Applied Network Science

RESEARCH Open Access

Towards explainable community finding

Sophie Sadler[1*], Derek Greene[2*] and Daniel Archambault[1*]

*Correspondence:
1915996@swansea.ac.uk;
derek.greene@ucd.ie;
d.w.archambault@swansea.ac.uk

[1] Swansea University, Swansea, UK
[2] School of Computer Science, University College Dublin, Dublin, Ireland

Abstract

The detection of communities of nodes is an important task in understanding the structure of networks. Multiple approaches have been developed to tackle this problem, many of which are in common usage in real-world applications, such as in public health networks. However, clear insight into the reasoning behind the community labels produced by these algorithms is rarely provided. Drawing inspiration from the machine learning literature, we aim to provide post-hoc explanations for the outputs of these algorithms using interpretable features of the network. In this paper, we propose a model-agnostic methodology that identifies a set of informative features to help explain the output of a community finding algorithm. We apply it to three well-known algorithms, though the methodology is designed to generalise to new approaches. As well as identifying important features for a post-hoc explanation system, we report on the common features found made by the different algorithms and the differences between the approaches.

Keywords: Network analysis, Graph mining, Community detection, Explainability

Introduction

Explainability is a growing area of study in machine learning, due to the "black-box" nature of many algorithms in this field (Adadi and Berrada 2018). Models can have millions of parameters, which rely on long training processes for optimal tuning. This often results in a lack of understanding as to why a model returns the outputs it does, and there is a growing concern among experts that this can lead to hidden biases in the trained models. Even when such flaws are present in the model's reasoning, these can remain undetected due to its good performance on a specific set of data. To combat this problem, machine learning experts have been developing techniques that provide explanations for the outputs produced by a trained model (Wachter et al. 2017).

Most Explainable AI (XAI) techniques have focused on algorithms which are typically applied to tabular or image data. However, black-box algorithms, which provide little explanation for their outputs, also exist in tasks outside of the traditional realm of machine learning. Network analysis is another field which relies on complex, stochastic algorithms to solve its well-known problems. One such problem is that of community finding, also known as community detection. In network analysis, relational data is represented by a graph structure, where data points known as nodes are connected by edges. Communities in this context are loosely defined as sets of densely-connected nodes in the graph, where connections between the identified sets are more sparse. These sets may overlap, although more commonly algorithms partition the nodes into disjoint communities, which is our focus here. In previous work, the definition of a community has sometimes varied slightly depending on the domain context. Nevertheless, a range of well-known algorithms have been developed to try and solve the problem of identifying community structure in networks (Fortunato 2010). Although these algorithms aim to optimize a quality function, usually through a heuristic, this optimization process can be very complex, leaving even experts with little intuitive understanding of the outputs. As with machine learning algorithms, little to no explanation for these outputs is provided.

One common factor which prevents a machine learning model's performance being reasonably understood is the presence of many (sometimes hundreds or thousands) of input features. Therefore, some of the popular techniques in explainability have focused on the idea of feature importance—i.e., generating an explanation for a previously-constructed model in the form of a set of important features and their relative importance (Guidotti et al. 2018; Saarela and Jauhiainen 2021). In general, such explanations ideally present the user with features that are readily interpretable. Here an interpretable feature can be considered to be one which carries meaning to a human domain expert, such as a patient's temperature to a doctor. This is in contrast to an uninterpretable feature, such as an individual pixel in an image. Interpretable features can be used to improve understanding in many ways, including but not limited to incorporating them as the inputs to a simpler, surrogate model which mimics the performance of the one to be explained (Keane and Kenny 2019). In a community finding context, some nodes consistently participate in a single community (a core node of that community) across several runs of the algorithm, whereas others may oscillate between two or more communities. At the moment, what distinguishes these two node types remains opaque to expert users, without network features to explain it. However, it may be that if one were to know that a node is of high betweenness centrality, one could infer that this node would likely oscillate, whereas a node with high clustering coefficient may not. This said, whether the current set of social network analysis metrics can be reliably used across algorithms to explain such phenomena, including the question of whether such a set even exists, remains unknown.

