Selin Aviyente and Abdullah Karaaslanli

Explainability in Graph Data Science

Interpretability, replicability, and reproducibility of community detection

©SHUTTERSTOCK.COM/G/TYLERBOYES

Digital Object Identifier 10.1109/MSP.2022.3149471
Date of current version: 28 June 2022

In many modern data science problems, data are represented by a graph (network), e.g., social, biological, and communication networks. Over the past decade, numerous signal processing and machine learning (ML) algorithms have been introduced for analyzing graph structured data. With the growth of interest in graphs and graph-based learning tasks in a variety of applications, there is a need to explore explainability in graph data science. In this article, we aim to approach the issue of explainable graph data science, focusing on one of the most fundamental learning tasks, community detection, as it is usually the first step in extracting information from graphs. A community is a dense subnetwork within a larger network that corresponds to a specific function. Despite the success of different community detection methods on synthetic networks with strong modular structure, much remains unknown about the quality and significance of the outputs of these algorithms when applied to real-world networks with unknown modular structure. Inspired by recent advances in explainable artificial intelligence (AI) and ML, in this article, we present methods and metrics from network science to quantify three different aspects of explainability, i.e., interpretability, replicability, and reproducibility, in the context of community detection.

Introduction

Modern data analysis involves large sets of structured data, where the structure carries critical information about the nature of the data. Typically, graphs are used as mathematical tools to describe the structure of such data. Graphs are ubiquitous in the real world, representing objects and their relationships in varied domains, including social networks, e-commerce networks, biological networks, traffic networks, and brain networks [1]. As a result, numerous signal processing and ML tasks have been extended for analyzing graph structured data, e.g., graph signal processing (GSP), graph topology inference, node classification, link prediction, community detection, and supervised learning with graphs [2]. Among these tasks, community detection is fundamental for uncovering links between structure and function in complex networks. The community detection problem is

Authorized licensed use limited to: Michigan State University. Downloaded on July 01,2023 at 17:48:16 UTC from IEEE Xplore. Restrictions apply.

challenging, in part, because it is not very well posed. For this reason, researchers have proposed a variety of definitions of what constitutes a community and an array of algorithms corresponding to these definitions [3]. While the success of these algorithms has been quantified for synthetic networks with ground truth information, it is harder to evaluate the accuracy, significance, and meaning of the obtained community structure for real networks. For these results to be useful in a variety of scientific and technological studies, there is a need to provide transparency to the community detection algorithms and their outputs.

Over the past decade, the explainability of data-driven methods, e.g., AI and ML, has been a focus of research in the ML and data mining communities. While the ML community is mostly focused on describing how black boxes work, data mining is more interested in explaining the decisions, without even understanding how the opaque decision systems work. Recent survey articles on the topic offer a multitude of terminologies, such as *interpretability*, *accountability*, *responsibility*, *transparency*, *comprehensibility*, *accuracy*, and *understandability*, to evaluate different dimensions of explainability [4], [5]. Along with these different terminologies, a variety of methods, including black-box input–output analysis, sensitivity analysis, saliency maps, attention heat maps, and approximation of the predictions using simple proxy models, have been introduced [4].

Although there has been growing interest in explainable ML models, most of the existing work focuses on explainable predictive models and interpretable neural networks. Thus, the focus has been on making black-box models commonly encountered in deep learning more interpretable and transparent. These methods inherently assume the existence of large-scale labeled training samples. However, in many applications, such as community detection, ground truth data may be missing due to the structural complexity of the data, limits of human knowledge, and significant volumes that complicate the categorization process. Unsupervised learning techniques, including clustering, are commonly used to offer a solution to this lack of ground truth. The interpretability of the clusters is critical in high-impact domains since decision makers need to understand a solution beyond how the data are grouped into clusters: what characterizes a cluster, and how it is different from other clusters?

To date, there has been only a handful of papers that study the issue of explainability for unsupervised learning and, in particular, for clustering methodologies. In [6], the application of interpretable algorithms based on decision tree principles is proposed as a guideline for clustering. Corral et al. [7] develop a system to explain and describe the results of unsupervised learning by looking at the attributes common to most of the points in a cluster. In Explain-It [8], a generic framework for unsupervised and self-explainable learning is introduced. In this framework, clustering results are modeled using a supervised learning model, which is then explained through the application of explainable AI approaches. More recently, interpretable clustering algorithms, such as tree-based approaches [9], and algorithms that try to achieve a balance between the quality and interpretability of clusters by jointly optimizing the homogeneity and quality of clusters [10] have been proposed.

With the growth of interest in graphs and graph-based learning tasks in a variety of applications, there is a need to explore explainability in graph data science. Explainability can be particularly helpful for graphs, even more than for images, as it is harder for nonexpert humans to intuitively determine the relevant context within a graph. Recent work in this area has focused on the explainability of graph neural networks (GNNs) [11]. Several approaches, such as saliency maps, class activation mapping, and excitation backpropagation, have been proposed to explain the predictions of GNNs and provide different levels of explanation. In addition to these efforts, in the area of GSP, algorithm unrolling approaches have been extended to the graph domain to implement an interpretable network for graph signal denoising [12].

In this article, we aim to approach the issue of explainable graph data science, focusing on one of the most fundamental learning tasks, community detection, as it is usually the first step in extracting information from graphs. In ML literature, several descriptive terms have been introduced to define the different aspects of explainability. In this article, we focus on three terms: *interpretability*, *replicability*, and *reproducibility*. In the context of community detection, interpretability relates to the community detection model and the comprehensibility of the resulting communities; replicability relates to the transparency and stability of the algorithm, and reproducibility relates to the scientific consistency of the obtained results. The different aspects of these terms are explored in the following sections.

Dimensions of explainability

In this section, we provide an overview of the three dimensions of explainability: interpretability, replicability, and reproducibility. In the following, these three terms are defined in the context of general ML algorithms.

Interpretability

In the context of ML, interpretability relates to the capability of making sense of a learned ML model. Thus, the aim of interpretability is to map an abstract concept, such as a predicted class label, into a domain that humans can make sense of. Different desired properties, including trust, causality, transferability, and informativeness, have been proposed to evaluate the interpretability of ML models [13], [14]. Coupled with these properties, various techniques have been proposed to confer interpretability. These fall broadly into two categories. The first relates to transparency (i.e., how does the model work?). The second consists of post hoc explanations (i.e., what else can the model tell me?) Transparency is considered at the level of the entire model (simulatability); at the level of individual components, such as parameters (decomposability); and at the level of the training algorithm (algorithmic transparency). Some common approaches to evaluate transparency include approximations using understandable proxy models, such as local linear models and decision trees, and visual approaches, including saliency maps and heat maps, based on sensitivity analysis and relevance scores that help to explain model decisions [15].

Authorized licensed use limited to: Michigan State University. Downloaded on July 01,2023 at 17:48:16 UTC from IEEE Xplore. Restrictions apply.

Post hoc interpretability represents a distinct approach to extracting information from learned models. While post hoc interpretations often do not elucidate precisely how a model works, they may nonetheless confer useful information for practitioners and end users of ML. Some common approaches to post hoc interpretations include natural language explanations, visualizations of learned representations and models, and explanations by example. Visualization methods for the post hoc interpretability of ML models aim to render visualizations in the hope of determining qualitatively what a model has learned. Some examples include t-distributed stochastic neighbor embedding [16] to visualize high-dimensional distributed representations and perturbation-based methods, where the input is altered through gradient descent to enhance the activation of certain nodes in the network [17]. Local explanation approaches focus on explaining what a neural network depends on locally, e.g., saliency maps. Finally, explanation by example focuses on reporting which other examples are most similar with respect to the model [18]. For any example, in addition to generating a prediction, one can use the activations of the hidden layers to identify the k-nearest neighbors based on the proximity in the space learned by the model.

Replicability

In natural sciences, *replicability* refers to measurements being obtained with stated precision by a different team using the same measurement procedure and the same measuring system, under the same operating conditions and in the same or a different location in multiple trials. Based on the standards established by experimental scientists, the Association for Computing Machinery [19] has adopted the following definition for replicability in computational sciences. For computational sciences, including ML, replicability means that an independent group can obtain the same result using the author's own artifacts, e.g., code, data, and so on [19]. Major issues that impede the replicability of ML algorithms include random initializations, hyperparameter selection and tuning, and reliance on benchmark data sets that may not be transferable to real-life data. ML methods use procedures such as resampling and k-fold cross validation to estimate the replicability of a given model and to avoid overfitting.

Reproducibility

Reproducibility refers to the ability of an independent researcher to obtain the same results using the same methodology with his or her own artifacts, e.g., software and data [19]. Goodman et al. [20] propose the following definitions for various types of reproducibility:

- *Method reproducibility*: Provide sufficient detail about procedures and data so that the same processes can be exactly repeated.
- *Results reproducibility*: Obtain the same results from an independent study with procedures as closely matched to the original study as possible.
- *Inferential reproducibility*: Draw qualitatively similar conclusions from either an independent replication of a study or a reanalysis of the original study.

The major challenges regarding reproducibility in ML are a lack of proper documentation of the information, e.g., the code and experimental set up, to reproduce the results and insufficient exploration of the variables that might affect the conclusions of a study. Different statistical tools, such as the intraclass correlation coefficient and coefficient of variation, have been employed to quantify the reproducibility of ML algorithms.

Modularity-based community detection

Background on graphs

In this article, without a loss of generality, we focus on undirected and weighted graphs, as they are commonly used to model different types of networks. Mathematically, a graph $G = (V, E)$ is described by a vertex set $V = \{1, 2, \ldots, n\}$ of cardinality n and an edge set $E \in V \times V$; $\mathbf{A} \in \mathbb{R}^{n \times n}$ denotes the adjacency matrix of a graph, where A_{ij} is nonzero and equal to the weight of the edge between nodes i and j if $(i, j) \in E$. For an undirected graph, $\mathbf{A}$ is symmetric; i.e., $\mathbf{A} = \mathbf{A}^\top$. The degree of node i is defined as $k_i = \Sigma_{j \neq i} A_{ij}$ and quantifies the number of edges connected to node i for binary graphs and the total strength of edges connected to node i for weighted graphs. The degree matrix is a diagonal matrix $\mathbf{D}$, with $D_{ii} = k_i$, and the sum of degrees across all nodes is denoted by $2m$.

The community structure of G can be one of the following: nonoverlapping, overlapping, hierarchical, or local [21]. Numerous methods have been proposed for detecting the community structure of networks [3]. Among these, the most commonly used ones are spectral clustering [22], methods based on statistical inference [23], approaches based on optimization [24], and techniques based on network dynamics [25]. In addition to these classical approaches focusing on nonoverlapping community structure, extensions for overlapping communities have also been considered [26]. In this article, we focus on the explainability of nonoverlapping community detection, which is the partitioning of node set V as $\mathcal{C} = \{C_1, \ldots, C_K\}$, where K is the number of communities. The community assignment vector corresponding to the partition $\mathcal{C}$ is $\mathbf{s} \in \mathbb{R}^n$, where $s_i \in \{1, 2, \ldots, K\}$ denotes the community membership of node i. In particular, we study the explainability of modularity optimization-based methods, as they are data driven and still the most widely used community detection method.

