Graph Neural Networks For Affective Social Media: A Comprehensive Overview

Michail Karavokyris*, Spyros Sioutas

Computer Engineering and Informatics Department, University of Patras, Patras 26504, Hellas

Abstract

Social media have become the main platforms for expressing and supplementing nuanced human activity such as engaging in public and private conversations, creating and sharing multimedia content, participating to digital culture events, and recently describing emotions about events, places, or even products. In this survey, we provide a comprehensive overview of graph mining and machine learning on affective social media through graph neural networks (GNNs). The latter are capable of performing a variety of tasks, such as graph and vertex classification, link prediction, and graph clustering using vertex information, edge information, and topological structure. These capabilities are critical in harnessing the vast emotional information available in social media in order to generate meaningful and scalable affective analytics.

Keywords

graph neural networks, distributed computation, graph mining, graph convolution, network topology, convergence, link prediction, label prediction, community discovery, affective computing, PyTorch,

1. Introduction

Currently social media are widely considered to be the digital reflection, or even the digital twin in certain cases, of individuals and groups. Among the prime information found in social media are affective indicators such as the emotional polarity of posts or reactions to them. This is especially true in Twitter which abounds with long conversations full with emotionally charged replies [1][2], whereas Facebook [3] and LinkedIn [4][5] have dedicated emotional reaction buttons for each post. Even Instagram contains images which have been reported to elicit emotional responses [6].

Typically, in deep learning applications, such as fraud detection, natural language processing (NLP), biomedical image processing, and computer vision, the datasets are represented as manifolds in the Euclidean space. However, recently the number of engineering scenarios requiring non-Euclidean data and instead rely on graphs has been rising. Therein topological relations and interconnectivity play a major role. Graphs enable the modeling of important problems in various scientific fields including complex systems, social networks, protein-protein interaction networks, logistics and long supply chains, transportation networks, knowledge graphs, and others.

Graph Neural Networks (GNNs) constitute a broad class of neural network architectures depending strongly on information propagation mechanisms such as message passing between graph nodes or attention functions between network layers to encapsulate the higher order communication flow and interplay inherent in graphs. Although their functionality may resemble that of other architectures like the established multilayer perceptrons (MLPs) found in many machine learning (ML) applications, it is fundamentally different mainly because the role of higher order patterns is more intense.

The primary research objective of this conference paper is the presentation of the predominant GNN architectures and their primary properties as well as how they can be applied to basic tasks related to affective social network analysis. This will give the interested reader a brief yet concise view of the research landscape of a field which is the focus of intense interdisciplinary research.

Table 1
Notation Summary

Symbol	Meaning	First in
$\triangleq$	Equality by definition	Eq. (1)
$\dot{\mathbf{x}}$	First vector derivative	Eq. (4)
$\tanh(\cdot)$	Hyperbolic tangent	Eq. (8)
$\deg(v)$	Degree of vertex v	Eq. (1)
$\mathrm{diag}\left[d_{1,1},\dots,d_{n,n}\right]$	Diagonal matrix	Eq. (1)
$\mathbf{I}_n$	$n \times n$ identity matrix	Eq. (2)

The remainder of this work is structured as follows. In section 2 the recent scientific literature regarding GNNs, affective social media, and graph mining is overviewed. Then in section 3 the primary properties of GNNs are enumerated in detail, whereas in section 4 the applications of GNNs to affective social network analysis are presented.

CIKM'22: 31st ACM International Conference on Information and Knowledge Management (companion volume), October 17–21, 2022, Atlanta, GA

*Corresponding author.

karavokyrism@gmail.com (M. Karavokyris); sioutas@ceid.upatras.gr (S. Sioutas)

0000-0002-1263-0785 (M. Karavokyris); 0000-0003-1825-5565 (S. Sioutas)

Future research directions are given in section 5. Capital boldface letters denote matrices, small boldface vectors, and normal small scalars. Acronyms are explained the first time they are encountered in the text. Additionally, the terms *vertex* and *node* are used interchangeably in this work. The same holds true for the terms *edge* and *link*. In function definitions parameters follow the respective arguments after a semicolon. Finally, in table 1 the notation used in this work is summarized.

2. Related Work

As stated earlier GNNs are neural networks tailored for natively handling graphs or any kind of linked data for that matter [7]. Techniques for doing so include graph embedding [8], message passing [9], and attention mechanisms [10], the latter primarily in the form of graph attention networks (GATs) [11]. The current state of the art in GNNs allows them to perform link prediction [12], graph convolution [13], semi-supervised [14] and unsupervised [15] graph clustering, and node classification [16]. Regarding applications, GNNs have been used to evaluate the affective coherence of ordinary [17] and fuzzy [18] Twitter graphs, to perform content filtering [19], to yield social recommendations [20], to compute recommendations at large scale systems [21], to perform image classification [22], to do vertex classification based on their susceptibility in SIS-type propagation models [23], for fake news discovery [24], and for rumor tracing [25]. Comprehensive field reviews regarding GNNs can be found in [26] and also in [27].