Outside the direct application of this work to social networks, explainable network analysis, and in particular, explainable community finding, would bring benefits for helping public health network interventions. In this area, social network analysis is used to understand phenomena with social components and accelerate behaviour change (e.g. alcohol misuse). Interventions are designed by using social network analysis metrics and community finding algorithms (Valente 2012; Park et al. 2020; Hunter et al. 2015, 2019; Valente et al. 2013, 2015; Gesell et al. 2013) as well as developing understanding of how these phenomena spread through a community, known as social contagion (Valente and Yon 2020; Brown et al. 2017). Therefore, as researchers in this field are well versed in social network metrics, permitting more explainable community finding results would bring benefits to studies in these areas.

The incorporation of interpretable features in a post-hoc explanation may be one promising approach for improving our understanding of community finding algorithms.

In particular, we focus on interpretable features which can be understood by end-users who have social network analysis expertise and wish to apply community detection techniques to applied problems such as those in public health. Thus, our contributions are:

- A novel methodology for identifying those interpretable features which provide most insight into the outputs from stochastic community finding algorithms, from an initial longlist of candidate features.
- An application of this methodology to three well-known community finding algorithms in two experiments, and thus a list of interpretable features which relate to their performance.
- A discussion of the insight gained into these algorithms from the results.

In our experiments, we find that the same features are identified across three algorithms, indicating common underlying optimisation among these algorithms, as well as a basis for believing that these features are relevant for explainable community finding as a whole. At the single node level, these features were: clustering coefficient; triangle participation; eigenvector centrality; and expansion. At the node-pair level, these features were: the Jaccard coefficient; the cosine similarity; and to a lesser degree, the maximum betweenness centrality of an edge along the shortest path between the two nodes. All of these features are defined in "Problem formulation" section.

As well as the insight gained here by the identification of the relevant features, we also envision that our proposed approach could be incorporated into future work which generates detailed explanations for the communities found in specific graphs.

Background and related work

Explainable AI

Recently there has been an extensive interest in explaining the outputs of "black-box" AI algorithms, frequently referred to as Explainable AI (XAI) (Adadi and Berrada 2018). One strand of this work has prioritised "model transparency" (Rudin 2019), where the core idea is that a model is transparent only if we can understand its inner workings. However, for certain types of data or for more complex algorithms, such an approach might not be feasible or effective. As an alternative, "post-hoc explanations" have become popular in the field of XAI, which are more concerned with why an algorithm produced a given output, and usually involve providing some kind of rationale or evidence for that output (Lipton 2018; Bach et al. 2015; Fong and Vedaldi 2017; Sundararajan et al. 2017). Work by Lundberg and Lee (2017) provided a unified framework for interpreting predictions by assigning each input feature an importance value for a particular input. This approach is based on the early work by Shapley (2016). Another of the most well-known post-hoc approaches is local interpretable model-agnostic explanations (LIME), proposed by Ribeiro et al. (2016), which tries to understand a previously-built model by perturbing the input data to see how the resulting predictions change. The output is a set of local feature importances which explain the classification of a specific observation. The authors also introduced SP-LIME (submodular pick LIME), which differs in that it provides global explanations for the model as a whole, rather than for individual observations. Both approaches are model-agnostic in the sense that they can be applied in conjunction with any classifier and do not require inspecting the internal workings of that classifier.

Other work in post-hoc explanation by Keane and Kenny (2019) examined the use of a *twin-systems* strategy, where a complicated neural network model is mapped to a simpler, more interpretable "twin" model. This allowed the authors to understand the outputs of the former "black-box" model by using the latter "white-box" model.

