. Additionally, the author provided instructions on how to choose the sample set for a particular bandwidth and how to get the cutoff frequency from a sampling set. It is important to emphasise that the sampling theorem there only applies to undirected graphs. Because the Laplacian matrix only characterizes undirected graphs, the sampling theory for directed graphs makes use of an adjacent matrix. An optimum operator that ensures perfect recovery and is noise-resistant for generic graphs was proposed in [35]. The fact that the signals of the former fall under the regular domain while those of the later fall under the irregular domain is one of the explicit differences between GSP and traditional signal processing. Classical signal processing can sample subsequent signals and recover subsequent signals from samplings to address sampling and recovery issues. GSP samples a discrete sequence and then reconstructs the original sequences from the samples.

By approaching the problem in this specific way, it is usually possible to divide it into two parts: selecting sampling vertex sets and reconstructing the original signals using various models.However,some algorithms call for matrix decomposition for a huge dataset in order t o save eigenvalues and retrieve frequencies, which is nearly difficult to accomplish. A rando m method can also be used in sampling as an easy method that can be applied to big datasets. Two sample strategies were put out by Puy et al. [41]: an adaptive random sampling strategy and a nonadaptive one that depends on a parameter. They expanded random sampling to large-scale graphs by loosening the optimal limitation. An further popular method is greedy sampling. To determine cutoff frequency precisely, Shomorony and Avestimehr [42] suggested an effective technique based on linear algebraic conditions. In the worst-case scenarios, Chamon and Ribeiro's [43] near-optimal guarantee for greedy sampling ensures that

it will perform as expected. Selective sampling, in which signals are seen on a selection of vertices, can be applied to all of the aforementioned sample techniques [43]. A type of sampling termed aggregation sampling [44], which employs observations taken at a single vertex as input and sequentially applies the graph shift operator, exists in addition to choosing sampling. As in conventional signal processing, the reconstruction work on graphs can also be seen as a data interpolation problem [45]. Projecting the samples into the proper signal space allows researchers to acquire interpolated signals. Reconstruction using least squares is a useful technique. Gadde and Ortega's generative model for signal recovery was developed using a pairwise Gaussian random field (GRF) and a covariance matrix on graphs [46]. The maximum posterior inference of GRF with low-rank approximation can be thought of as the reconstruction of graph signals under sampling theorem. In order to achieve the distributed reconstruction of time-varying band limited signals, Wang et al. [47] devised iterative distributed least squares reconstruction (DLSR). Time-varying signals can be traced and flawlessly recreated using DLSR. Di Lorenzo et al. [48] established the linear mean squares (LMS) approach for adaptive estimation. LMS can enable online reconstruction and tracking by using a subset of the observed vertices. It enables the subset to change over time as well. Additionally, a sparse online estimator was suggested as a solution to the issues with uncertain bandwidth. Smoothness is another popular method for restoring original signals. In graph signals with low frequencies, missing values are inferred using smoothness. The term "local set" was defined by Wang et al. [17]. Two iterative methods to recover band-limited signals on graphs were presented based on the notion of graph signals. Additionally, Romero et al [49]'s recommendation of kernel regression as a paradigm for GSP modelling and reconstruction is noteworthy. To address a single optimization problem for parameter selection in estimators, two multikernel techniques were proposed. Other problems with compressed sensing recovery were also investigated by several researchers [50]. Additionally, investigations on sampling a variety of signals, such as piecewise smooth, piecewise constant, and smooth graph signals, have been conducted [51]. Chen et al. [51] offered a unified approach for analysing graph signals. In [52], it was investigated how to recover a known graph signal with sparse signal, or few nonzero vertices. Three distinct rebuilding strategies that correspond to various seeding patterns were examined. By comparing single simultaneous injection, single consecutive value injection, and multiple successive simultaneous injections, the prerequisites for flawless reconstruction on any vertices were discovered.