Neural network architectures are ubiquitous in ML [28, 29], especially in conjunction with low rank tensor approximation [30], and signal processing [31]. Bayesian neural networks stem directly from non-classical signal estimation theory [32]. Convolutional neural networks (CNNs) are extensively used in image processing [33]. Recently deep neural networks have been trained to obey physical laws [34]. In [35] a sequence of social graphs is compressed with the two dimensional discrete cosine transform (DCT2) but expanded with a tensor stack network (TSN) trained with information from the entire sequence. Moreover, TSNs have been used for sound classification [36] and large scale urban network speed prediction [37]. Self organizing maps (SOMs) for cultural content recommendation are described in [38]. Recent and extensive reviews on neural network architectures include [39] and [40], where an extended and neural network taxonomy is described as well.

Graph mining aims at locating and extracting latent and non-trivial knowledge from graphs such as cycle lengths in massive graphs [41], higher order spatiotemporal patterns [42], and triangles [43]. Techniques include employing intelligent agents for autonomous mining [44], approximating directed graphs with undirected ones based on enegry criteria [45], managing graph streams with relational algebra [46], computing graph topological correlation [47], efficiently inferring graph isomorphism [48] and performing generic pattern search [49] with GNNs on graphs, and massive graph visualization with feedback for graph matching [50]. Applications of graph mining include among others co-author recommendation [51], efficient new drug discovery [52], consensus protocols in blockchains [53], and energy management in smart power grids [54]. Other considerations include fairness [55], explainability and automation [48], and application to the emerging field microservices [56].

Social network analysis, although it relies heavily on graph mining [57], it is a distinct field since it also focuses on social media functionality [58], which includes posts [59], conversations [60], and even digital trust as a conditional extension of the one found in the real world [61, 62]. Moreover, psychological aspects such as self-esteem [63] and cognitive ones like consumer engagement and online time [64] play a central role. Among the numerous social media applications can be found stock market trend prediction [65], the acceleration under suitable conditions of open innovation [66], the selection database architecture according to social queries regarding Twitter account influence [67], the alteration of the value of NFTs depending on the Twitter influence of the respective holder [68], and the data-driven deployment of digital marketing [69]. Reviews of the field include [70] which places special emphasis on community structure discovery, [71] which explores the dynamics of academic social networks and online communities, and [72] where collaborative innovation processes are explored.

3. Graph Neural Networks

3.1. Overview

In this section first the most frequent tasks performed by the GNN architectures are described. Then, the most prominent GNN types and their properties are presented.

3.2. GNN Tasks

Typically, every application for affective social media fits into one of the following basic tasks:

- **Node classification:** The goal is to predict missing node labels in a social network using the labels of the neighbor nodes. For example, the emotional state of a user can be predicted as a function of the attributes of that user and of its neighbours.
- **Link prediction:** In this scenario the objective is to predict the link between various entities in a

network by utilizing a partial or otherwise incomplete adjacency matrix. This task is frequently used in social network settings because it can predict whether any two vertices, which may well be accounts, pages, or even entire communities, are likely to be connected. Moreover, in certain cases and depending on the available features, the strength of this link may be estimated as well.
- **Community detection:** The case here is to allocate nodes into clusters whose size is unknown beforehand, namely it is a clustering problem. This can be done by partitioning the vertex sex based on edge features like weights or, alternatively, by viewing the nodes as items and by grouping together items with comparable properties. For instance, community detection can be used on affective social media analysis to locate communities with similar emotional characteristics.

3.3. Architectures

GNNs constitute a class of neural networks based on the dependence between the elements of the graph. The term GNN does not refer to a single algorithm or architecture but rather to a plethora of distinct algorithms. The common denominator for each GNN is the ability to exploit the information inherent in graph topology in order to compute a global steady state. This is more evident in the message passing architectures, but this can also be seen in some other of the most common GNN architectures that have been developed in recent years like graph convolutional networks (GCNs) and graph attention networks (GATs). In table 2 the architectures examined here and their main properties are presented.

3.3.1. Graph Convolutional Networks

Graph convolutional networks (GCNs), which seek to imitate the functionality of ordinary CNNs, are currently the prime candidate architectures for most real life applications. Specifically, the main idea behind GCNs is to adapt CNNs to natively handle linked data, namely graphs. CNNs in order to create highly expressive representations can extract multiscale localized spatial information and combine it in order to yield the final result. In this sense, they exploit the higher order patterns inherent in graphs. Since CNNs are able to capture meaningful features across the entire data sets, GCNs adjust the operation of convolution from grid data to graph data. Graph convolution uses the features of the neighbors of a given node to make predictions by transforming the features of a node in a latent space. The objective for these models is to train a function of features on a graph where the input is a set of nodes and edges which are described from a feature vector that contains their attributes.

There are two different types of graph convolution operations, which in turn determine the domain a given GCN is defined on:

- **Spatial convolution:** These GCNs operate directly on the graph adjacency matrix as if were a grid but with additional constraints. Thus, convolution is performed in a way similar to images by using spatial features learned from the graph. This is the equivalent to the time domain filtering.
- **Spectral convolution:** These GCNs utilize the eigendecomposition of the graph Laplacian matrix in order to propagate information across nodes. Therefore, processing takes place in the two-dimensional spatial frequency domain akin to the transform domain adaptive algorithms.