Despite the extensive attention paid to XAI in recent years, the majority of this work has focused on either image or tabular data. In particular, little attention has been paid to tasks involving network data. Some initial work has begun to incorporate explainability into graph neural networks (GNNs) (Ying et al. 2019; Yuan et al. 2020), but network analysis tasks such as community detection remain unexplored, though there is some work on a similar problem in clustering algorithms (Morichetta et al. 2019; Loyola-Gonzalez et al. 2020). In this paper, our focus is specifically on community finding techniques for network data, rather than classification. We aim to identify sets of useful features which can allow us to explain the outputs of these algorithms in a post-hoc manner.

Community finding

As discussed, there has been little work on explainability or interpretability for community finding algorithms and network analysis in general, to the best of our knowledge. However, existing work on comparing the performance of several algorithms on benchmark graphs has guided our choice of algorithms and data for the experimental evaluation of our proposed features. Lancichinetti et al. (2008) propose the LFR benchmark graph dataset generator, which creates graphs with ground truth community labels on each of the nodes. They assume that both the node degrees and the community sizes follow power law distributions, and define a mixing parameter, μ, which introduces noise to the communities relative to its value. For low values of μ, the communities remain well separated and thus easy to detect, but as the mixing parameter increases, communities become harder to identify. In a subsequent paper (Lancichinetti and Fortunato 2009), the performance of several well-known community finding algorithms is then compared on this benchmark data. Lee and Archambault (2016) find that humans behave in a similar way to Lancichinetti et al. when observing their own social network, confirming that the Infomap, Louvain and Girvan-Newman algorithms were the best-performing. This led to our decision to include Infomap (Rosvall and Bergstrom 2008) and the Louvain algorithm (Blondel 2008) in our experimental evaluation.

Previous work in computational social science has also compared the performance of community finding algorithms on other datasets, including real data (Dao et al. 2020; Ghasemian et al. 2019; Peel et al. 2017). For this study, we choose to focus on the LFR data as it allows us to generate much larger datasets and to vary the mixing parameter to observe its effect on the results. The study by Bothorel et al. (2020) proposes a methodology to describe results of the algorithm to non-experts, however this differs from ours in that their aim is to assist in making a choice of algorithm for a particular problem, not to specifically explain the algorithm's results.

In addition to the LFR data and the community detection algorithms needed for our experiments, we also rely on the notion of the a node's ease of clustering. Nodes which are easy to cluster are those which are consistently assigned to the same community, while a node which the algorithm finds it hard to cluster will oscillate between two or more communities across successive runs. Existing literature in this vein originates in papers unrelated to networks and community finding, but focused on more general clustering algorithms, e.g. *k*-means clustering (von Luxburg 2010; Ben-David et al. 2007). However, our proposed work differs from theirs as we centre our definition of a node's ease of clustering on its entropy in a coassociation matrix. Entries in the matrix describe how frequently two nodes are clustered into the same community. The concept of a coassociation matrix describing the relationship between pairs of nodes was derived from work proposed by Strehl (2002). In a similar approach, the authors of the LFR benchmark explore the use of consensus clustering to determine community structure over successive runs of the algorithm (Lancichinetti and Fortunato 2012a). Other works addressing the consistency of community finding algorithms (Chakraborty et al. 2013; Francisco and Oliveira 2011) are not directly relevant to our node feature experiments described in this paper, but may have relevance to the future work we propose in identifying community features.

Problem formulation

Due to their widespread adoption and suitability for our proposed methodology, in our experiments we focus on stochastic algorithms, where the community structure can change between successive runs, and on algorithms which find node partitions (i.e. each node belongs to exactly one community). Extending our approach to algorithms which generate overlapping communities will require additional steps, so we reserve this for future work. As the intention is to identify features which contribute intuitive understanding, our emphasis is on selecting features which are simple and easily understood to end-users, though specifically those with social network analysis expertise. We propose a model-agnostic methodology which can be adapted to any stochastic algorithm of interest, however we test it here on three in particular.