3) Learning Topology Structure From Data: Most application sceneries use links between entity correlations to build their graphs. For example, in sensor networks, the correlations between the sensors usually correspond to their distance. Social networks' edges are connections between people, such as friends or coworkers [53]. Edges in biological networks are the result of interactions. Despite the fact that GSP is a helpful framework for dealing with graph-related problems including sampling, reconstruction, and detection, it is lacking a step for extracting relationships from datasets. Numerous datasets still include linkages even when there are no explicit records. There are a few different ways to infer them, which is fortunate. In order to learn whole graphs from datasets, scholars are interested in estimating graph Laplacian. The problem of learning a graph from a dataset is defined as a Laplacian, or graph topology[54]. Typically, they demand that the graph satisfy specific requirements, including smoothness and sparsity. Smoothness is a typical assumption in networks built from datasets. It is therefore frequently used to restrict observed signals and provide graph signals a logical

guarantee. Researchers have utilised it for graph topology learning. According to the theory underpinning smoothness-based algorithms, the lowest frequency is often attained by shifting the result because the majority of signals on a graph are stationary. A popular theory for teaching graph topology in GSP is the Gauss Markov random field (GMRF) [54, [57], [58]. The graphs that are chosen by the GRMF-based graph topology learning models are those that are more likely to produce signals that resemble those produced by GMRF. Egilmez *et al.* [54] formulated the problem as a maximum posterior parameter estimation of GMRF, and the graph Laplacian is a precision matrix. Pavez and Ortega [57] also formulated the problem as a precision matrix estimation, and the rows and columns are updated iteratively by optimizing a quadratic problem. Both of them restrict the result matrix, which should be Laplacian. In [58], Pavez *et al.* chose a two steps framework to find the structure of the underlying graph. First, a graph topology inference step is employed to select a proper topology. Then, a generalized graph Laplacian is estimated. An error bound of Laplacian estimation is computed. In the next step, the error bound can be utilized to obtain a matrix in a specific form as the precision matrix estimation. It is one of the first work that suggests adjusting the model to obtain a graph satisfying the requirement of various problems.

A useful model that can be used to address the topology interfering problem is diffusi on [39],[59]-[61]. Diffusion describes how a node continuously affects the areas around it. In graphs, nodes with higher values will have an increased influence on nodes that are close by. Using a limited number of components to represent signals will make it easier to identify the crucial components in signal production. Diffusion models commonly make the assumption that identical signals are dispersed independently. Pasdeloup et al. [59] established the concept of valid graphs and assumed that the signals are identified after diffusion in order to explain signals. Segarra et al. [60] also agreed that the shift proceeds via a diffusion process and that indicators can be noticed. The signals in [61] were explained by a linear combination of a few components. Researchers sought to build time-sequential networks for time series captured in data. For example, Mei and Moura [62] introduced an estimation approach for graphs that takes into account both time and spatial dependencies and models them using an autoregressive process. An approach that Segarra et al. [63] suggested can be thought of as a development of graph learning. The article's goal was to find a solution to the issue of jointly identifying a graph filter and its input signal. A well-known partial inference issue for recovery algorithms is recommendation [45], [64], and [65]. Collaborative filtering (CF) is the common algorithm used in recommendation [66]. The goal of CF is to estimate the entire rating matrix from the observed ratings in a matrix. On networks expressing correlations between people and items, Huang et al research [65] showed that CF might be thought of as a particular band-stop graph filter. Furthermore, the band-limited interpretation problem can be used to describe linear latent factor approaches.

Discussion: Because GSP algorithms place stringent restrictions on experimental data, they are less frequently used in practical settings. Additionally, as GSP techniques require the input data to precisely match the entire graph, only a portion of the graph's data can be used as an input. As a result, this type of methods may have a high computational complexity. The scalability of GSP algorithms is subpar when compared to other types of graph learning techniques.

III. APPLICATIONS

Graph learning techniques, including as supervised, semisupervised, unsupervised, and reinforcement learning, can address a wide range of issues. The applications of graph learning have been categorised by some researchers into three categories: structural scenarios, nonstructural scenarios, and other application scenarios [18]. Explicit relational structures, such as those seen in physical systems, chemical structures, and knowledge networks, are examples of structural situations. Nonstructural scenarios are situations where data, such images and messages, have ambiguous relationship structures. Combinatorial optimization issues and model integration are two examples of other application scenarios.

A. Datasets and Open-Source Libraries: A variety of datasets and benchmarks are used to assess how well graph learning techniques perform in solving issues. numerous tasks, including graph visualisation, node categorization, and link prediction. For instance, nodes, edges, labels, or node attributes are included in datasets like Cora1 (a citation network), Pubmed2 (a citation network), BlogCatalog3 (a social network), Wikipedia4 (a language network), and PPI5 (a biological network). Several academic institutions have created libraries of graph learning algorithms, both common and traditional. For instance, OpenKE6 is a PyTorch-based Python toolkit for knowledge graph embedding. The implementations of RESCAL, HolE, DistMult, ComplEx, etc. are available in the open-source framework. The CogDL7 framework for learning graph representations can be used for a variety of tasks, including graph classification, link prediction, and node classification.