Recall that the graph Laplacian of equation (2) can be defined based on the graph degree matrix of equation (1). Observe that nodes of zero degree essentially do not contribute to the overall graph structure and thus are considered to have been removed during a preprocessing stage. Therefore, matrix $\mathbf{D}$ is always invertible.

$$\mathbf{D} \triangleq \operatorname{diag}\left[\deg(v_1), \dots, \deg(v_n)\right] \tag{1}$$

With this knowledge the graph Laplacian matrix can then be constructed from the respective adjacency matrix $\mathbf{A}$ as shown in equation (2). The eigenexpansion of $\mathbf{L}$ is the graph spectrum on the corresponding basis.

$$\mathbf{L} \triangleq \mathbf{I}_n - \mathbf{D}^{-1}\mathbf{A} \tag{2}$$

Although spectral GCNs can construct powerful graph representations and act as convolutional filters for graph classification with considerable accuracy, they fail to utilize feature locality commonly found in most graphs. Additionally, spectral GCNs come with great computational cost, especially for large networks.

In order to address the issues of locality and computational complexity, ChebNets were developed in order to combine CNNs with the spectral networks theory. Thus, in ChebNets the representation of any feature vector should only be influenced by the k-hop neighbors. Therefore, ChebNets provide the essential algorithmic foundation and effective schemes since the convolution is computed using Chebyshev polynomials instead of the eigenvectors of the Laplacian matrix. Therefore, spectral GCNs can be considered as ChebNets where the neighborhood depth equals one. The objective of this model is to learn a function of features which operates on a graph G represented as in equation (3):

$$G \triangleq (V, E) \tag{3}$$

Specifically, a ChebNet is designed to build an $N \times F$ output matrix where F is the number of output attributes

and N is the number of vertices. Said matrix is iteratively constructed given the following graph input.

- Feature description vectors, one for each of the N nodes, are stacked are form a $N \times D$ feature matrix where D denotes the number of features.
- The $N \times N$ graph adjacency matrix. Therein are contained all local patterns and its powers encode all higher order ones.

Each network layer has a nonlinear function which acts as the ChebNet propagation rule. Based in the choice of the propagation rule and the numbers this is successively applied models may vary. The most common propagation rule is ReLU operating on a linear combination of the outputs of the previous layers. The features processed at each layer are aggregated to form the attributes of the following layer. This implies that each node in the k-th layer will collect information from their k-hop neighbors. It has been observed that a small number of layers, typically at most four, suffices.

Since in this model the aggregated representation of each vertex includes only local features, namely those of its neighbors, this has to be taken into consideration in the structure of the adjacency matrix. This is done in two ways, by adding the identity matrix to it to allow the construction of its powers and also by normalizing it similarly to the graph Laplacian of (2). So when GCNs and ChebNets are trained by stochastic gradient descent algorithms, which tend to be sensitive to the scale of input features, there are no vanishing or exploding gradients which frequently delay or even derail training.

It should be also mentioned that GCNs are mainly used for semi-supervised node classification, whether binary or multi-class by adding a softmax layer at the end. Also by combining graph convolution layers with graph pooling layers the GCN model will be able to predict the class labels for an entire graph.

3.3.2. Graph Attention Networks

Analogous to GCNs, GATs average hidden attributes on a local level. But unlike GCN, which compute the propagation weights explicitly during training, GATs define them implicitly. This is accomplished by the *attention mechanism*, namely a learnable function to re-weight synapses between neurons as a function of the values of the hidden features. In this way, the significance of each node can be specified by utilizing more information than the structure of the graph and the connectivity patterns contained in the latter. However, this local aggregation has to be eventually compensated for when values are propagated to other layers and this is in fact one of the factors differentiating GATs.

In particular, the synaptic weights are computed as a result of an attention mechanism which computes the normalised coefficients from the unnormalized ones. Typically, the softmax function is the key to normalizing these coefficients as it can convert a set of raw scores to an exponentially weighted distribution.

3.3.3. Message Passing Neural Networks

Message passing neural networks (MPNNs) are decentralized architectures which rely heavily on message passing in order to perform a given computation. Such communication may take place synchronously or asynchronously. Each node starts with a local ground truth vector and progressively based on input from neighboring vertices evolves into a steady state vector. Although initially the information exchanged between vertices may be inaccurate, this is remedied at later stages, provided the update mechanisms are designed to do so. This is by no means a trivial task as essentially this is a decentralized nonlinear control problem. Therefore, extended care must be taken beforehand in order to avoid effects such as Witsenhausen's counterexample [73].

In contrast to other neural network architectures, MPNNs have a flat architecture in the sense that there are no layers. This implies that the diameter of the network plays a crucial role as it represents the maximum amount of time, measured in the number of hops, which is necessary in order for a given piece of information to be transmitted across the MPNN. Related metrics such as the effective diameter reveal the links necessary for a considerable segment of the graph to be reached. Strong locality, expressed in the number of triangles or equivalently in the clustering coefficient, contributes to quick propagation. On the contrary, bridges may be congestion points. In any case, topology is central in MPNNs and its effects are more intense compared to other GNN types.

In table 2 are listed some of the most representative convergence schemes proposed in the bibliography.