We distinguish between two "levels" of graph feature, allowing for understanding of the nodes' community membership from two different perspectives. The first of these is at the *node-level*. Features at this level are calculated for individual nodes of the graph, with the aim to understand the community membership of that specific node. To motivate this problem in a social network context, suppose a node is occasionally classified as belonging to a community with extremist views on certain runs of a community finding algorithm. Understanding why this node has this varying classification would be important as this classification is not certain and could have important repercussions for the individual. The second is at the *node-pair-level*, where features are calculated for pairs of nodes. The aim is to understand why two nodes belong to either the same or different communities. In a social network context, if two nodes belong a community that holds an extremist view, it is important to understand why they have been placed into this community. Similarly, if one node belongs to this community and another does not, it is important to understand why the nodes have been separated.

In this work we use a large number of synthetically-generated graphs to verify our approach. We employ the use of synthetic data to ensure the results are not a consequence of the characteristics of a single network (as real data is sparsely available) and to allow us to vary parameters of the network structure consistently to observe how these parameters affect the results. However, with the aim to apply these results to real data in the future, we use a synthetic generation process which can closely mimic the observed structure of real-world networks. Specifically, the synthetic graphs were generated using the LFR benchmark algorithm (Lancichinetti et al. 2008). An additional benefit of this approach is that existing work has already evaluated the performance of community finding algorithms on LFR graphs. We use several values of the LFR mixing parameter μ in order to ascertain whether the separation of the communities affects the identified features.

Our approach is to identify a longlist of features at both the node-level and the node-pair-level. We then use these features as the input data for a classification task, and extract the most informative features using permutation importance (Algorithm 1) for our trained model. Since some of the features in our longlist depend on nodes' community labels, we calculate these from many runs of the community finding algorithm using a mean average. If the feature does not depend on community label, it can be directly calculated once for each node or pair of nodes. To inform our choice of features, we consult the survey by Chakraborty et al. (2017) and select features which are widely adopted, state-of-the-art metrics for community evaluation such that they can be easily recognised and interpreted by experts on network analysis. The features selected for our experiments are described below.

Algorithm 1 Permutation Importance

```
Input: Fitted predictive model $M$; data, $D$ (training or validation)
Calculate score $s$ of model $M$ on $D$ (here, classification accuracy).
for feature $j$, i.e. column of $D$ do
    for repetition $k = 1$ to $K$ do
        Randomly shuffle column $j$ of $D$ to receive corrupted data, $\tilde{D}_{k,j}$.
        Compute the score $s_{k,j}$ of model $M$ on $\tilde{D}_{k,j}$.
    end for
    Compute importance $i_j$ for feature $f_j$
                    $i_j = s - \frac{1}{K} \sum_{k=1}^{K} s_{k,j}$
end for
```

Node features

The node-level features selected are defined as follows for node i (letting w be the number of nodes in the same community as i). Let the graph be $G = (V, E)$ with V denoting the set of nodes and E the set of edges:

- *Degree*: The number of edges adjacent to i, $\deg(i)$.
- E_{in}: The number of edges adjacent to i within its community. Adapted to a single node from the original definition by Radicchi et al. (2004)
- E_{out}: The number of edges adjacent to i which connect it to nodes outside its community. Adapted to a single node from the definition by Radicchi et al. (2004). Note that $E_{in} + E_{out} = \deg(i)$.

- *E_{in} over E_{out}*: For a given node, the ratio of the number of edges connecting it to other nodes within the same community, relative to the number of edges it has to nodes in other communities:

  $$\frac{E_{in}}{E_{out}}.$$

- *Out Degree Fraction (ODF)*: Adapted to a single node from the definition by Flake et al. (2000), this is the ratio of edges connecting node i to nodes in other communities, relative to its total degree:

  $$\frac{E_{out}}{\deg(i)}$$

- *Expansion*: Adapted from the definition by Radicchi et al. (2004), this is number of edges from a single node i to nodes assigned to other communities, normalised with respect to the number of nodes in the same community as i:

  $$\frac{E_{out}}{w}$$

- *Cut Ratio*: Adapted to a single node from the graph cut measure discussed by Fortunato (2010). As with the metric above, this considers the number of outgoing edges from i to other communities, but in this case normalised with respect to the number of nodes **not** in the same community as i:

  $$\frac{E_{out}}{|V| - w}$$

- *Conductance*: Adapted to a single node from the clustering objective described by Shi and Malik (2000), this measure is the ratio between the connections for node i within its community and its total number of connections:

  $$\frac{E_{out}}{\deg(i) + E_{in}}$$

- *Average Shortest Path*: The mean of the shortest path lengths from node i to all other nodes in the graph.
- *Triangle Participation*: Let c_i be the number of nodes with which node i shares a common neighbour within its assigned community. Then triangle participation is given by the fraction:

  $$\frac{c_i}{w}$$

- *Clustering Coefficient*: The local clustering coefficient of a node measures how close its neighbours are to forming a clique (Watts and Strogatz 1998). Formally, let T_i be the number of triangles containing i across the whole graph. Then the clustering coefficient of node i is given by:

$$\frac{2T_i}{\deg(i)(\deg(i)-1)}$$

- *Betweenness Centrality*: Let $\sigma(j,k)$ be the number of shortest (j, k) paths, and $\sigma(j,k|i)$ be the number of those paths that pass through i. Then the betweenness centrality of node i is given by Brandes (2001):

$$\sum_{j,k \in V} \frac{\sigma(j,k|i)}{\sigma(j,k)}$$

  Note that if $j = k, \sigma(j,k) = 1$ and if either j or $k = i$, then $\sigma(j,k|i) = 0$. Intuitively, a high betweenness centrality score for a node often indicates that it holds a bridging position in a network.
- *Eigenvector Centrality*: Proposed by Bonacich (1986). The eigenvector centrality of node i is the ith entry in the vector $\mathbf{x}$ which solves the eigenvector equation:

$$\mathbf{Ax} = \lambda\mathbf{x}$$

  where $\mathbf{A}$ is the adjacency matrix with node i represented in the ith row/column. Based on the definition above, this measures deems that a node is important if it is connected to other important nodes.
- *Closeness Centrality*: Refers to the centrality measure proposed by Freeman (1979). Let $d(i, j)$ be the length of the shortest path between nodes i and j. Then the closeness centrality of node i is given by:

$$\frac{|V|-1}{\sum_{j \neq i} d(j,i)}$$

  This provides us with an assessment of the extent to which node i is close to all other nodes in a network, either directly or indirectly.

Node-pair features

Given a pair of nodes (i, j), we define a number of node-pair-level features:

- *Shortest Path Length*: The least number of edges separating nodes i and j.
- *Common Neighbours*: The number of shared nodes adjacent to both i and j, which we denote as n_{ij}.
- *Max Edge Centrality*: The maximum over centralities of all edges along the shortest path. The edge centrality is defined in a similar manner to betweenness centrality for nodes (Brandes 2001). That is, for a given edge e, we compute

$$\sum_{j,k \in V} \frac{\sigma(j,k|e)}{\sigma(j,k)}$$

  where $\sigma(j,k|e)$ now refers to the number of shortest paths between j and k passing through an edge e rather than a node i.
- *Cosine Similarity*: Frequently used to measure similarity for textual data, but can also be applied to assess node-pair similarity in the context of graphs:

$$\frac{n_{ij}}{\sqrt{\deg(i)}\sqrt{\deg(j)}}$$

- *Jaccard Coefficient*: A common set similarity measure, originally proposed in Jaccard (1912). In a graph context, let $\Gamma(i)$ be the set of neighbours of node i. Then the Jaccard coefficient of nodes i and j is given by:

  $$\frac{|\Gamma(i) \cap \Gamma(j)|}{|\Gamma(i) \cup \Gamma(j)|}$$

  A higher value for this measure indicates a greater level of overlap between the neighbours of i and j, relative to their full sets of individual connections.

Classification problems

For node-pair-level features, there is an obvious binary classification problem where pairs of nodes are labelled as belonging to the "same community" or to "different communities". Since the algorithms of interest in our work are stochastic in nature, a pair of nodes may sometimes be in the same community, while for other runs of the algorithm the pair may not appear in the same community. Over the course of many runs of a given algorithm, pairs can simply be labelled as "same community" if they are in the same community for more than half of the runs, and "different community" if they are in the same community for less than half of the runs. In the unlikely event they are in the same community for exactly half of the runs, we have chosen arbitrarily to label them "same community".