B. Text: Many data sources, including web pages, emails, documents (technical and corporate), books, digital libraries, consumer complaints, letters, and patents, all contain large amounts of text. Given that writing frequently provides rich contextual information, textual data are poorly organised for extracting any relevant information. Text can be used in a wide variety of applications, such as text categorization, sequence labelling, sentiment classification, etc. One of the oldest issues in natural language processing is text classification.

C. Images: Social relationship understanding, image classification, visual question answering (VQA), object recognition, region classification, semantic segmentation, and other applications of graph learning to images are included. Automatically analysing these ties is crucial for comprehending human actions since in the actual world, social relationships like friendships serve as the foundation of social networks. In order to learn a propagation mechanism, GRM introduces GGNNs. The classic challenge of image categorization has seen promising results using GNNs. Natural language processing and computer vision are both used in the learning activity known as VQA. A VQA system receives an open natural language query in the form of a specific image as input and produces a natural language response as output.

IV. OPEN ISSUES

A. Dynamic Graph Learning

We briefly review a number of potential future research topics and open graph learning issues in this section. Most current techniques are acceptable for static networks without any particular restrictions for the purpose of graph learning. Dynamic networks, like traffic networks, change with time, though. They are challenging to manage as a result. In the

literature, dynamic graph learning methods have rarely been investigated. Designing dynamic graph learning algorithms to retain good performance is crucial, especially when dealing with dynamic graphs.

B. Generative Graph Learning

Generative graph learning algorithms, which were inspired by GANs, can combine the generative and discriminative models by engaging in a game-theoretical min-max game. By improving the performance of generative and discriminative models alternately and repeatedly, this generative graph learning technique can be utilised for link prediction, network evolution, and recommendation.

C. Fair Graph Learning

The majority of graph learning techniques use deep neural networks, and the generated vectors might have unintentionally gathered sensitive data. Since the bias in the network is perpetuated, it is crucial to incorporate fair metrics into the graph learning algorithms to solve the problem of inherent bias.

D. Interpretable Graph Learning

By including both network structure and feature information, graph learning models are typically complex. Since the structures of graph learning algorithms are still a mystery, the interpretability of graph learning (based) algorithms remains an open problem. Examples of applications for graph learning algorithms include drug discovery. However, neither the method nor the rationale for this drug's invention are known. It is important to continue researching the interpretability of graph learning.

Conclusion:

This article provides a general overview of graph learning and a thorough analysis of the cutting-edge graph learning techniques. GSP-based techniques primarily describe the graph learning applications that have been demonstrated in fields including text, images, science, knowledge graphs, and combinatorial optimization. We then go over some potential future areas for graph learning research. At this time, graph learning is expanding at an unprecedented rate. We sincerely hope that this study will aid practitioners and researchers in their work on graph learning and related fields.

1[Online]. Available: https://relational.fit.cvut.cz/dataset/CORA
2[Online]. Available: https://catalog.data.gov/dataset/pubmed
3[Online]. Available: http://networkrepository.com/soc-BlogCatalog.php
4[Online]. Available: https://en.wikipedia.org/wiki/Wikipedia:Database_download
5[Online]. Available: https://openwetware.org/wiki/Proteinprotein_interaction_databases
6[Online]. Available: https://github.com/thunlp/OpenKE
7[Online]. Available: https://github.com/THUDM/cogdl/

REFERENCES

[1] S. Fortunato et al., "Science of science," Science, vol. 359, no. 6379, 2018, Art no. eaao0185.

[2] J. Liu, J. Ren, W. Zheng, L. Chi, I. Lee, and F. Xia, "Web of scholars: A scholar knowledge graph," in Proc. 43rd Int. ACM SIGIR Conf. Res.Develop. Inf. Retrieval, Jul. 2020, pp. 2153–2156.

[3] J. Liu et al., "Shifu2: A network representation learning based model for advisor-advisee relationship mining," IEEE Trans. Knowl. Data Eng.,vol. 33, no. 4, pp. 1763–1777, Apr. 2021.

[4] D. Zhang, J. Yin, X. Zhu, and C. Zhang, "Network representation learning: A survey," IEEE Trans. Big Data, vol. 6, no. 1, pp. 3–28, Mar. 2020.