Table 2
GNN Architectures

GNN architecture	Description
Message passing	Communication with messages
Graph convolution	Aggregation of hidden features
ChebNet	Aggregation of attributes
Graph attention	Self attention mechanism

3.4. Convergence

3.4.1. State Vectors

Convergence is a major topic since GNNs are distributed and, hence, there is not a single point of centralized control. As such, various techniques based on traditional control equations such as those describing continuous,

linear, and time invariant systems as in equation (4) do not directly apply. Therein **A** is the system plant, **b** is the input distribution vector, and **x** is the state vector.

$$\dot{\mathbf{x}} \triangleq \mathbf{A}\mathbf{x} + \mathbf{b}u, \quad \mathbf{A} \in \mathbb{R}^{n\times n}, \mathbf{b} \in \mathbb{R}^{n\times 1} \tag{4}$$

In equation (4) $\dot{\mathbf{x}}$ is defined as the column vector containing the first time derivatives of the control variables $x[1]$ to $x[n]$ as shown in equation (5). The selection of these variables essentially determines the graph model.

$$\dot{\mathbf{x}} \triangleq \left[\frac{\partial x[1]}{\partial t} \quad \frac{\partial x[2]}{\partial t} \quad \dots \quad \frac{\partial x[n]}{\partial t}\right]^T \in \mathbb{R}^{n\times 1} \tag{5}$$

Another control model based also on the concept of the state vector which is more general but at the same time less tractable is that of the nonlinear control model of equation (6). In the latter $f(\cdot)$ is a nonlinear differentiable vector valued function codifying network dynamics.

$$\dot{\mathbf{x}} \triangleq f(\mathbf{x}, u), \qquad f\colon \mathbb{R}^{(n+1)\times 1} \to \mathbb{R}^{n\times 1} \tag{6}$$

Although the nonlinear control model of (6) covers more cases than that of its linear counterpart of (4), there are less analytical tools to explore and handle it. Moreover, many control related results depend heavily on the properties of $f(\cdot)$. On the contrary, the control model of (4) is appealing for a number of reasons including tractability and explainability. To this end, often many instances of (6) are linearlized with various methods to a time varying version of equation (4) where the properties of the latter hold true locally.

3.4.2. Brower's Fixed Point Lemma

For most message passing architectures an alternative methodology to monitor convergence lies in the Brower's fixed point lemma (BFPL). The latter states that any continuous function $f(\cdot)$ mapping any interval I_0 to itself has at least one fixed point $s_0 \in I_0$ as shown in (7).

$$s_0 = f(s_0), \qquad f : I_0 \to I_0 \tag{7}$$

The existence of the fixed point s_0 guarantees that the MPNN cannot escape from it and as such it is in one of the potentially many steady states. However, that requires that a significant number of neurons reach that state before they start propagating it to their neighbors. Moreover, methodologies based on the BFPL are considered to be indirect in the sense that they monitor the output of each node s and not their internal state vector **s** as before. Therefore, the global convergence is tracked through individual vertices. Still, they have been applied successfully, especially when the processing involves smooth functions, in cases where the local computation is yields a single scalar. For instance, BFPL has been applied to MPNNs which employ with proper scaling the sigmoid or hyperbolic function as activation function as shown in equation (8).

$$\varphi(s;\alpha_0,\beta_0) \triangleq \alpha_0 \tanh(\beta_0 s) \triangleq \alpha_0 \frac{e^{\beta_0 s} - e^{-\beta_0 s}}{e^{\beta_0 s} + e^{-\beta_0 s}} \tag{8}$$

As stated earlier, topology plays a central role in convergence, since it determines the average and maximum rate of spatial information propagation in terms of the number of links between any two processing vertices.

In table 3 are listed some of the most representative convergence schemes proposed in the bibliography.

Table 3
Graph Neural Network Convergence Criteria

Type	Description
BFPL	Based on continuous maps
State convergence	Aggregation of local convergence

3.5. Learning Tasks

Irrespective of their architectural classification GNNs are called to perform the following fundamental algorithmic tasks across a broad spectrum of applications. These include discovering graph community structure, setting a message passing mechanism, performing vertex classification, and doing graph convolution.

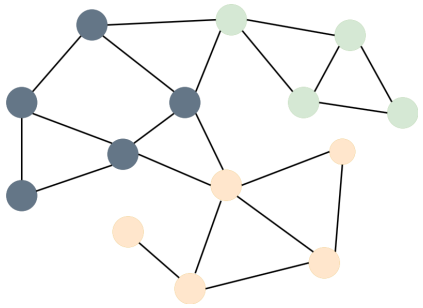

Figure 1: Graph community discovery.

Graph community structure discovery is paramount in graph mining as it reveals latent dynamics as shown in figure 1. Still, in the scientific literature there is more than one definition of what makes a community as this may well depend on the semantics of the underlying domain. For instance, graphs may be weighted, signed, or undirected. Each such property adds constraints to community discovery. Moreover, since this task relies on higher order patterns, it is also computationally chal-

lenging. Consequently, a number of diverse heuristics have been developed for it.

Message passing mechanisms are crucial in most engineering scenarios involving graphs, even indirectly since most networks are set up in order to achieve coherency and communication. Especially in MPNNs selecting the attributes represented in the ground truth vector of each vertex is of paramount importance since that determines what is exchanged during communication. A static snapshot of message passing is shown in figure 2.

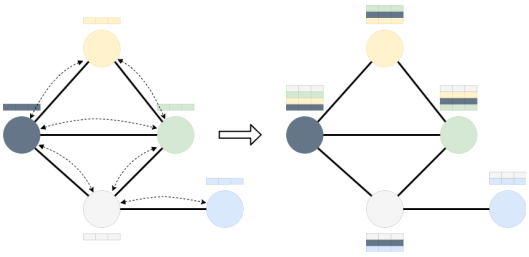

Figure 2: GNN message passing.