Definition of modularity

The most popular approach to detect communities in graphs relies on the optimization of the modularity function (Q). The modularity function compares the observed pattern of connections in a network against the pattern that would be expected under a specified null model and is calculated as

$$Q = \sum_{i,j} [A_{ij} - P_{ij}] \delta_{s_i s_j}, \tag{1}$$

where P_{ij} is the expected connection between nodes i and j in the null model and $\delta_{s_i s_j}$ is the Kronecker delta function that is equal to one when $s_i = s_j$. Depending on the graph under study, different expressions for P_{ij} can be assumed. The most

Authorized licensed use limited to: Michigan State University. Downloaded on July 01,2023 at 17:48:16 UTC from IEEE Xplore. Restrictions apply.

commonly used null models are the configuration and Erdős–Rényi null models. Further discussion of null models is given in detail in the "Model Explainability" section.

Despite its popularity, modularity is known to suffer from a resolution limit that restricts the size of detectable communities; communities smaller than some size are mathematically indiscernible. To detect communities of all sizes, modularity has been extended to include a resolution parameter, γ, that can be tuned to uncover communities of different sizes [27]:

$$Q(\gamma) = \sum_{i,j} [A_{ij} - \gamma P_{ij}] \delta_{s_i s_j}. \tag{2}$$

By varying the value of γ, one can highlight communities of different sizes; i.e., when γ is big or small, maximizing modularity will return small or large communities, respectively, resulting in a multiscale community structure.

The modularity function can be either positive or negative, with positive values indicating the possible presence of community structure. Thus, one can search for community structure by maximizing the modularity function. An exhaustive optimization of modularity is impossible due to the huge number of ways in which it is feasible to partition a graph. However, there are several algorithms that are fairly good at approximately maximizing modularity in a reasonable time. The most commonly used techniques for modularity optimization are greedy algorithms, simulated annealing, extremal optimization, and spectral clustering [3]. While there are still some questions about the optimality of greedy algorithms [22], the approaches are the most widely used modularity optimization methods for community detection and are the focus of this article.

Greedy algorithms for modularity maximization

One of the most popular greedy algorithms for modularity maximization is the Louvain algorithm [24], due to its low computational complexity. The Louvain algorithm is an agglomerative greedy algorithm consisting of two stages, as depicted in Figure 1. The first stage starts by assigning each node to a different community. Next, the algorithm iterates through each node in a random order to calculate the gain of modularity, ΔQ, that would be obtained if the node were moved from its own community to another. The node is then placed in the community that yields the largest $\Delta Q > 0$. This process iterates over all the nodes until no further improvement of modularity is possible with node movements. In the second stage of the algorithm, all nodes of a community are aggregated to a single "metanode"; with an edge from each metanode to itself, where the weight of the edge is the sum of all the intracommunity edge weights within that community; and an edge between two metanodes, where the weight of the edge is the sum of all intercommunity edge weights among the corresponding communities. These two stages of the algorithm are then repeated until maximum modularity is attained.

Recently, modified versions of the Louvain algorithm, such as the Leiden algorithm, have been proposed to address some of the shortcomings of the Louvain algorithm [28]. In particular, it has been shown that the Louvain algorithm might move a node that acts as a bridge between two components of a community to another community. Moving this node to another community causes the old community to become disconnected. To overcome this problem, the Leiden algorithm applies a refinement stage after the first stage of the Louvain algorithm to ensure that the communities found in the first stage are internally well connected.

Interpretability

In this section, we present the different aspects of interpretability in the context of modularity optimization-based community detection.

Model explainability

An important aspect of interpretable data science is model explainability, which requires an elucidation of the overall model and its components. In the context of modularity-based community detection, model explainability can be divided into

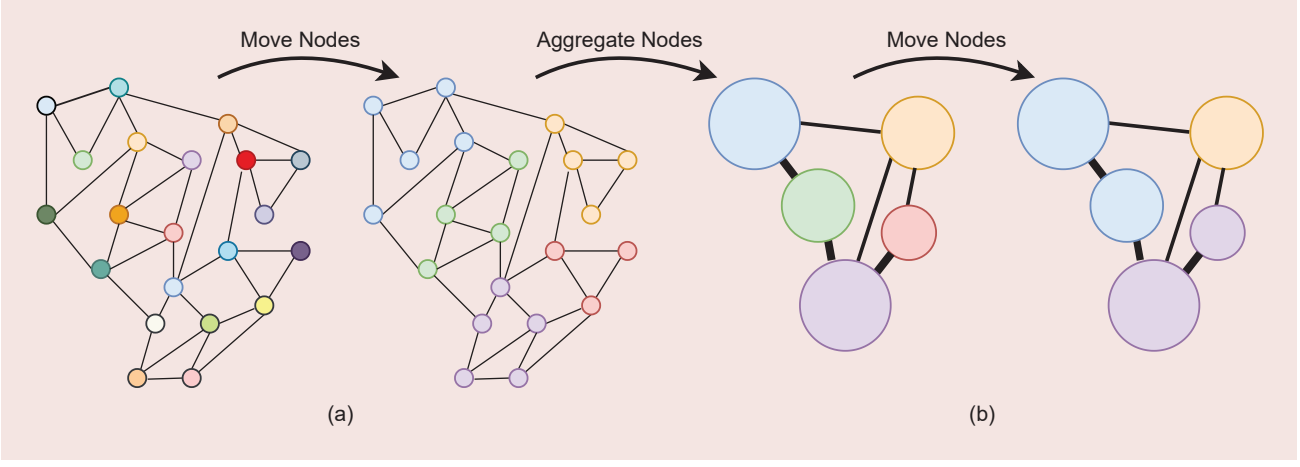


FIGURE 1. The two stages of the Louvain algorithm. (a) In the first, the algorithm passes through each node in a random order and calculates the gain of modularity ΔQ; the nodes are moved to their new communities to obtain the largest ΔQ. (b) In the second, all nodes of a community are merged, and the first stage is repeated.

Authorized licensed use limited to: Michigan State University. Downloaded on July 01,2023 at 17:48:16 UTC from IEEE Xplore. Restrictions apply.

three parts: 1) the physical meaning of the modularity function, 2) the meaning of the selected null model and its interpretation, and 3) the interpretation of the resolution parameter. To bring transparency and interpretation to these three parts, the modularity function can be analyzed from two different perspectives: the dynamic viewpoint of modularity and the relationship between the modularity function and stochastic blockmodels (SBMs).

Meaning of the modularity function

One explanation for the modularity function comes from the dynamic viewpoint of networks. This approach relates community structure to dynamic processes taking place on the network, such as information spread. Random walks are used to model these dynamic processes. A random walk on a graph is defined as a Markov chain, where at each step the random walker arbitrarily jumps from the current node to a neighboring node. In [29], modularity is studied by considering the clustered autocovariance matrix of random walks:

$$\mathbf{R}_t(\mathbf{H}) = \mathbf{H}^\top(\Pi\mathbf{M}^t - \pi^\top\pi)\mathbf{H}, \tag{3}$$

where $\mathbf{H} \in \mathbb{R}^{n\times K}$ is the community membership matrix, $\mathbf{M} \in \mathbb{R}^{n\times n}$ is the transition matrix, and $\Pi \in \mathbb{R}^{n\times n}$ is the diagonal matrix of the stationary distribution of random walk $\pi \in \mathbb{R}^n$; $\mathbf{R}_t \in \mathbb{R}^{K\times K}$ is defined with R_{ij} being the probability of a random walker starting at the ith community and ending at the jth community after taking t steps. If communities are well defined in the sense that they can trap random walks for t steps, then $\mathbf{R}_t$ is approximately diagonal. The Markov stability of the community structure is then defined as $r_t(\mathbf{H}) = \mathrm{tr}(\mathbf{R}_t)$ and can be used to measure the quality of communities. By relating the Markov stability of a community structure to modularity, one can explain the modularity function as follows: $r_1(H)$ is equal to the original modularity function with the configuration model. Thus, the modularity function quantifies the quality of communities, based on whether a random walker can escape a community in one time step.

A second approach to interpret modularity is to relate it to a generative network model. This relationship can be established by showing that maximizing the modularity function is equivalent to maximizing the likelihood of a planted partition version of an SBM/degree-corrected SBM (DCSBM) [30]. SBMs are statistical network models used to generate networks with community structure [31]. In SBMs, each node is assigned to a community $s_i \in \{1,\ldots,K\}$, and nodes i and j are connected with an edge probability of $\theta_{s_is_j}$. A planted partition SBM is a restricted version of an SBM, where $\theta_{s_is_j} = \theta_{\text{in}}$ if $s_i = s_j$ and $\theta_{s_is_j} = \theta_{\text{out}}$, otherwise. For networks with heterogeneous degree distribution, the DCSBM is proposed [23], where edge probabilities are not only determined by $\theta_{s_is_j}$ but also by node degrees.

More formally, an edge between nodes i and j is drawn from a Poisson distribution with mean $\lambda_{ij} = (\theta_{s_is_j}k_ik_j/2m)$. For a planted partition DCSBM, the log likelihood can be written as follows [30]:

$$\log P(G\,|\,\theta,\mathbf{s}) = \frac{1}{2}\sum_{i,j}\left[A_{ij} - \frac{\theta_{\text{in}}-\theta_{\text{out}}}{\log\theta_{\text{in}}-\log\theta_{\text{out}}}\frac{k_ik_j}{2m}\right]\delta_{s_is_j}, \tag{4}$$

where terms that do not depend on the community structure are ignored. This form of the log likelihood function is equal to modularity with the configuration model [see (2)], where the resolution parameter is set as

$$\gamma = \frac{\theta_{\text{in}}-\theta_{\text{out}}}{\log\theta_{\text{in}}-\log\theta_{\text{out}}}. \tag{5}$$

This equivalence between modularity and the planted partition model indicates that the modularity function has the same assumption as an SBM in that the communities are statistically similar; i.e., their intra- and intercommunity edge densities are all proportional to θ_{in} and θ_{out}, respectively.

Null models

The modularity function quantifies the quality of the community structure by comparing intracommunity edge densities in an observed graph to expected intracommunity edge density under a null hypothesis. Two widely used null models are the configuration null model and the Erdős–Rényi null model. The former preserves the degree distribution of the observed graph while randomizing everything else, which leads to $P_{ij} = (k_ik_j/2m)$. The latter preserves the edge density of the graph, resulting in $P_{ij} = (2m/n(n-1))$. Hereafter, modularity with the configuration and Erdős–Rényi null models is denoted as Q_{CM} and Q_{ER}, respectively. Ideally, it is desirable for the null models to preserve all properties of the observed graph that are unrelated to community structure. Since this is not the case for Q_{CM} and Q_{ER}, it is hard to interpret the specific choice of the null model. This ambiguity reduces the model explainability of the modularity function. The dynamic viewpoint and the equivalence between modularity and an SBM/DCSBM can be used to explain these null models.