For node-level features, defining a classification problem is harder since, on consecutive runs of the community detection algorithm, the number of communities can vary, or the community labels can be permuted. Thus, classifying a node into its "correct" community is not a well-defined problem. Instead, we propose a binary classification problem determining whether the node is "easy" or "hard" to assign to a community, by observing how frequently it flips between communities on successive algorithmic runs. To define this mathematically, we require a coassociation matrix, described in "Coassociation matrix" section below. This will allow us to identify features that are predictive in whether a node is strongly associated with a specific community (near its "centre"), or whether it lies on the border between two or more communities. Nodes of the latter type may be of particular interest in certain domains, such as public health.

In order to label the nodes as "easy" or "hard" to assign to a community, we incorporate the use of a coassociation matrix, defined below.

Coassociation matrix

For a given graph and community detection algorithm, we can construct a *coassociation matrix*, C, using the outputs of many runs of the algorithm on the graph. In our methodology, we use the same set of runs to calculate both the community-dependent features, and the coassociation matrix. Let r_{ij} be the number of runs for which nodes i and j are in the same community, and let R be the total number of runs. The value for the entry ij in the matrix is given by:

$$C_{ij} = \frac{r_{ij}}{R}$$

Intuitively, the coassociation matrix represents the proportion of runs for which two nodes are in the same community, for every pair of nodes.

In order to classify nodes as either "easy to cluster" or "hard to cluster", we then calculate the entropy of each node from the coassociation matrix as follows:

$$E_i = \frac{\sum_j p_{ij}}{N}$$

where N is the number of nodes and p_{ij} is defined as follows:

$$p_{ij} = \begin{cases} -C_{ij}\log_2(C_{ij}) & \text{if } C_{ij} > 0 \\ 0 & \text{if } C_{ij} \leq 0 \end{cases}$$

Unfortunately, these entropy values are not as intuitively understood as the raw coassociation matrix entries. Thus, it is not as simple to label nodes as "easy to cluster" or "hard to cluster" directly from their entropy values as it is to label pairs as "same community" or "different community" directly from the coassociation matrix. Instead, once every node is assigned an entropy, we use one-dimensional k-means clustering (with $k = 2$ clusters) to separate nodes into two training classes: those with low entropy belong to the "easy to cluster" class, and those with high entropy belong to the "hard to cluster" class. Intuitively, these correspond to nodes which are often assigned to the same community by the algorithm and those which are often assigned to different communities.

Summary

Our aim is to identify human-interpretable graph features which relate to the community membership determined by a community finding algorithm. In order to select the more informative features from a predefined longlist of candidates, we define two simple binary classification problems: one for node-level features, where we will predict a node's ease of assignment to a community; and one for node-pair-level features, where we will predict whether the two nodes belong to the same community or not. We will then find the permutation importance of each feature from our model to identify which features provide the most information about the output label.

Methodology

Our experiments take place on more than one graph, μ value (as described in "Community finding" section), algorithm, and even classification task. Having several independent variables enables us to answer the following research questions:

> *RQ1: Do the most informative node features depend on the community finding algorithm used?*
>
> *RQ2: Do the most informative node-pair features depend on the community finding algorithm used?*
>
> *RQ3: How do the most informative node features vary with the degree of community*

separation, as defined by the mixing parameter, μ?

RQ4: How do the most informative node-pair features vary with the degree of community separation, as defined by the mixing parameter, μ?

RQ5: In all cases, what are the most predictive features?

Although we did not form a strong hypothesis for the latter three questions, we hypothesise that the most predictive features would vary by algorithm.

H1: The most informative node and features will depend on the community finding algorithm.

H2: The most informative node-pair and features will depend on the community finding algorithm.