Node classification is another important task where each vertex is assigned one out of many possible labels drawn out of a finite label set based on a decision rule. This functionality is shown in figure 3. Labels may be repeated and, depending on the problem, some vertices may already have a label. Moreover, this task has close ties with the community discovery task, although in classification nonadjacent nodes may have the same label. More recently ML models which can utilize structural and functional attributes, whenever the latter are available, have been proposed in the literature. It should be noted though that functional features depend heavily on the underlying domain, whereas structural attributes can be applied to any scenario.

Graph convolution is an operation involving a pair of graphs and yields a larger one whose topology depends on theirs. This allows the efficient discovery of local patterns and, depending on how convolution is defined, even their variants or incomplete ones. This operation initially appeared in the field of computer vision and has found numerous applications in social media analysis and ML. Figure 4 shows an instance of this operation.

Finally in figure 5 the task of link prediction task is shown. It is an important task where given a partial graph or an evolving one and a decision rule must be devised which can predict whether a link between any

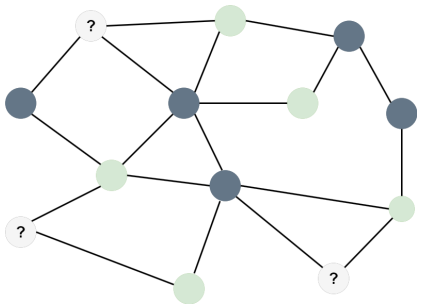

Figure 3: Node classification.

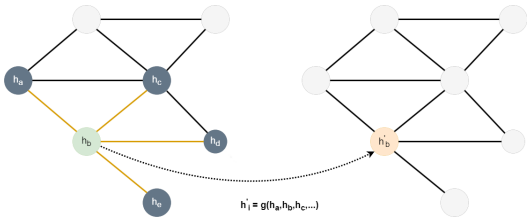

Figure 4: Graph convolution.

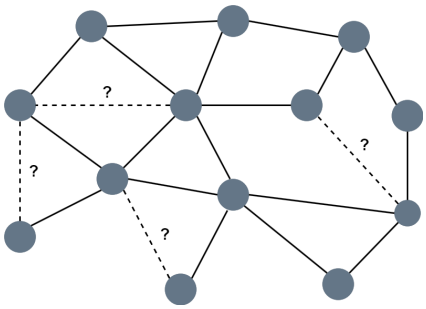

Figure 5: Link prediction.

two given nodes exists. In order to determine whether such link should be added to the graph, a segment of the graph considered as ground truth is used along with the assumption that scale free graphs exhibit self-similarity in many levels. Alternatively, state vectors in every vertex or structural patterns may be used to train an ML model. Either case may require a considerable amount

Table 4
Computational Tasks For Each Affective Computing Task

Affective task	Computational tasks
Node affective state	Graph attention, node classification
Edge emotional potential	Node classification, message passing, graph attention
Post emotional potential	Node classification, link prediction, graph convolution
Node affective influence	Message passing, link prediction, node classification
Affective communities	Community discovery, node classification, link prediction

of computational resources, depending on the algorithm.

4. Affective Social Media Analysis

Affective computing is a recent field which extends the existing knowledge in social network analysis with emotional attributes and their study. It has already bore fruits [5, 4] and its prospects look bright with the advent of sophisticated DL techniques such as the GNN architectures described earlier but also like autoencoders, graph adversarial networks (GANs), and CNNs. All these models operate on a plethora of affective attributes including among others word length and polarity, number of sentences, use of punctuation, mentions, and words having special meaning such as modifiers, negations, and of considerable emotional weight.

As stated above, affective social media analysis places emphasis on the emotional state of social media accounts through their posts as well as through the interactions between them. The methodologies most commonly found in the scientific literature can be broadly divided into the following categories. Furthermore, in table 4 is shown how each of the affective applications presented in this section can take advantage of the potential offered by the learning tasks of GNNs.

The determination of the affective state of a node or a group of nodes is paramount as it allows, among others, for locating potential starting points for various online digital campaigns with political, commercial, or social topics. Moreover, it determines which sort or messages are appropriate for a given node given its affective state. To this end, a number of node classification techniques or, more recently graph attention-based mechanisms, can be applied. Given the phenomenon of *homophily* in social media stating that nodes with similar behavior eventually tend to connect with each other, the neighborhood of the vertex under consideration may as well provide additional affective attributes.

In a sense the dual problem of the above is finding out the affective potential of an edge as the latter is primarily a function of the affective state of its endpoints. However, since links in a network may accommodate other communication needs, for instance that of the respective communities in case of a bridge, it also depends on its functionality. As such, in addition to node classification and graph attention analysis pertaining to message passing should be employed.

Tracing the emotional effect of a post is more challenging since a number of interconnected instances of the previous problem should be studied as a post propagates through a graph. Moreover, possible variations of or intentional modification to the latter should be also taken into consideration as well as the overall information context of the adjacent edges and vertices. Consequently, the entire route of a post should be analyzed in this case using graph convolutions and node classification, whereas certain propagation patterns of important posts may be explained with link prediction techniques.

The affective influence of a node can be considered as a generalization of a potentially nonlinear combination of determining the emotional state of a number of vertices with evaluating the impact of the posts of the node under consideration. This happens as influence is frequently taken to be a function of the topological properties of its high order neighborhood and of the emotional potential of its post. In order to evaluate said affective influence, node classification techniques, message passing, and link prediction are frequently employed.