[5] K. Sun, J. Liu, S. Yu, B. Xu, and F. Xia, "Graph force learning," in Proc. IEEE Int. Conf. Big Data, 2020, pp. 2987–2994.

[6] F. Xia, J. Wang, X. Kong, D. Zhang, and Z. Wang, "Ranking station importance with human mobility patterns using subway network datasets," IEEE Trans. Intell. Transp. Syst., vol. 21, no. 7, pp. 2840–2852, Jul. 2020.

[7] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in Proc. 22nd ACM Int. Conf. Knowl. Discov. Data Mining,Aug. 2016, pp. 855–864.

[8] K. Sun, L.Wang, B. Xu,W. Zhao, S.W. Teng, and F. Xia, "Network representation learning: From traditional feature learning to deep learning,"IEEE Access, vol. 8, no. 1, pp. 205600–205617 , Nov. 2020.

[9] S. Yu, F. Xia, J. Xu, Z. Chen, and I. Lee, "Offer: A Motif dimensional framework for network representation learning," in Proc. 29th ACM Int.Conf. Inf. Knowl. Manage., 2020, pp. 3349–3352.

[10] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," IEEE Trans. Pattern Anal. Mach. Intell.,vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[11] T. Guo et al., "Graduate employment prediction with bias," in Proc. 34th AAAI Conf. Artif. Intell., 2020, pp. 670–677.

[12] F. Xia, A. M. Ahmed, L. T. Yang, J. Ma, and J. J. Rodrigues, "Exploiting social relationship to enable efficient replica allocation in ad-hoc social networks," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 12,pp. 3167–3176, Dec. 2014.

[13] J. Zhang, W. Wang, F. Xia, Y.-R. Lin, and H. Tong, "Data-driven computational social science: A survey," Big Data Res., vol. 21, Aug. 2020,Art. no. 100145.

[14] F. Xia, A. M. Ahmed, L. T. Yang, and Z. Luo, "Community-based event dissemination with optimal load balancing," IEEE Trans. Comput.,vol. 64, no. 7, pp. 1857–1869, Jul. 2015.

[15] F. Xia, H. Wei, S. Yu, D. Zhang, and B. Xu, "A survey of measures for network motifs," IEEE Access, vol. 7, no. 1, pp. 106576–106587, 2019.

[16] J. Leskovec and C. Faloutsos, "Sampling from large graphs," in Proc. 12th ACM Int. Conf. Knowl. Discov. Data Mining, Aug. 2006, pp. 631–636.

[17] X. Wang, P. Liu, and Y. Gu, "Local-set-based graph signal reconstruction,"IEEE Trans. Signal Process., vol. 63, no. 9, pp. 2432–2444, Mar. 2015.

[18] Z. Zhang, P. Cui, andW. Zhu, "Deep learning on graphs:Asurvey," IEEE Trans.Knowl. Data Eng., doi: 10.1109/TKDE.2020.2981333,Mar. 2020.

[19] J. Xu et al., "Multivariate relations aggregation learning in social networks," in Proc. ACM/IEEE Joint Conf. Digit. Libraries, Aug. 2020,pp. 77–86.

[20] H. D. Bedru et al., "Big networks: A survey," Comput. Sci. Rev., vol. 37,Aug. 2020, Art.