From the dynamic point of view, the two null models are related to the two different random walks [29]. In particular, Q_{CM} corresponds to simple random walks, in which the random walker waits at each node for the same amount of time. On the other hand, Q_{ER} corresponds to random walkers that wait at each node for a time period inversely proportional to its degree. Such random walks can, for example, be related to information spread where high-degree nodes spread information faster. In terms of the detected communities, Q_{ER} balances communities with respect to their size, while Q_{CM} balances their volume, i.e., the sum of the node degrees in a community. Thus, Q_{CM} tends to assign high-degree nodes to different communities. When modularity is interpreted using an SBM/DCSBM, maximizing Q_{CM} is equivalent to fitting a planted partition DCSBM, while maximizing Q_{ER} is equivalent to fitting a planted partition SBM [30]. High-degree nodes are assigned to different communities when communities are detected by fitting a DCSBM, whereas there is no such tendency when fitting an SBM.

In Figure 2, we illustrate the difference in the detected communities for the two null models for a biological network. To see how high-degree nodes are distributed across communities, nodes are ordered along the vertical axis according to their degrees. When the community structure is found by

Authorized licensed use limited to: Michigan State University. Downloaded on July 01,2023 at 17:48:16 UTC from IEEE Xplore. Restrictions apply.

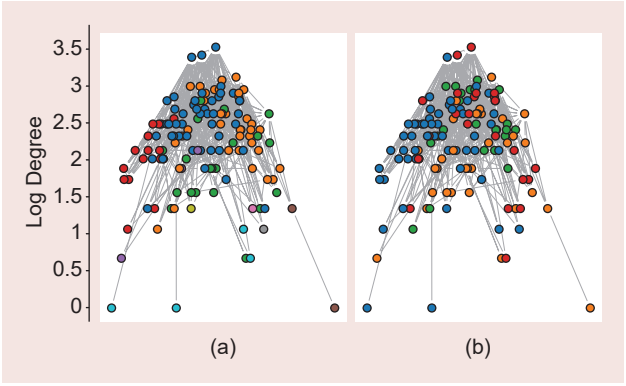


FIGURE 2. The community structures of *C. Elegans* frontal neural network [32] is found by maximizing (a) Q_{ER} and (b) Q_{CM} by using the Leiden algorithm followed by consensus clustering to improve replicability. Resolution parameters are set such that the community structure mainly consists of four large communities. Nodes are ordered along the vertical axis according to their degrees. The experiment is based on [29] and illustrates how high-degree nodes are distributed across communities for Q_{ER} and Q_{CM}.

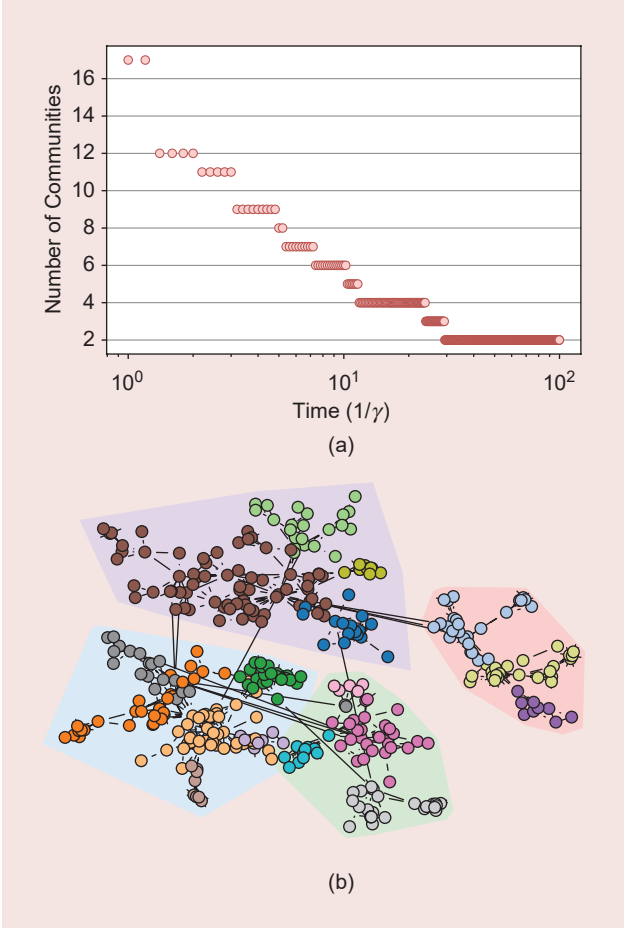


FIGURE 3. The community structure of the network of network scientists [34]. (a) The number of communities versus the time of the random walker (or $1/\gamma$). (b) The community structure with 17 and four communities: node colors indicate the community structure with 17 and four communities, and shaded polygons indicate the community structure with four communities. The communities are detected by maximizing Q_{CM} through the Leiden algorithm followed by consensus clustering to improve replicability.

maximizing Q_{ER} [Figure 2(a)], it can be seen that high-degree nodes are mostly shared by two communities. On the other hand, nodes with a high degree are distributed across all communities when Q_{CM} is maximized [Figure 2(b)].

Resolution parameter

The last element of model explainability for modularity optimization is the choice of the resolution parameter in (2). Based on the dynamic viewpoint, the resolution parameter measures the quality of a community based on the duration of the random walk. If the resolution parameter is set to γ, the quality of the community is defined based on whether it can trap the random walker for a duration of $1/\gamma$ or not [29], [33]. Thus, smaller values of the resolution parameter lead to larger communities since when the random walker wanders longer, it covers more nodes. From Figure 3, it can be seen that as the resolution parameter decreases, i.e., the time for the random walker increases, the number of communities decreases, and communities become larger. Figure 3(b) demonstrates the multiscale nature of the communities detected by setting γ to two different values corresponding to $K = 4$ and $K = 17$. The most relevant resolution parameters are the ones that lead to partitions that are optimal for longer periods of time [33]. In Figure 3(a), this corresponds to community structures with two and four communities.

Alternatively, the resolution parameter can be interpreted by the equivalence between the modularity function and SBM/DCSBM. In (5), the resolution parameter is related to the parameters of the planted partition model, i.e., θ_{in} and θ_{out}. In the planted partition model, the difference between θ_{in} and θ_{out} indicates how modular the graph is. The resolution parameter as found in (5) is proportional to this difference. In particular, large values of γ imply "stronger" community structures corresponding to small but densely connected communities, while small γ values imply "weaker" community structures yielding larger and less densely connected communities.

Comprehensibility of the output

An important aspect of interpretable data science is to evaluate the comprehensibility of the output. In the context of community detection, it is important to determine whether the extracted communities are meaningful. Existing measures and methods that can be employed to quantify the comprehensibility of detected communities include extrinsic and intrinsic measures of what constitutes a good community and statistical approaches to quantify the significance of detected communities [35].

Extrinsic measures

In unsupervised learning, such as data clustering and community detection, it is very common to quantify the comprehensibility of obtained communities through an extrinsic measure. These measures usually assume the existence of ground truth community structure and evaluate an algorithm with respect to this ground truth. Some commonly used metrics in this context are the adjusted Rand index (ARI), normalized mutual information (NMI), purity, and the F-score. However, this approach for evaluating the output of community detection algorithms is

Authorized licensed use limited to: Michigan State University. Downloaded on July 01,2023 at 17:48:16 UTC from IEEE Xplore. Restrictions apply.

problematic. First, metrics such as NMI have a computational complexity that is quadratic in the number of communities of the network, which makes them unsuitable on large-scale complex networks. Second, in most cases, there is a lack of reliable ground truth, as identifying ground truth communities requires some metadata. Finally, evaluating community structure with respect to some node metadata as if they were ground truth communities can lead to incorrect scientific conclusions. Therefore, conventional methods that quantify the comprehensibility of a community structure by using extrinsic measures fall short of fully explaining the outputs [36].

Intrinsic measures

Another common way to quantify the quality of the community structure is to use intrinsic measures. In general, intrinsic metrics are classified into four classes [21]: 1) metrics considering internal connections only (e.g., the internal density and average degree), 2) metrics considering external connections only (e.g, expansion and the cut ratio), 3) metrics considering internal and external connections (e.g., conductance and the normalized cut), and 4) model-based metrics (e.g., modularity, permanence, surprise, and communitude). While a variety of intrinsic metrics have been proposed, there is still no consensus on which metric explains the detected community structure better.

To compare the effectiveness of different intrinsic and extrinsic measures in quantifying the quality of a detected community structure, we generated Lancichinetti–Fortunato–Radicchi (LFR) benchmark networks [37] with varying values of mixing coefficient μ, which is the ratio of the external degree of a node with respect to its community to the total degree. Thus, μ controls the strength of the community structure such that smaller μ values imply more modular networks. For each network, 100 optimal and 400 suboptimal community structures are detected. Intrinsic and extrinsic measures for these 500 community structures are calculated. The Spearman's rank correlation among different intrinsic and extrinsic measures is calculated and reported in Figure 4. If an intrinsic measure is highly correlated with the extrinsic measures, this means that optimizing this intrinsic metric yields community structures that are very close to ground truth. From Figure 4, surprise is observed to be the best-performing intrinsic measure across all extrinsic measures and mixing coefficients, which is in line with recent findings [38].

Significance of detected communities

While extrinsic and intrinsic measures try to explain community structure, they cannot determine whether community structures actually exist. Therefore, it is important to measure the "significance of communities." There are two approaches proposed in the literature to tackle this problem: determining the significance of the partition and determining the significance of individual communities. The first approach tries to assess whether a given graph has a significant modular structure or not. In recent work, it has been argued that the concept of significance should be related to the robustness of a partition. Intuitively, if a network is modular, its community structure should be robust to perturbation [39]. In [39], a procedure to quantify robustness has been proposed as follows. First, the community structure of the original network is detected. Then, a fraction of the edges is rewired to generate a perturbed network, and the community structure of the perturbed network is found. The distance between the community structure of the original and perturbed networks is quantified using some partition similarity measure, such as NMI. Large values of NMI indicate that the detected communities are robust to perturbation, i.e., significant. In Figure 5, we illustrate this concept for an LFR benchmark network for various levels of mixing coefficient μ. The figure illustrates that for networks with small μ values, the community structure is robust against high ratios of rewired edges, while the community structure detected from a random network has low robustness. This difference indicates how robustness can be used to assess the significance of the community structure.