Finally, affective community discovery is perhaps the most challenging of the tasks commonly encountered in affective social media analysis since it entails the computation of various higher order influence metrics. Therefore, a considerable portion of or even the entire graph topology and, depending on the problem perhaps the associated functionality, must be factored in. However, a far more accurate insight into the total network dynamics is obtained. Therefore, approximate analysis of an evolving network for a number of steps can take place before such a computation can be performed again.

5. Conclusions

This conference paper focuses on a comprehensive presentation of a large number of graph neural network architectures tailored for performing affective analysis on social media. The latter abound with heterogeneous

human emotional information coming from sources so diverse as text, music, images, and even direct emotional markings. Therefore, there is more than sufficient space in social networks to develop information processing strategies aiming at deducing numerous affective attributes such as word and sentence emotional polarities Such attributes are critical in applications such as political or commercial digital campaigns or even in assisting professionals in timely diagnosing mental illness.

Regarding future research directions, more affective applications of GNNs can be explored. Moreover, new GNN architectures may be better suited for the tasks presented here.

Acknowledgments

This conference paper is part of Project 451, a long term research initiative with a primary objective of developing novel, scalable, numerically stable, and interpretable higher order analytics.

References

[1] S. Rani, A. K. Bashir, A. Alhudhaif, D. Koundal, E. S. Gunduz, et al., An efficient CNN-LSTM model for sentiment detection in #BlackLivesMatter, Expert Systems with Applications 193 (2022).

[2] W. Luo, W. Zhang, Y. Zhao, A survey of transformer and GNN for aspect-based sentiment analysis, in: CISAI, IEEE, 2021, pp. 353–357.

[3] A. Rodriguez, Y.-L. Chen, C. Argueta, FADOHS: Framework for detection and integration of unstructured data of hate speech on facebook using sentiment and emotion analysis, IEEE Access 10 (2022) 22400–22419.

[4] M. Bossetta, R. Schmøkel, Cross-platform emotions and audience engagement in social media political campaigning: Comparing candidates' Facebook and Instagram images in the 2020 US election, Political Communication (2022) 1–21.

[5] V. Ahire, S. Borse, Emotion detection from social media using machine learning techniques: A survey, in: Applied Information Processing Systems, Springer, 2022, pp. 83–92.

[6] S. E. McComb, J. S. Mills, Young women's body image following upwards comparison to Instagram models: The role of physical appearance perfectionism and cognitive emotion regulation, Body Image 38 (2021) 49–62.

[7] B. Sanchez-Lengeling, E. Reif, A. Pearce, A. B. Wiltschko, A gentle introduction to graph neural networks, Distill 6 (2021) e33.

[8] S. Kumar, A. Mallik, A. Khetarpal, B. Panda, Influence maximization in social networks using graph embedding and graph neural network, Information Sciences 607 (2022) 1617–1636.

[9] F. Gao, J. Zhang, Y. Zhang, Neural enhanced dynamic message passing, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2022, pp. 10471–10482.

[10] S. Miao, M. Liu, P. Li, Interpretable and generalizable graph learning via stochastic attention mechanism, in: ICML, PMLR, 2022, pp. 15524–15543.

[11] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, 2017. arXiv:1710.10903.

[12] Y. Long, M. Wu, Y. Liu, Y. Fang, C. K. Kwoh, J. Chen, J. Luo, X. Li, Pre-training graph neural networks for link prediction in biomedical networks, Bioinformatics 38 (2022) 2254–2262.

[13] K. Han, Y. Wang, J. Guo, Y. Tang, E. Wu, Vision GNN: An image is worth graph of nodes, 2022. arXiv:2206.00272.

[14] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016. arXiv:1609.02907.

[15] A. Gupta, P. Matta, B. Pant, Graph neural network: Current state of art, challenges and applications, Materials Today: Proceedings 46 (2021) 10927–10932.

[16] Y. Chen, Y. Zheng, Z. Xu, T. Tang, Z. Tang, J. Chen, Y. Liu, Cross-domain few-shot classification based on lightweight Res2Net and flexible GNN, Knowledge-Based Systems 247 (2022).

[17] G. Drakopoulos, I. Giannoukou, P. Mylonas, S. Sioutas, A graph neural network for assessing the affective coherence of Twitter graphs, in: IEEE Big Data, IEEE, 2020, pp. 3618–3627. doi:10.1109/BigData50022.2020.9378492.

[18] G. Drakopoulos, E. Kafeza, P. Mylonas, S. Sioutas, A graph neural network for fuzzy Twitter graphs, in: G. Cong, M. Ramanath (Eds.), CIKM companion volume, volume 3052, CEUR-WS.org, 2021.

[19] Y. Liu, K. Zeng, H. Wang, X. Song, B. Zhou, Content matters: A GNN-based model combined with text semantics for social network cascade prediction, in: PAKDD, Springer, 2021, pp. 728–740.

[20] Z. Guo, H. Wang, A deep graph neural network-based mechanism for social recommendations, IEEE Transactions on Industrial Informatics 17 (2020) 2776–2783.

[21] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, J. Leskovec, Graph convolutional neural networks for Web-scale recommender systems, in: KDD, ACM, 2018, pp. 974–983.

[22] G. Drakopoulos, I. Giannoukou, P. Mylonas, S. Sioutas, On tensor distances for self organizing maps: Clustering cognitive tasks, in: DEXA, volume 12392 of *Lecture Notes in Computer Sci-*

ence, Springer, 2020, pp. 195–210. doi:10.1007/978-3-030-59051-2_13.