no. 100247.
[21] P. Goyal and E. Ferrara, “Graph embedding techniques, applications, and performance: A survey,” Knowl.-Based Syst., vol. 151, pp. 78–94,Jul. 2018.
[22] H. Cai, V. W. Zheng, and K. C.-C. Chang, “A comprehensive survey of graph embedding: Problems, techniques, and applications,” IEEE Trans. Knowl. Data Eng., vol. 30, no. 9, pp. 1616–1637, Feb. 2018.
[23] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, “A review of relational machine learning for knowledge graphs,” Proc. IEEE Proc. IRE, vol. 104, no. 1, pp. 11–33, Jan. 2016.
[24] L. Akoglu, H. Tong, and D. Koutra, “Graph based anomaly detection and description: A survey,” Data Mining Knowl. Discov., vol. 29, no. 3, pp. 626–688, May 2015.
[25] Z.Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A comprehensive survey on graph neural networks,” IEEE Trans. Neural Netw. Learn.Syst., vol. 32, no. 1, pp. 4–24, Jan. 2021.
[26] A.Ortega, P. Frossard, J. Kovaˇcevi´c, J. M. Moura, and P. Vandergheynst,“Graph signal processing: Overview, challenges, and applications,” Proc. IEEE Proc. IRE, vol. 106, no. 5, pp. 808–828, May 2018.
[27] W. Huang et al., “A graph signal processing perspective on functional brain imaging,” Proc. IEEE Proc. IRE, vol. 106, no. 5, pp. 868–885,May 2018.
[28] F. Xia,W.Wang, T. M. Bekele, andH. Liu, “Big scholarly data:Asurvey,” IEEE Trans. Big Data, vol. 3, no. 1, pp. 18–35, Mar. 2017.
[29] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553,2015, Art. no. 436.
[30] J. Liu et al., “Artificial intelligence in the 21st century,” IEEE Access,vol. 6, pp. 34403–34421, 2018.
[31] A. Sandryhaila and J. M. Moura, “Discrete signal processing on graphs,” IEEE Trans. Signal Process., vol. 61, no. 7, pp. 1644–1656,Apr. 2013.
[32] D. Shuman, S. Narang, P. Frossard, A. Ortega, and P. Vandergheynst,“The emerging field of signal processing on graphs: Extending highdimensional data analysis to networks and other irregular domains,” IEEE Signal Process. Mag., vol. 3, no. 30, pp. 83–98, May 2013.
[33] M. Puschel and J. M. Moura, “Algebraic signal processing theory: Foundation and 1-D time,” IEEE Trans. Signal Process., vol. 56, no. 8,pp. 3572–3585, Aug. 2008.
[34] A. Sandryhaila and J. M. Moura, “Discrete signal processing on graphs:Graph filters,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process.,2013, pp. 6163–6166.
[35] S. Chen, R. Varma, A. Sandryhaila, and J. Kovaˇcevi´c, “Discrete signal processing on graphs: Sampling theory,” IEEE Trans. Signal Process., vol. 63, no. 24, pp. 6510–6523, Dec. 2015.
[36] U. V. Luxburg, “A tutorial on spectral clustering,” Statist. Comput.,vol. 17, no. 4, pp. 395–416, Dec. 2007.
[37] X. Zhu and M. Rabbat, “Graph spectral compressed sensing for sensor networks,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process.,2012, pp. 2865–2868.
[38] A. Gavili and X.-P. Zhang, “On the shift operator, graph frequency, and optimal filtering in graph signal processing,” IEEE Trans. Signal Process.,vol. 65, no. 23, pp. 6303–6318, Dec. 2017.
[39] B. Pasdeloup, M. Rabbat, V. Gripon, D. Pastor, and G. Mercier, “Graph reconstruction from the observation of diffused signals,” in Proc. 53rd Annu. Allerton Conf. Commun., Control, Comput., 2015, pp. 1386–1390.