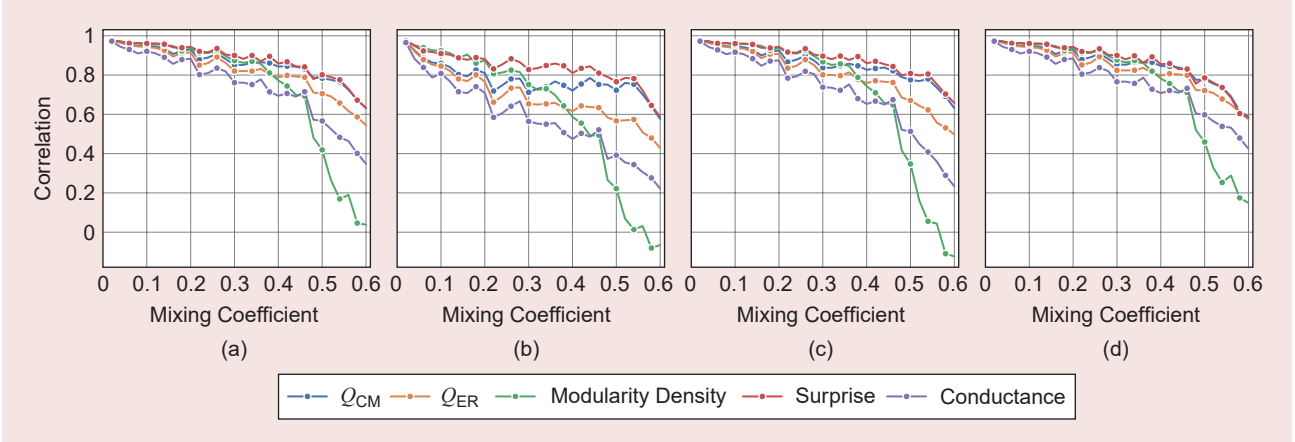


FIGURE 4. The Spearman's rank correlation among various intrinsic and extrinsic measures. LFR networks with 1,000 nodes and changing mixing coefficient μ were generated. For each μ, we detected 100 community structures by running the Leiden algorithm, with $Q_{CM}(\gamma = 1)$, 100 times. To explore the modularity landscape better, we also generated 400 suboptimal community structures by randomly shuffling 5% of the nodes' community assignments. Intrinsic and extrinsic measures for 500 community structures are calculated. (a) Adjusted mutual information. (b) ARI. (c) NMI. (d) Variation of information.

Authorized licensed use limited to: Michigan State University. Downloaded on July 01,2023 at 17:48:16 UTC from IEEE Xplore. Restrictions apply.

While the measures for assessing the statistical significance of a network partition can provide a global view of the network structure, they convey no information about the statistical significance of the individual communities. Recently, statistical methods that discriminate between a single community and structures arising as topological fluctuations have been proposed [40]. The statistical significance of a community has been quantified using metrics such as the C-score and B-score [40], information-theoretic significance measure [41], network community profile [35], and fast optimized community significance [42]. These methods quantify the statistical significance of each cluster by computing the likelihood of observing this cluster in a null model without communities. Figure 6 demonstrates this notion of significance for the detected communities of a biological metabolic network [40]. The C-score quantifies the significance of a community by assigning a rank to each node of the community based on its internal and external connections. The score of the worst node is then compared to the expected score under a null model to measure the significance of the community.

More specifically, consider a community C in an observed graph G and a configuration null model with the same degree distribution as G. Let $\hat{C}$ be a random subgraph in the null network with the same number of internal connections as C. Let k_i^{int} be the internal degree of node i, i.e., the number of connections node i makes with nodes in C. Node i is then assigned a score r_i, which is the probability of observing a node in $\hat{C}$ with an internal degree greater than or equal to k_i^{int}. This probability can be approximated by a hypergeometric distribution for the configuration null model. If G were a random graph without community structure, the r_i's would be uniformly distributed in $[0,1]$. The C-score of C is then calculated by considering its worst node w, which is the node with the highest-ranking score r_w. The C-score is the probability of observing r_w in the uniformly distributed interval $[r_{w'},1]$, where $r_{w'}$ is the second-highest score among all r_i's from C. Ideally, one would want to find communities whose C-score is smaller than a significance level, e.g., 0.05. The B-score is a refinement of the C-score and evaluates the significance of a community based on a group of nodes instead of using only the worst node of the cluster, as considering only the worst node can be a very conservative measure to assign community significance.

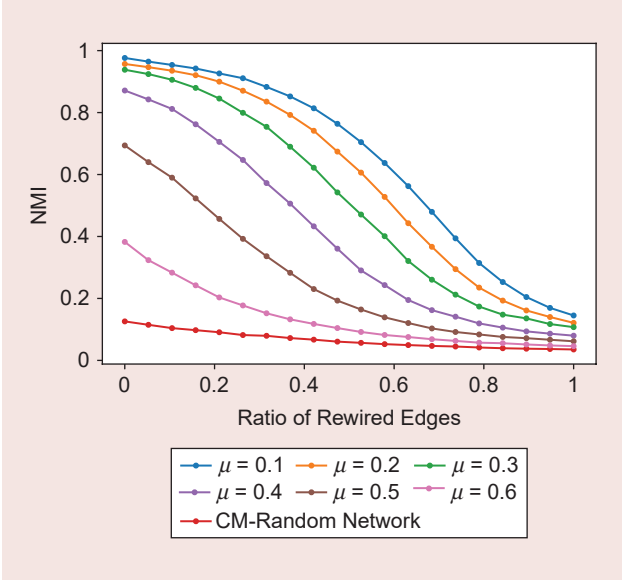


FIGURE 5. The assessment of network modularity based on the robustness of the community structure to edge perturbations. The LFR benchmark is used to generate networks with different mixing coefficients μ. Community structures are detected by running the Leiden algorithm, with $Q_{\text{CM}}(\gamma=1)$. The community structure of the unperturbed network is compared to the community structure of edge-perturbed networks to quantify robustness. Results for a random network without modular structure are also reported.

In Figure 6, only the pink community is found to be significant based on the C-score. On the other hand, green and pink communities are found to be significant based on the B-score. These two scores can also be used to define community cores [40], which are the set of nodes whose C- or B-scores are significant. In particular, the worst node is removed from the community, and the C-score (or B-score) of the remaining nodes is calculated. If the C-score is significant, the remaining nodes are the core of the community. If it is not, the worst among the remaining nodes is removed, and the C-score is recalculated. This process is iteratively repeated until the core is found or all nodes are removed. For example, based on the C-score, 19 out of the 45 nodes in the brown community form the community core, as indicated by the green nodes in Figure 6(b).

Algorithmic transparency

Algorithmic transparency in ML refers to the notion of understanding key concepts about the algorithm behavior, including the shape of the error surface, convergence of the algorithm, and uniqueness of the solution. In the context of modularity optimization-based community detection, e.g., the Louvain algorithm, there are some key algorithmic steps that are not always transparent to the user. The first issue with the transparency of the Louvain algorithm is its initialization. Usually, the Louvain algorithm starts from a singleton partition, in which each node is its own community (see Figure 1). However, there is no good justification for this choice. It is also possible to start the algorithm from a different partition. In particular, in an attempt to find better partitions, multiple consecutive iterations of the algorithm can be performed using the partition identified in one iteration as the starting point for the next iteration.

The second issue with respect to the transparency of the Louvain algorithm is in the first stage of the algorithm, where for each node, the best community assignment is determined based on the change in the modularity function. This pass across all nodes of the graph is done in a random order. This random pass affects the transparency of the detected community structure, as different random orderings of the nodes may lead to various results. The random initializations along with the random pass bring up the question of the replicability of the detected community structure (see the "Replicability" section).

Another issue that affects the transparency of an algorithm is its convergence rate. In recent work, the convergence and complexity of the Louvain algorithm have been studied [43]. The algorithm's time complexity is $O(n \log n)$. While

Authorized licensed use limited to: Michigan State University. Downloaded on July 01,2023 at 17:48:16 UTC from IEEE Xplore. Restrictions apply.

no upper bound has been established on the number of iterations and the number of passes, the algorithm is guaranteed to terminate with the use of a cutoff for the modularity gain (because of modularity being a monotonically increasing function until termination). In practice, the method needs only tens of iterations and fewer passes to terminate on most real-world inputs. Recently, it has been argued that if the initial nodes are selected based on their degree (in descending order) rather than in a random manner, the algorithm converges faster with comparable performance [44].

Post hoc interpretability

In the context of community detection, the post hoc interpretability of the detected community structure refers to the sensitivity of the community detection algorithms to the input network structure and is closely related to the comprehensibility of the output discussed in the "Comprehensibility of the Output" section. One way to quantify post hoc interpretability is perturbation analysis, where a network is perturbed by removing each node one at a time and finding the community structure of the perturbed network. The similarity of the original community structure to the community structure of the perturbed network can then be quantified using extrinsic measures, such as NMI. In Figure 7, we detail this concept for Zachary's karate club network, where the larger a node is, the greater the change in community structure, due to its removal. For instance, removing node 2 causes a larger change to the community structure compared to other nodes, as its removal substantially disturbs the internal connectivity of the blue community. This is similar to heat maps used for post hoc interpretability in ML and can help us identify influential nodes for community formation.

In parallel to this perturbation analysis, different metrics from network science literature can be adopted for post hoc interpretability. Once the community structure is detected, nodes can be classified into universal roles according to their intra- and intermodule connection patterns to explain the resulting modules [46]. Early approaches to summarizing the role of nodes in community formation are the within-module z-score (z_i) and participation coefficient (P_i), which define how a node is positioned in its own module and with respect to other modules, respectively. The within-module z-score z_i measures how "well connected" node i is to other nodes in the module and is defined as