[23] W. Xia, Y. Li, J. Wu, S. Li, Deepis: Susceptibility estimation on social networks, in: WSDM, 2021, pp. 761–769.

[24] A. Benamira, B. Devillers, E. Lesot, A. K. Ray, M. Saadi, F. D. Malliaros, Semi-supervised learning and graph neural networks for fake news detection, in: ASONAM, EEE, 2019, pp. 568–569.

[25] S. Xu, X. Liu, K. Ma, F. Dong, B. Riskhan, S. Xiang, C. Bing, Rumor detection on social media using hierarchically aggregated feature via graph neural networks, Applied Intelligence (2022) 1–14.

[26] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, Graph neural networks: A review of methods and applications, AI Open 1 (2020) 57–81.

[27] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S. Y. Philip, A comprehensive survey on graph neural networks, IEEE Transactions on neural networks and learning systems 32 (2020) 4–24.

[28] R. Zhao, Z. Yang, H. Zheng, Y. Wu, F. Liu, Z. Wu, L. Li, F. Chen, S. Song, J. Zhu, et al., A framework for the general design and computation of hybrid neural networks, Nature communications 13 (2022) 1–12.

[29] K. T. Schütt, M. Gastegger, A. Tkatchenko, K.-R. Müller, R. J. Maurer, Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions, Nature communications 10 (2019) 1–10.

[30] X. Jiang, L. Zhang, L. Qiao, D. Shen, Estimating functional connectivity networks via low-rank tensor approximation with applications to MCI identification, IEEE Transactions on Biomedical Engineering 67 (2019) 1912–1920.

[31] R. Agarwal, L. Melnick, N. Frosst, X. Zhang, B. Lengerich, R. Caruana, G. E. Hinton, Neural additive models: Interpretable machine learning with neural nets, Advances in Neural Information Processing Systems 34 (2021) 4699–4711.

[32] L. V. Jospin, H. Laga, F. Boussaid, W. Buntine, M. Bennamoun, Hands-on Bayesian neural networks – A tutorial for deep learning users, IEEE Computational Intelligence Magazine 17 (2022) 29–48.

[33] J. Guo, K. Han, H. Wu, Y. Tang, X. Chen, Y. Wang, C. Xu, CMT: Convolutional neural networks meet vision transformers, in: CVPR, IEEE/CVF, 2022, pp. 12175–12185.

[34] L. G. Wright, T. Onodera, M. M. Stein, T. Wang, D. T. Schachter, Z. Hu, P. L. McMahon, Deep physical neural networks trained with backpropagation, Nature 601 (2022) 549–555.

[35] G. Drakopoulos, E. Kafeza, P. Mylonas, L. Iliadis, Transform-based graph topology similarity metrics, NCAA 33 (2021) 16363–16375. doi:10.1007/s00521-021-06235-9.

[36] A. Khamparia, D. Gupta, N. G. Nguyen, A. Khanna, B. Pandey, P. Tiwari, Sound classification using convolutional neural network and tensor deep stacking network, IEEE Access 7 (2019) 7717–7727.

[37] L. Zhou, S. Zhang, J. Yu, X. Chen, Spatial–temporal deep tensor neural networks for large-scale urban network speed prediction, IEEE Transactions on Intelligent Transportation Systems 21 (2019) 3718–3729.

[38] G. Drakopoulos, I. Giannoukou, S. Sioutas, P. Mylonas, Self organizing maps for cultural content delivery, NCAA (2022). doi:10.1007/s00521-022-07376-1.

[39] H. Song, M. Kim, D. Park, Y. Shin, J.-G. Lee, Learning from noisy labels with deep neural networks: A survey, IEEE Transactions on Neural Networks and Learning Systems (2022).

[40] H. Yuan, H. Yu, S. Gui, S. Ji, Explainability in graph neural networks: A taxonomic survey, IEEE Transactions on Pattern Analysis and Machine Intelligence (2022).

[41] L. Torres, P. Suárez-Serrato, T. Eliassi-Rad, Non-backtracking cycles: Length spectrum theory and graph mining applications, Applied Network Science 4 (2019) 1–35.

[42] Y. Gao, X. Li, J. Li, Y. Gao, N. Guo, Graph mining-based trust evaluation mechanism with multidimensional features for large-scale heterogeneous threat intelligence, in: IEEE International Conference on Big Data, IEEE, 2018, pp. 1272–1277.

[43] M. Koohi Esfahani, P. Kilpatrick, H. Vandierendonck, LOTUS: Locality optimizing triangle counting, in: Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, 2022, pp. 219–233.

[44] M. Yoon, T. Gervet, B. Hooi, C. Faloutsos, Autonomous graph mining algorithm search with best performance trade-off, Knowledge and Information Systems (2022) 1–32.

[45] G. Drakopoulos, E. Kafeza, P. Mylonas, S. Sioutas, Approximate high dimensional graph mining with matrix polar factorization: A Twitter application, in: IEEE Big Data, IEEE, 2021, pp. 4441–4449. doi:10.1109/BigData52589.2021.9671926.

[46] K. Wang, Z. Zuo, J. Thorpe, T. Q. Nguyen, G. H. Xu, RStream: Marrying relational algebra with streaming for efficient graph mining on a single machine, in: OSDI, 2018, pp. 763–782.