[40] A. Anis, A. Gadde, and A. Ortega, “Towards a sampling theorem for signals on arbitrary graphs,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2014, pp. 3864–3868.
[41] G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst, “Random sampling of bandlimited signals on graphs,” Appl. Comput. Harmon. Anal., vol. 44, no. 2, pp. 446–475, Mar. 2018.
[42] H. Shomorony and A. S. Avestimehr, “Sampling large data on graphs,” in Proc. IEEE Glob. Conf. Signal Inf. Process., 2014, pp. 933–936.
[43] L. F. Chamon and A. Ribeiro, “Greedy sampling of graph signals,” IEEE Trans. Signal Process., vol. 66, no. 1, pp. 34–47, Jan. 2018.
[44] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, “Sampling of graph signals with successive local aggregations,” IEEE Trans. Signal Process., vol. 64, no. 7, pp. 1832–1843, Apr. 2016.
[45] S. K. Narang, A. Gadde, and A. Ortega, “Signal processing techniques for interpolation in graph structured data,” in Proc. IEEE Int. Conf. Acoust.,Speech Signal Process., 2013, pp. 5445–5449.
[46] A. Gadde and A. Ortega, “A probabilistic interpretation of sampling theory of graph signals,” in Proc. IEEEInt. Conf.Acoustics, Speech Signal Process., 2015, pp. 3257–3261.
[47] X. Wang, M. Wang, and Y. Gu, “A distributed tracking algorithm for reconstruction of graph signals,” IEEE J. Sel. Topics Signal Process.,vol. 9, no. 4, pp. 728–740, Jun. 2015.
[48] P. Di Lorenzo, S. Barbarossa, P. Banelli, and S. Sardellitti, “Adaptive least mean squares estimation of graph signals,” IEEE Trans. Signal Inf. Process. Netw., vol. 2, no. 4, pp. 555–568, Dec. 2016.
[49] D. Romero,M.Ma, andG.B.Giannakis, “Kernel-based reconstruction of graph signals.” IEEE Trans. Signal Process., vol. 65, no. 3, pp. 764–778, Feb. 2017.
[50] M. Nagahara, “Discrete signal reconstruction by sum of absolute values,”IEEE Signal Process. Lett., vol. 22, no. 10, pp. 1575–1579, Oct. 2015.
[51] S. Chen, R. Varma, A. Singh, and J. Kovaˇcevi´c, “Signal representations on graphs: Tools and applications,”2015, arXiv:1512.05406.
[52] S. Segarra, A. G. Marques, G. Leus, and A. Ribeiro, “Reconstruction of graph signals through percolation from seeding nodes,” IEEE Trans. Signal Process., vol. 64, no. 16, pp. 4363–4378, Aug. 2016.
[53] F. Xia, J. Liu, J. Ren, W. Wang, and X. Kong, “Turing number: How far are you to a. m. turing award?” ACM SIGWEB Newslett., vol. Autumn, pp. 1–5, Nov. 2020.
[54] H. E. Egilmez, E. Pavez, and A. Ortega, “Graph learning from data under Laplacian and structural constraints,” IEEE J. Sel. Topics Signal Process., vol. 11, no. 6, pp. 825–841, Sep. 2017.
[55] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, “Learning Laplacian matrix in smooth graph signal representations,” IEEE Trans.Signal Process., vol. 64, no. 23, pp. 6160–6173, Aug. 2016.
[56] V. Kalofolias, “How to learn a graph from smooth signals,” in Proc. Artif. Intell. Statist., 2016, pp. 920–929.
[57] E. Pavez and A. Ortega, “Generalized Laplacian precision matrix estimation for graph signal processing,” in Proc. IEEE Int. Conf. Acoust.,Speech Signal Process., 2016, pp. 6350–6354.
[58] E. Pavez, H. E. Egilmez, andA.Ortega, “Learning graphs with monotone topology properties and multiple connected components,” IEEE Trans. Signal Process., vol. 66, no. 9,

pp. 2399–2413, May 2018.
[59] B. Pasdeloup,V. Gripon, G. Mercier,D. Pastor, andM.G. Rabbat, “Characterization and inference of graph diffusion processes from observations of stationary signals,” IEEE Trans. Signal Inf. Process. Netw., vol. 4, no. 3,pp. 481–496, Sep. 2018.
[60] S. Segarra,A.G.Marques, G. Mateos, andA.Ribeiro, “Network topology inference from spectral templates,” IEEE Trans. Signal Inf. Process. Netw., vol. 3, no. 3, pp. 467–483, Sep. 2017.
[61] D. Thanou, X. Dong, D. Kressner, and P. Frossard, “Learning heat diffusion graphs,” IEEE Trans. Signal Inf. Process. Netw., vol. 3, no. 3,pp. 484–499, Jul. 2017.
[62] J. Mei and J. M. Moura, “Signal processing on graphs: Causal modelling of unstructured data,” IEEE Trans. Signal Process., vol. 65, no. 8, pp. 2077–2092, Apr. 2017.
[63] S. Segarra, G. Mateos, A. G. Marques, and A. Ribeiro, “Blind identification of graph filters,” IEEE Trans. Signal Process., vol. 65, no. 5, pp. 1146–1159, Mar. 2017.
[64] F. Xia, N. Y. Asabere, A. M. Ahmed, J. Li, and X. Kong, “Mobile multimedia recommendation in smart communities: A survey,” IEEE Access, vol. 1, no. 1, pp. 606–624, Sep. 2013.
[65] W. Huang, A. G. Marques, and A. R. Ribeiro, “Rating prediction via graph signal processing,” IEEE Trans. Signal Process., vol. 66, no. 19, pp. 5066–5081, Oct. 2018.
[66] F. Xia, H. Liu, I. Lee, and L. Cao, “Scientific article recommendation: Exploiting common author relations and historical preferences,” *IEEE Trans. Big Data*, vol. 2, no. 2, pp. 101–112, Jun. 2016.