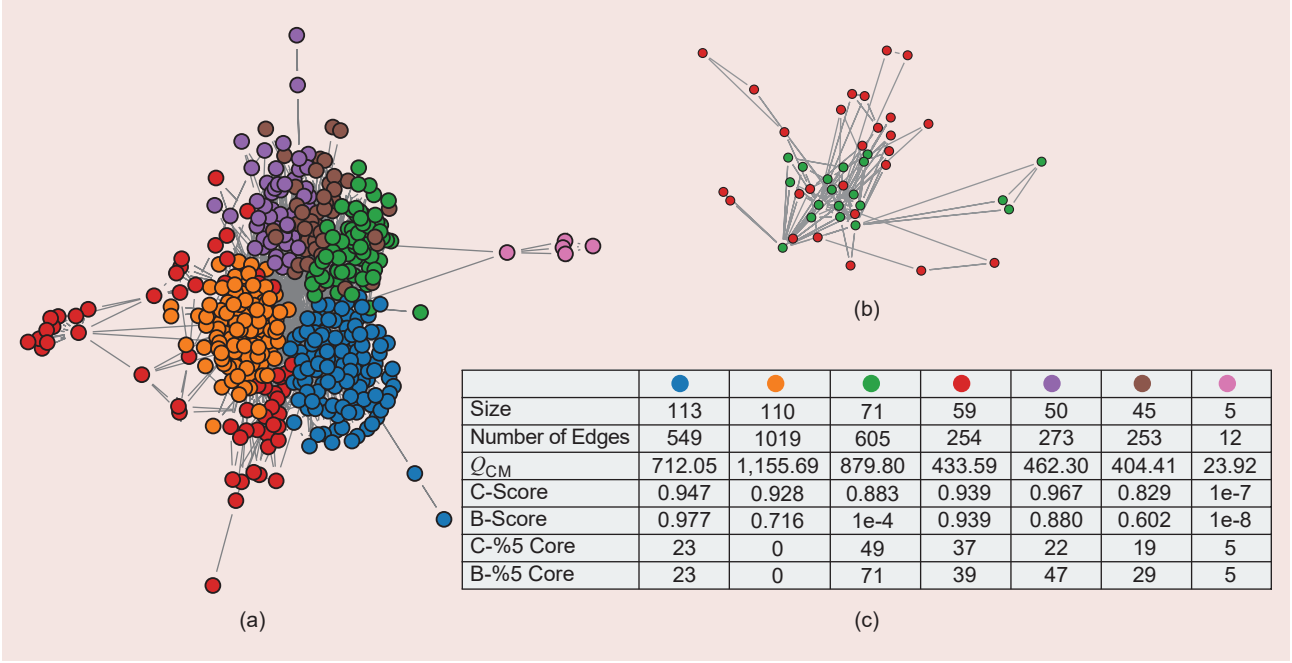
$$z_i = \frac{\kappa_i - \overline{\kappa_{s_i}}}{\sigma_{\kappa_{s_i}}}, \tag{6}$$

where κ_i is the number of links of node i within its module s_i, $\overline{\kappa_{s_i}}$ is the average of κ_i's for all nodes in s_i, and $\sigma_{\kappa_{s_i}}$ is the standard deviation of κ_i's in s_i. High values of z_i indicate high within-module degrees and vice versa.

The participation coefficient P_i measures how "well distributed" the links of node i are among different modules and is defined as

$$P_i = 1 - \sum_{s=1}^{K} \left(\frac{\kappa_{is}}{k_i}\right)^2, \tag{7}$$

where κ_{is} is the number of links of node i to nodes in module s. The participation coefficient P_i is close to one if the links of node i are uniformly distributed among all the modules and zero if all a node's links are within its own module. These metrics can be used to interpret the contribution of different nodes to the community structure, as depicted in Figure 8. Within-module z-scores and participation coefficients are indicators of



	●	●	●	●	●	●	●
Size	113	110	71	59	50	45	5
Number of Edges	549	1019	605	254	273	253	12
Q_{CM}	712.05	1,155.69	879.80	433.59	462.30	404.41	23.92
C-Score	0.947	0.928	0.883	0.939	0.967	0.829	1e-7
B-Score	0.977	0.716	1e-4	0.939	0.880	0.602	1e-8
C-%5 Core	23	0	49	37	22	19	5
B-%5 Core	23	0	71	39	47	29	5

FIGURE 6. The communities of *C. Elegans* metabolic network and analysis of significance. (a) The community structure detected by maximizing Q_{CM} using the Leiden algorithm followed by consensus clustering to improve replicability. (b) Nodes of the brown community in (a) and its C-%5core highlighted by green nodes. (c) The C- and B-scores of each community along with intrinsic measures and modularity values.

Authorized licensed use limited to: Michigan State University. Downloaded on July 01,2023 at 17:48:16 UTC from IEEE Xplore. Restrictions apply.

nodes' positioning in network topology and community structure [46]. These positionings are used to assign nodes to different roles: ultraperipheral nodes (R1), peripheral nodes (R2), nonhub connectors (R3), nonhub kinless nodes (R4), provincial hubs (R5), connector hubs (R6), and kinless hubs (R7).

Another class of network metrics that can be used for post hoc interpretability is centrality measures that quantify the importance of nodes. While most centrality measures do not take community structure into account, recently, community-aware centrality measures have been defined to quantify the importance of nodes with respect to community structure [47]. These measures reveal how influential nodes are for their respective communities and overall community structure of the network. One such measure, community centrality [34], uses eigenvectors of the modularity matrix to determine the contribution of nodes to their communities.

Replicability

In the context of community detection, replicability implies that anyone that has access to the same network data can obtain the same communities, given the code. For modularity-based community detection, replicability is hindered by two major factors: 1) the nonuniqueness of the optimum of the modularity function and 2) the stochastic nature of the optimization algorithms. First, despite the popularity of modularity maximization, there is a widespread misconception that empirical networks with modular structure tend to exhibit a clear optimal partition and that high-modularity partitions of an empirical network are structurally similar to this optimal partition. Good et al. [50] show that when modularity maximization is applied to networks with modular or hierarchical structure, these assumptions do not necessarily hold. This is known as the *extreme degeneracies of the modularity function*. The existence of extreme degeneracies in the modularity function does not depend on the detailed structure of the particular network or on any external notion of a "true" module. Instead, these solutions exist whenever a network is composed of many groups of nodes with relatively few intergroup connections.

As the number of these modules increases, the number of ways to combine them in these suboptimal ways grows exponentially. Thus, as a network becomes more modular, the globally optimal partition becomes harder to find among the growing number of suboptimal but competitive alternatives. Therefore, finding the partition with a guarantee of globally

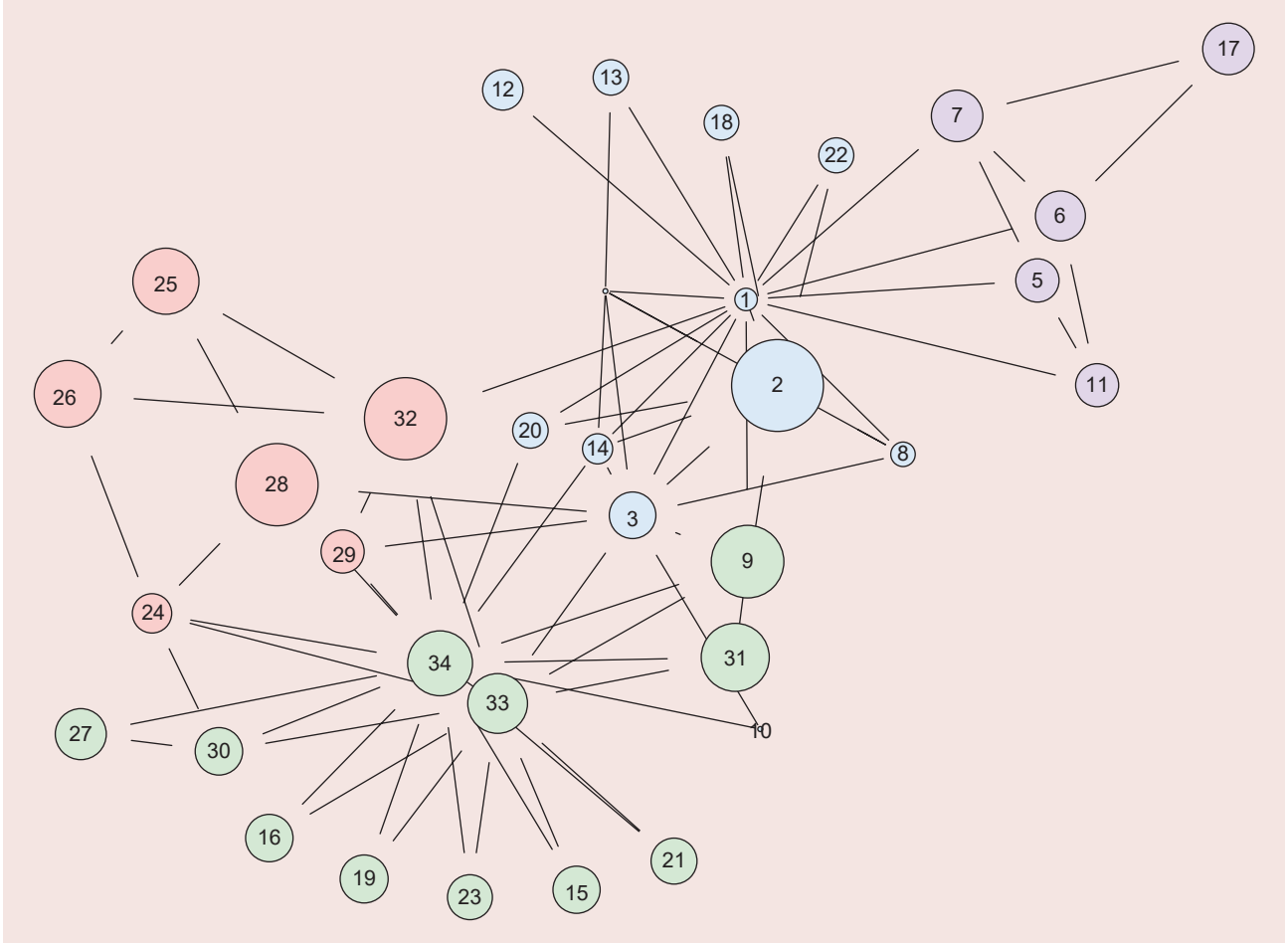


FIGURE 7. Perturbation-based post hoc interpretability applied to Zachary's karate club network [45]. Each node is removed from the network one at a time, and the change in the community structure is quantified using NMI. The community structure is detected by maximizing $Q_{\mathrm{CM}}(\gamma = 1)$ via the Leiden algorithm. Larger nodes indicate bigger changes to the community structure when those nodes are removed from the network.

Authorized licensed use limited to: Michigan State University. Downloaded on July 01,2023 at 17:48:16 UTC from IEEE Xplore. Restrictions apply.

optimizing modularity is not computationally feasible except in the smallest networks. For the multiresolution generalized version of the modularity function, $Q(\gamma)$ (2), choosing $\gamma < 1$ increases the severity of the degeneracy problem by reducing the penalty for merging modules, while choosing $\gamma > 1$ reduces it by increasing the penalty. For any fixed γ, however, there exist many networks that will exhibit severe degeneracies, and moreover, it remains unclear how to identify the "correct" value of γ without resorting to an external definition of a "true" module.

The second problem with respect to the replicability of modularity optimization methods is the stochastic nature of the optimization algorithms. Since identifying globally optimal community structure is computationally intractable, these algorithms are usually run stochastically and with random initial conditions to account for entrapment in local extrema. For example, in the Louvain algorithm, the first pass starts by ordering the nodes in a random manner and then computing the change in modularity when each node is moved to a different community. With each run of the Louvain method, this random ordering of the nodes changes, resulting in variation across the results. This variation across multiple runs can be mitigated by using numerous consecutive iterations of the algorithm through the partition identified in one iteration as the starting point for the next iteration.