[47] G. Drakopoulos, E. Kafeza, One dimensional cross-correlation methods for deterministic and stochastic graph signals with a Twitter application in Julia, in: SEEDA-CECNSM, IEEE, 2020. doi:10.1109/

SEEDA-CECNSM49515.2020.9221815.

[48] Z. Chen, S. Villar, L. Chen, J. Bruna, On the equivalence between graph isomorphism testing and function approximation with GNNs, Advances in neural information processing systems 32 (2019).

[49] W. Jin, Graph mining with graph neural networks, in: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, 2021, pp. 1119–1120.

[50] Q. Zhang, X. Song, Y. Yang, H. Ma, R. Shibasaki, Visual graph mining for graph matching, Computer Vision and Image Understanding 178 (2019) 16–29.

[51] F. Ebrahimi, A. Asemi, A. Nezarat, A. Ko, Developing a mathematical model of the co-author recommender system using graph mining techniques and big data applications, Journal of Big Data 8 (2021) 1–15.

[52] R. S. Olayan, H. Ashoor, V. B. Bajic, DDR: Efficient computational method to predict drug–target interactions using graph mining and machine learning approaches, Bioinformatics 34 (2018) 1164–1173.

[53] G. Drakopoulos, E. Kafeza, H. Al Katheeri, Proof systems in blockchains: A survey, in: SEEDA-CECNSM, IEEE, 2019. doi:10.1109/SEEDA-CECNSM.2019.8908397.

[54] C. Fan, M. Song, F. Xiao, X. Xue, Discovering complex knowledge in massive building operational data using graph mining for building energy management, Energy Procedia 158 (2019) 2481–2487.

[55] J. Kang, J. He, R. Maciejewski, H. Tong, Inform: Individual fairness on graph mining, in: KDD, 2020, pp. 379–389.

[56] W. Lin, M. Ma, D. Pan, P. Wang, Facgraph: Frequent anomaly correlation graph mining for root cause diagnose in micro-service architecture, in: IPCCC, IEEE, 2018, pp. 1–8.

[57] S. Tabassum, F. S. Pereira, S. Fernandes, J. Gama, Social network analysis: An overview, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8 (2018).

[58] F. Baumann, P. Lorenz-Spreen, I. M. Sokolov, M. Starnini, Modeling echo chambers and polarization dynamics in social networks, Physical Review Letters 124 (2020).

[59] R. Urena, G. Kou, Y. Dong, F. Chiclana, E. Herrera-Viedma, A review on trust propagation and opinion dynamics in social networks and group decision making frameworks, Information Sciences 478 (2019) 461–475.

[60] D. Ye, S. Pennisi, Analysing interactions in online discussions through social network analysis, Journal of Computer Assisted Learning 38 (2022) 784–796.

[61] M. Zhang, P. Xu, Y. Ye, Trust in social media brands and perceived media values: A survey study in China, Computers in Human Behavior 127 (2022).

[62] G. Drakopoulos, E. Kafeza, P. Mylonas, H. Al Katheeri, Building trusted startup teams from LinkedIn attributes: A higher order probabilistic analysis, in: ICTAI, IEEE, 2020, pp. 867–874. doi:10.1109/ICTAI50040.2020.00136.

[63] D. P. Cingel, M. C. Carter, H.-V. Krause, Social media and self-esteem, Current Opinion in Psychology (2022).

[64] S. P. Eslami, M. Ghasemaghaei, K. Hassanein, Understanding consumer engagement in social media: The role of product lifecycle, Decision Support Systems 162 (2022).

[65] D. Valle-Cruz, V. Fernandez-Cortez, A. López-Chau, R. Sandoval-Almazán, Does Twitter affect stock market decisions? Financial sentiment analysis during pandemics: A comparative study of the H1N1 and the COVID-19 periods, Cognitive computation 14 (2022) 372–387.

[66] J. R. Saura, D. Palacios-Marqués, D. Ribeiro-Soriano, Exploring the boundaries of open innovation: Evidence from social media mining, Technovation (2022).

[67] M. Marountas, G. Drakopoulos, P. Mylonas, S. Sioutas, Recommending database architectures for social queries: A Twitter case study, in: AIAI, Springer, 2021. doi:10.1007/978-3-030-79150-6_56.

[68] A. Kapoor, D. Guhathakurta, M. Mathur, R. Yadav, M. Gupta, P. Kumaraguru, Tweetboost: Influence of social media on NFT valuation, in: Companion Proceedings of the Web Conference, 2022, pp. 621–629.

[69] B. Nyagadza, Search engine marketing and social media marketing predictive trends, Journal of Digital Media & Policy (2022).

[70] D. Naik, D. Ramesh, A. H. Gandomi, N. B. Gorojanam, Parallel and distributed paradigms for community detection in social networks: A methodological review, Expert Systems with Applications 187 (2022).

[71] X. Kong, Y. Shi, S. Yu, J. Liu, F. Xia, Academic social networks: Modeling, analysis, mining and applications, Journal of Network and Computer Applications 132 (2019) 86–103.

[72] M. B. Dahesh, G. Tabarsa, M. Zandieh, M. Hamidizadeh, Reviewing the intellectual structure and evolution of the innovation systems approach: A social network analysis, Technology in Society 63 (2020).

[73] T. Basar, Variations on the theme of the witsenhausen counterexample, in: Conference on Decision and Control, IEEE, 2008, pp. 1614–1619.