One common approach to address these issues related to replicability is to obtain multiple partitions that achieve high modularity and then acquire a single partition that is more robust through consensus clustering [51]. The goal is to search for the consensus partition, i.e., the partition that is most similar, on average, to all the input partitions. In its standard formulation, consensus clustering is a difficult combinatorial optimization problem. Usually, an alternative greedy strategy is used. The association matrix, i.e., a matrix based on the co-occurrence of vertices in clusters of the input partitions, is the input to the community detection method leading to a new set of partitions. These partitions generate a new association matrix until a unique partition, which cannot be altered by further iterations, is finally reached. This procedure has proved to quickly lead to consistent and stable partitions in real networks. In Figure 9, we show this approach for a functional connectivity brain network constructed from electroencephalogram (EEG) data discussed in [48].

Recently, it has been shown that if the different partitions vary substantially, then the consensus partition may not capture the full range of behaviors and will be a poor representation of the community structure [52]. In cases like these, summarizing the community structure may require not just one but several representative partitions, which may themselves be consensus partitions for a local cluster of network divisions.

Reproducibility

In the context of community detection, *reproducibility* refers to the ability to obtain similar community structures by using the procedures provided by the original researchers. In terms of method reproducibility, one big obstacle in community detection is the selection of the number of communities. To be able to obtain consistent results across different data sets, how the number of communities is selected must be specified clearly. For modularity optimization, this corresponds to the appropriate selection of the resolution parameter γ. As discussed in the "Resolution

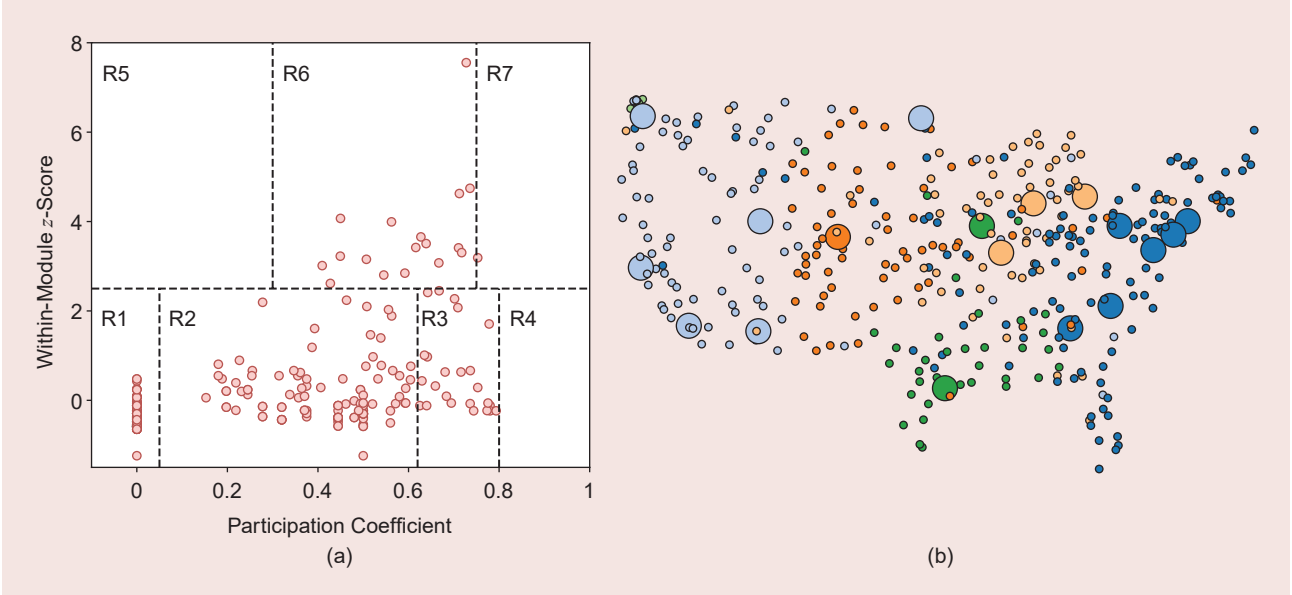


FIGURE 8. The roles of the nodes in the airport transportation network among cities in the mainland United States [46]. (a) The participation coefficients and z-scores of each node are plotted. Regions indicated as R1–7 show the different roles that can be assigned to nodes based on which intervals the participation coefficient and within-module z-score fall into. (b) The cities are plotted, where node colors indicate their communities, which are detected by maximizing $Q_{\text{CM}}(\gamma = 1)$ by using the Leiden algorithm followed by consensus clustering to improve replicability. Nodes in R6 (connector hubs) are highlighted, as they correspond to big cities that are mostly hubs of various airlines, such as Chicago; Detroit; and Atlanta.

Authorized licensed use limited to: Michigan State University. Downloaded on July 01,2023 at 17:48:16 UTC from IEEE Xplore. Restrictions apply.

Parameter" section, different γ values result in various numbers of communities. While the equivalence of modularity maximization and fitting an SBM in (5) provide a systematic way of choosing γ, they require empirical estimation of θ_{in} and θ_{out}, which is not guaranteed to converge to the optimal γ value [30]. In practice, the selection of a resolution parameter usually involves running modularity maximization algorithms with various γ values, selecting the partition with the greatest modularity at that specific value of γ and comparing the partitions.

To determine whether the obtained community structures are "robust" to the γ selection, one might look for stable plateaus in the number of communities [see Figure 3 (a)]; consider another metric, such as significance [41]; directly visualize the different community assignments across parameters; and compare obtained communities with ground truth labels using one of the extrinsic measures. A more computationally demanding approach that directly addresses this problem is to compare the obtained best modularity at each γ with the distribution of modularities obtained by running community detection across some selected random graph model, repeating this process for different γ's to identify parameter values where the obtained communities are strongest relative to the random cases [49]. Additionally, one may use a given set of partitions to generate a new partition by ensemble learning and consensus clustering [51]. Recently, a different approach, called the *Convex Hull of Admissible Modularity Partitions* (*CHAMP*) [53], that uses the union of all computed partitions to identify the CHAMP in the parameter space, has been proposed. CHAMP identifies the domains of optimality across a set of partitions by ignoring the γ that was used to compute each partition, finding instead the full domain in γ for which each partition is optimal relative to the rest of the input partitions.

The second type of reproducibility is results reproducibility, implying that the same or similar community structures are obtained from an independent study with procedures as closely matched to the original study as possible. One metric

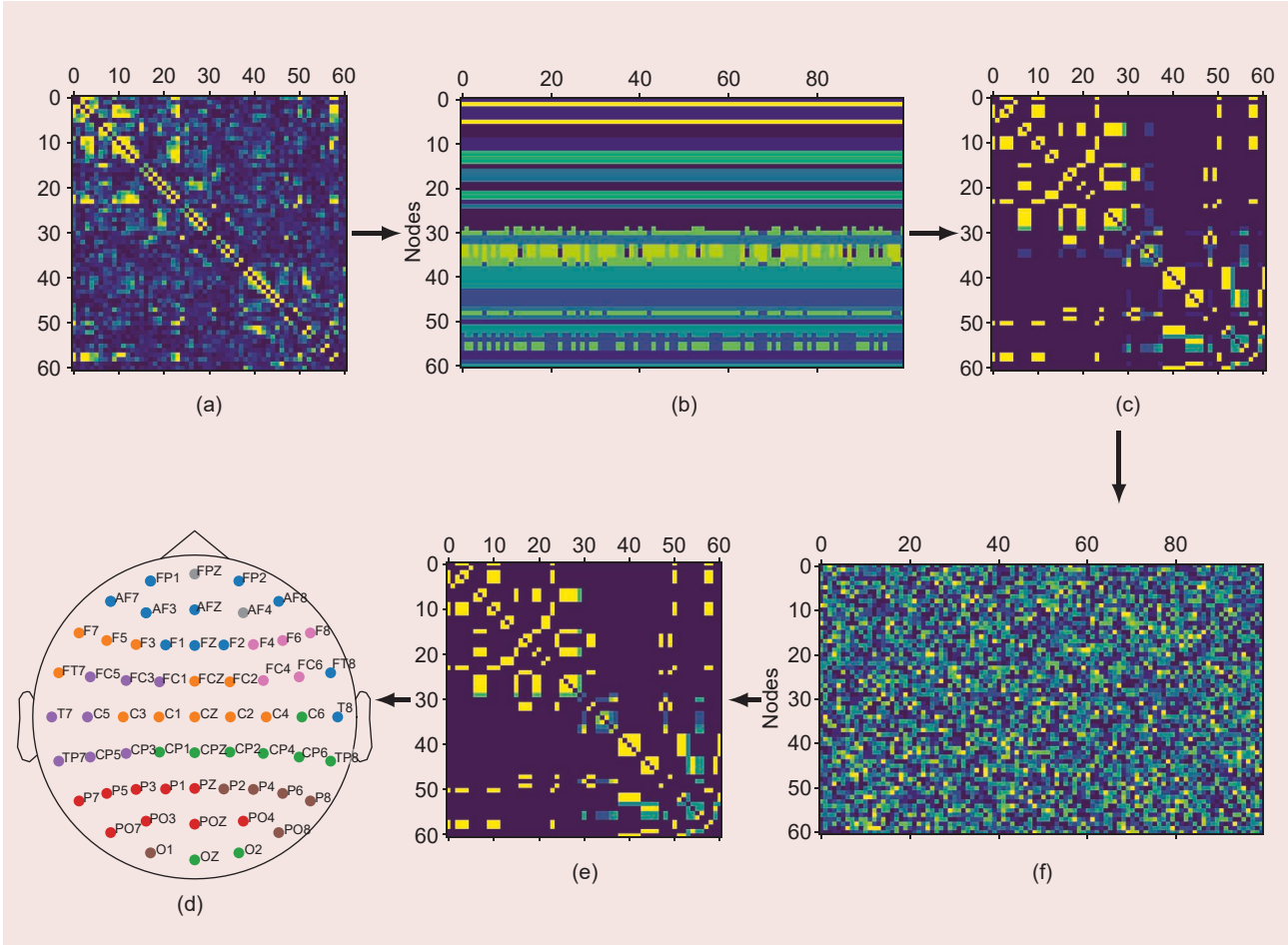


FIGURE 9. The replicability for a functional connectivity brain network from [48], using the consensus clustering approach described in [49]. Given the functional connectivity graph, multiple partitions are found by running the Louvain algorithm 100 times. From these partitions, an association matrix, whose entries indicate the number of times a pair of nodes is assigned to the same community across all partitions, is constructed. The association matrix is noisy and needs to be thresholded. The threshold is set to the expected number of times two nodes will be assigned to the same community if the partitions are randomly generated. Arbitrary partitions are found by randomizing the original partitions. From the thresholded association matrix, the consensus community structure is detected by applying the Louvain algorithm. (a) The adjacency matrix. (b) The partitions. (c) The association matrix. (d) The consensus community structure. (e) The thresholded association matrix. (f) The randomized partitions. FCZ: frontal central electrode.

Authorized licensed use limited to: Michigan State University. Downloaded on July 01,2023 at 17:48:16 UTC from IEEE Xplore. Restrictions apply.

to quantify results reproducibility is scaled inclusivity (SI) [54], which is a method to quantify the change in community structure across networks. SI independently evaluates the consistency of the classification of every node in a network. This method identifies the nodes that tend to remain in the same community across different networks' partitions, forming a "core" of that community. Likewise, the method also enables the identification of transient nodes that become part of different communities across various networks' partitions. For example, if a node i is part of module A in network l and module B in network m, then the SI between the two modules is calculated as

$$SI_i = \frac{|S_A \cap S_B|}{|S_A|} \frac{|S_A \cap S_B|}{|S_B|}, \tag{8}$$

where S_A and S_B denote sets of nodes in modules A and B and $|\cdot|$ denotes the cardinality of a set.

Different versions of SI, global SI, and module-specific SI have been implemented in prior reproducibility studies. A global SI map demonstrates the consistency of modules at each node across networks. To compute the global SI, first, the community structure of each network is detected. Next, one network is chosen as the reference, and any overlap among that network's modules and any other modules from the other networks is determined. This process results in maps of overlapping nodes among modules along with SI values summarizing the fidelity of the overlaps. A weighted sum of the overlap maps, with the SI values as the weights, is calculated, yielding a network-specific SI map. This process is repeated for all networks, and a weighted average of the network-specific SI maps, with the Jaccard indices as weights, is then calculated, resulting in the global SI map summarizing the consistency of the modular organization across networks at the nodal level. The module-specific SI, on the other hand, shows the consistency of the representative module across multiple networks. From the network-specific SI maps, it is possible to determine the most representative network with the highest SI for a particular node of interest. The module containing the node of interest is identified as the representative module. Next, modules with any overlap with the representative module are identified, and the corresponding SI values are calculated. A weighted sum of the overlapping modules is calculated with the SI values as weights, summing modules centered around the representative module.

In Figure 10, we illustrate the reproducibility of community detection for EEG functional connectivity networks constructed from 91 subjects performing the same task, i.e., error monitoring [48]. For each subject, the community structure is detected, and the global SI and module-specific SI are calculated. From Figure 10(a), we can see that frontal-central and occipital brain regions tend to preserve their community structure across subjects, while temporal brain regions do not. This is consistent with prior studies that show increased synchronized activity within these regions for error monitoring tasks [48]. In Figure 10(b), we provide the module-specific SI for the brain region corresponding to the frontal central electrode (FCz) electrode. In particular, module-specific SI computes how many times across 91 subjects this node's community structure includes any of the other nodes. From the figure, it can be seen that FCz consistently falls in the same community with other frontal and central electrodes, consistent with prior community detection results for the same data set [48].

Conclusions and future directions

In this article, we explored the issue of explainability in the context of community detection methods for graphs. While the different aspects of explainability have been previously studied in detail for various ML black-box models, the issue of explainability has not been addressed for unsupervised

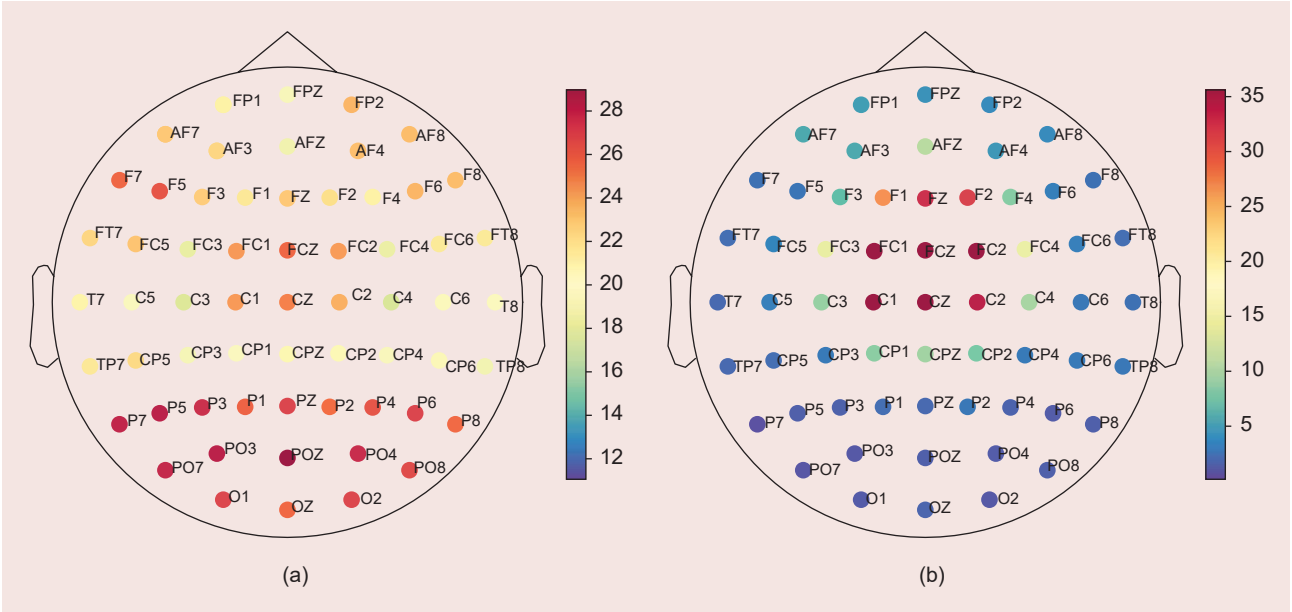


FIGURE 10. The reproducibility of community detection in a study of functional connectivity EEG networks constructed from 91 subjects [48]. (a) The global SI. (b) The module-specific SI for the frontal central electrode's (FCz's) representative community.

Authorized licensed use limited to: Michigan State University. Downloaded on July 01,2023 at 17:48:16 UTC from IEEE Xplore. Restrictions apply.

learning algorithms, such as clustering. In this article, we illustrated how metrics and concepts from network science can be adapted to study the explainability of community detection algorithms. While the availability of benchmark networks and open source code have enabled reproducible research in community detection, there are still some key issues that prevent complete transparency in community detection. These issues can be summarized as the degeneracy of the modularity function, the stochastic nature of the algorithms, and the selection of different parameters, e.g., the resolution parameter. The methods described in this article can be used to address some of these issues and provide guidelines to reduce the opacity of community detection algorithms and their outputs. While the focus of this article has been on modularity optimization-based community detection algorithms, the approaches and issues that were outlined can be extended to other community detection and data clustering algorithms as well as different learning tasks on graphs, as in the following:

- *Extensions to GSP tasks*: While the primary task considered in this article is graph-based clustering, there are other GSP tasks, such as graph signal regression and graph learning, that can be studied in terms of their explainability. GSP-based graph learning frameworks have the advantage of enforcing certain desirable representations of the signals, including the smoothness and diffusion via frequency-domain analysis, and filtering operations on graphs, making them interpretable. Moreover, recently, GSP tools have been used to better understand complex metalearning tasks by enabling users to incorporate rich semantic information [55]. In particular, high-pass graph filtering reveals which nodes can maximally describe the variations in the label agreement signal, which can be translated into an interpretable explanation.
- *Extensions to general unsupervised learning*: Unsupervised learning, e.g., clustering and outlier detection, is commonly encountered in a variety of applications where ground truth data are not available. Analyzing and interpreting results obtained through clustering is a cumbersome and challenging task, often requiring time and sophisticated, expert-based manual inspection. Unsupervised quality metrics provide only structural insights into the obtained results, and they do not explain why the clustering methodology grouped points in the same cluster. The approaches and techniques described in this article, such as quantifying the significance of the detected communities, perturbation analysis for determining the importance of each node to community formation, consensus clustering for replicability, and reproducibility metrics, can be easily adapted to data clustering and used to evaluate the explainability of different unsupervised learning algorithms. Moreover, these metrics can be incorporated into the objective functions of existing unsupervised learning methods to obtain clusters with better explainability.
- *Extensions to other data-driven community detection approaches*: Even though this article focuses on modularity-based community detection algorithms, the approaches outlined here can be extended to study the explainability of different data-driven community detection algorithms, including GNNs [56]. While the metrics introduced in this article can be employed to evaluate the comprehensibility, replicability, and reproducibility of the detected community structure, new tools will be required to capture model explainability. Recent work on the explainability of GNNs employs tools such as gradient-based methods, perturbation-based methods, surrogate methods, and decomposition methods [11]. However, most of these approaches are suitable for supervised learning tasks on graphs, including node classification and link prediction, and need to be extended for unsupervised learning tasks, such as community detection.
- *Extensions to higher-order graphs*: More recently, multilayer graphs, where each layer records a certain kind of interaction among entities, have become popular in a variety of applications. Community detection methods have been introduced for bipartite, temporal, multiplex, and multilayer graphs, where there is little consensus on what constitutes a community. For example, in the case of multiplex graphs, while some methods focus on extracting common communities across layers, others aim to uncover the heterogeneity across layers by defining common and private communities across layers. This ambiguity in what constitutes a community in these more complex graphs brings a need for new explainability tools.

Authors

Selin Aviyente (aviyente@msu.edu) received her Ph.D. degree in electrical engineering: systems from the University of Michigan, Ann Arbor. She joined the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, Michigan, 48824, USA, in 2002, where she is currently a professor. Her research interests include statistical and nonstationary signal processing, network science, and applications to neuronal signals. She serves on several technical committees of the IEEE Signal Processing Society and is an associate editor for multiple journals. She is a Senior Member of IEEE.

Abdullah Karaaslanli (karaasl1@msu.edu) received his B.S. degree in electrical and electronics engineering from Bogazici University, Istanbul, Turkey, in 2017 and is working toward his Ph.D. degree in electrical and computer engineering at Michigan State University, East Lansing, Michigan, 48824, USA. His research interests include community detection in dynamic and multilayer networks and graph signal processing for graph learning and graph signal recovery. He is a Student Member of IEEE.

References

[1] A.-L. Barabási, "Network science," *Philos. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 371, no. 1987, p. 20,120,375, 2013, doi: 10.1098/rsta.2012.0375.

[2] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, 2018, doi: 10.1109/JPROC.2018.2820126.

[3] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Phys. Rep.*, vol. 659, pp. 1–44, Nov. 2016, doi: 10.1016/j.physrep.2016.09.002.

[4] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–42, 2018, doi: 10.1145/3236009.

Authorized licensed use limited to: Michigan State University. Downloaded on July 01,2023 at 17:48:16 UTC from IEEE Xplore. Restrictions apply.

[5] R. Roscher, B. Bohn, M. F. Duarte, and J. Garcke, "Explainable machine learning for scientific insights and discoveries," *IEEE Access*, vol. 8, pp. 42,200–42,216, Feb. 2020, doi: 10.1109/ACCESS.2020.2976199.

[6] J. Basak and R. Krishnapuram, "Interpretable hierarchical clustering by constructing an unsupervised decision tree," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 1, pp. 121–132, 2005, doi: 10.1109/TKDE.2005.11.

[7] G. Corral, E. Armengol, A. Fornells, and E. Golobardes, "Explanations of unsupervised learning clustering applied to data security analysis," *Neurocomputing*, vol. 72, nos. 13–15, pp. 2754–2762, 2009, doi: 10.1016/j.neucom.2008.09.021.

[8] A. Morichetta, P. Casas, and M. Mellia, "EXPLAIN-IT: Towards explainable AI for unsupervised network traffic analysis," in *Proc. 3rd ACM CoNEXT Workshop Big Data, Mach. Learn. Artif. Intell. Data Commun. Netw.*, 2019, pp. 22–28, doi: 10.1145/3359992.3366639.

[9] D. Bertsimas, A. Orfanoudaki, and H. Wiberg, "Interpretable clustering: An optimization approach," *Mach. Learn.*, vol. 110, no. 1, pp. 89–138, 2021, doi: 10.1007/s10994-020-05896-2.

[10] S. Saisubramanian, S. Galhotra, and S. Zilberstein, "Balancing the tradeoff between clustering value and interpretability," in *Proc. AAAI/ACM Conf. AI, Ethics, Soc.*, 2020, pp. 351–357, doi: 10.1145/3375627.3375843.

[11] H. Yuan, H. Yu, S. Gui, and S. Ji, "Explainability in graph neural networks: A taxonomic survey," 2020, arXiv:2012.15445.

[12] S. Chen, Y. C. Eldar, and L. Zhao, "Graph unrolling networks: Interpretable neural networks for graph signal denoising," *IEEE Trans. Signal Process.*, vol. 69, pp. 3699–3713, Jun. 2021, doi: 10.1109/TSP.2021.3087905.

[13] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017, arXiv:1702.08608.

[14] A. A. Freitas, "Comprehensible classification models: A position paper," *ACM SIGKDD Explorations Newslett.*, vol. 15, no. 1, pp. 1–10, 2014, doi: 10.1145/2594473.2594475.

[15] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digit. Signal Process.*, vol. 73, pp. 1–15, Feb. 2018, doi: 10.1016/j.dsp.2017.10.011.

[16] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.

[17] A. Mordvintsev, C. Olah, and M. Tyka, "Inceptionism: Going deeper into neural networks," 2015. [Online]. Available: https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html

[18] R. Caruana, H. Kangarloo, J. D. Dionisio, U. Sinha, and D. Johnson, "Case-based explanation of non-case-based learning methods," in *Proc. Amer. Med. Inform. Assoc. Symp.*, 1999, pp. 212–215.

[19] H. E. Plesser, "Reproducibility vs. replicability: A brief history of a confused terminology," *Frontiers Neuroinform.*, vol. 11, p. 76, Jan. 2018, doi: 10.3389/fninf.2017.00076.

[20] S. N. Goodman, D. Fanelli, and J. P. Ioannidis, "What does research reproducibility mean?" *Sci. Transl. Med.*, vol. 8, no. 341, p. 341ps12, 2016, doi: 10.1126/scitranslmed.aaf5027.

[21] T. Chakraborty, A. Dalmia, A. Mukherjee, and N. Ganguly, "Metrics for community analysis: A survey," *ACM Comput. Surv.*, vol. 50, no. 4, pp. 1–37, 2017, doi: 10.1145/3091106.

[22] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, 2010, doi: 10.1016/j.physrep.2009.11.002.

[23] B. Karrer and M. E. Newman, "Stochastic blockmodels and community structure in networks," *Phys. Rev. E*, vol. 83, no. 1, p. 016107, 2011, doi: 10.1103/PhysRevE.83.016107.

[24] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Statistical Mech., Theory Exp.*, vol. 2008, no. 10, p. P10008, 2008, doi: 10.1088/1742-5468/2008/10/P10008.

[25] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proc. Nat. Acad. Sci.*, vol. 105, no. 4, pp. 1118–1123, 2008, doi: 10.1073/pnas.0706851105.

[26] J. Eustace, X. Wang, and Y. Cui, "Overlapping community detection using neighborhood ratio matrix," *Physica A, Statistical Mech. Appl.*, vol. 421, pp. 510–521, Mar. 2015, doi: 10.1016/j.physa.2014.11.039.

[27] J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection," *Phys. Rev. E*, vol. 74, no. 1, p. 016110, 2006, doi: 10.1103/PhysRevE.74.016110.

[28] V. A. Traag, L. Waltman, and N. J. Van Eck, "From Louvain to Leiden: Guaranteeing well-connected communities," *Scientific Rep.*, vol. 9, no. 1, p. 5233, 2019, doi: 10.1038/s41598-019-41695-z.

[29] R. Lambiotte, J.-C. Delvenne, and M. Barahona, "Random walks, Markov processes and the multiscale modular organization of complex networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 1, no. 2, pp. 76–90, 2014, doi: 10.1109/TNSE.2015.2391998.

[30] M. E. Newman, "Equivalence between modularity optimization and maximum likelihood methods for community detection," *Phys. Rev. E*, vol. 94, no. 5, p. 052315, 2016, doi: 10.1103/PhysRevE.94.052315.

[31] A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi, "A survey of statistical network models," 2010, arXiv:0912.5410v1.

[32] M. Kaiser and C. C. Hilgetag, "Nonoptimal component placement, but short processing paths, due to long-distance projections in neural systems," *PLoS Comput. Biol.*, vol. 2, no. 7, p. e95, 2006, doi: 10.1371/journal.pcbi.0020095.

[33] J.-C. Delvenne, S. N. Yaliraki, and M. Barahona, "Stability of graph communities across time scales," *Proc. Nat. Acad. Sci.*, vol. 107, no. 29, pp. 12,755–12,760, 2010, doi: 10.1073/pnas.0903215107.

[34] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E*, vol. 74, no. 3, p. 036104, 2006, doi: 10.1103/PhysRevE.74.036104.

[35] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Statistical properties of community structure in large social and information networks," in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 695–704, doi: 10.1145/1367497.1367591.

[36] L. Peel, D. B. Larremore, and A. Clauset, "The ground truth about metadata and community detection in networks," *Sci. Adv.*, vol. 3, no. 5, p. e1602548, 2017, doi: 10.1126/sciadv.1602548.

[37] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E*, vol. 78, no. 4, p. 046110, 2008, doi: 10.1103/PhysRevE.78.046110.

[38] V. A. Traag, R. Aldecoa, and J.-C. Delvenne, "Detecting communities using asymptotical surprise," *Phys. Rev. E*, vol. 92, no. 2, p. 022816, 2015, doi: 10.1103/PhysRevE.92.022816.

[39] B. Karrer, E. Levina, and M. E. Newman, "Robustness of community structure in networks," *Phys. Rev. E*, vol. 77, no. 4, p. 046119, 2008, doi: 10.1103/PhysRevE.77.046119.

[40] A. Lancichinetti, F. Radicchi, and J. J. Ramasco, "Statistical significance of communities in networks," *Phys. Rev. E*, vol. 81, no. 4, p. 046110, 2010, doi: 10.1103/PhysRevE.81.046110.

[41] V. A. Traag, G. Krings, and P. Van Dooren, "Significant scales in community structure," *Scientific Rep.*, vol. 3, no. 1, p. 2930, 2013, doi: 10.1038/srep02930.

[42] J. Palowitch, "Computing the statistical significance of optimized communities in networks," *Scientific Rep.*, vol. 9, no. 1, p. 18,444, 2019, doi: 10.1038/s41598-019-54708-8.

[43] V. A. Traag, "Faster unfolding of communities: Speeding up the Louvain algorithm," *Phys. Rev. E*, vol. 92, no. 3, p. 032801, 2015, doi: 10.1103/PhysRevE.92.032801.

[44] C. Wickramaarachchi, M. Frincu, P. Small, and V. K. Prasanna, "Fast parallel algorithm for unfolding of communities in large graphs," in *Proc. 2014 IEEE High Perform. Extreme Comput. Conf. (HPEC)*, pp. 1–6, doi: 10.1109/HPEC.2014.7040973.

[45] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropological Res.*, vol. 33, no. 4, pp. 452–473, 1977, doi: 10.1086/jar.33.4.3629752.

[46] R. Guimera and L. A. N. Amaral, "Cartography of complex networks: Modules and universal roles," *J. Statistical Mech., Theory Exp.*, vol. 2005, no. 2, p. P02001, 2005, doi: 10.1088/1742-5468/2005/02/P02001.

[47] Z. Ghalmane, M. El Hassouni, C. Cherifi, and H. Cherifi, "Centrality in modular networks," *EPJ Data Sci.*, vol. 8, no. 1, p. 15, 2019, doi: 10.1140/epjds/s13688-019-0195-7.

[48] A. Ozdemir, M. Bolanos, E. Bernat, and S. Aviyente, "Hierarchical spectral consensus clustering for group analysis of functional brain networks," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 9, pp. 2158–2169, 2015, doi: 10.1109/TBME.2015.2415733.

[49] D. S. Bassett, M. A. Porter, N. F. Wymbs, S. T. Grafton, J. M. Carlson, and P. J. Mucha, "Robust detection of dynamic community structure in networks," *Chaos, Interdisciplinary J. Nonlinear Sci.*, vol. 23, no. 1, p. 013142, 2013, doi: 10.1063/1.4790830.

[50] B. H. Good, Y.-A. D. Montjoye, and A. Clauset, "Performance of modularity maximization in practical contexts," *Phys. Rev. E*, vol. 81, no. 4, p. 046106, 2010, doi: 10.1103/PhysRevE.81.046106.

[51] A. Lancichinetti and S. Fortunato, "Consensus clustering in complex networks," *Scientific Rep.*, vol. 2, no. 1, p. 336, 2012, doi: 10.1038/srep00336.

[52] T. P. Peixoto, "Revealing consensus and dissensus between network partitions," *Phys. Rev. X*, vol. 11, no. 2, p. 021003, 2021, doi: 10.1103/PhysRevX.11.021003.

[53] W. H. Weir, S. Emmons, R. Gibson, D. Taylor, and P. J. Mucha, "Post-processing partitions to identify domains of modularity optimization," *Algorithms*, vol. 10, no. 3, p. 93, 2017, doi: 10.3390/a10030093.

[54] M. Steen, S. Hayasaka, K. Joyce, and P. Laurienti, "Assessing the consistency of community structure in complex networks," *Phys. Rev. E*, vol. 84, no. 1, p. 016111, 2011, doi: 10.1103/PhysRevE.84.016111.

[55] X. Dong, D. Thanou, L. Toni, M. Bronstein, and P. Frossard, "Graph signal processing for machine learning: A review and new perspectives," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 117–127, 2020, doi: 10.1109/MSP.2020.3014591.

[56] F. Liu *et al.*, "Deep learning for community detection: Progress, challenges and opportunities," 2020, arXiv:2005.08225.

SP

Authorized licensed use limited to: Michigan State University. Downloaded on July 01,2023 at 17:48:16 UTC from IEEE Xplore. Restrictions apply.