In order to answer the questions above, we now present our experimental and statistical methodology, which may prove useful in tackling the evaluation of community finding algorithms more generally. This methodology is also illustrated in Fig. 1.

Experimental methodology

We test our approach using three popular methods to detect community structure, each based on different concepts (for example, we only use one modularity optimization algorithm):

1. *Infomap* (Rosvall and Bergstrom 2008), also known as the map equation, uses an information-theoretic approach, where nodes are represented by codewords composed of two parts, the first of which is provided by the community it belongs to. The community memberships are optimised by minimising the average code length describing random walks on the network.
2. *Louvain* (Blondel 2008), is a modularity optimization approach which involves two steps. Firstly, the common modularity objective is optimized at a local level to create small communities. Next, each small community is treated as a single node and the

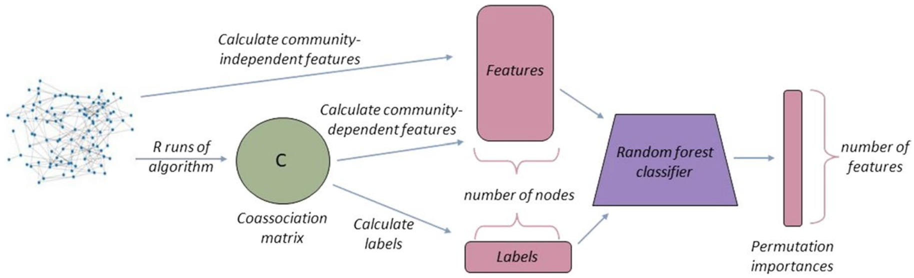


Fig. 1 Experiments for determining explainable social network analysis metrics in the node feature experiment. A similar methodology is applied to the node-pair experiment. After *R* runs of the algorithm, a coassociation matrix is constructed encoding how often two nodes are classified in the same community. Feature values are computed and provided as input to a random forest classifier to determine permutation importance. The distributions of permutation importance can be compared across all graphs to identify explainable metrics

first step is repeated. By following this agglomerative process, a hierarchy of communities is constructed.

3. *LPA*, the label propagation algorithm proposed by Raghavan et al. (2007), assumes that nodes should belong to the same community as most of their neighbours. To begin, each node is initialised with a unique community and then these labels are then iteratively propagated through the network. After each iteration, a node receives the same label as the majority of its neighbours. Once this process is complete, nodes sharing the same label are grouped together as communities.

When constructing our networks, we selected μ values of 0.2, 0.3, and 0.4. As described in "Community finding" section, this parameter controls the level of separation or mixing between communities, where the higher the value of μ, the less easy it is to distinguish between different communities (Fig. 2).

At each value of μ, a set of graphs, Γ, are generated before any experiments take place. This set of graphs is the same size, $|\Gamma|$, for each value of μ. In order to match the hyperparameters used by Lancichinetti et al. (2008) in the original LFR benchmark paper, we use the LFR generator in NetworkX to generate networks with 1000 nodes of average degree 20 and maximum degree 50. We set the hyperparameters τ_1 and τ_2 to 3 and 2 respectively.

Each experiment is then defined by three categories: the μ value; the community detection algorithm; and the feature type (node vs node-pair). This results in 18 possible experiments from the 3 algorithms, 3 mixing parameters and 2 feature types. Data from the $|\Gamma|$ graphs at the relevant value of μ are used for the experiment. For each μ -algorithm-feature type combination, the following procedure is then performed.

Firstly, the algorithm is run 1000 times on each of the $|\Gamma|$ graphs. Using these runs, any community-dependent features are calculated, along with the coassociation matrix. Features which are community-independent are also calculated at this stage, although they do not depend on the runs. The nodes or pairs-of-nodes must then be labelled according to the binary classification problem. The labelling procedures are described separately for the two feature-types in the relevant experiment sections.

Now, for each of the graphs of the experiment, we have a dataset of either nodes or pairs of nodes, each member of which is labelled and has a list of feature values. A random forest with 100 trees is then trained to classify the dataset for the specific graph.

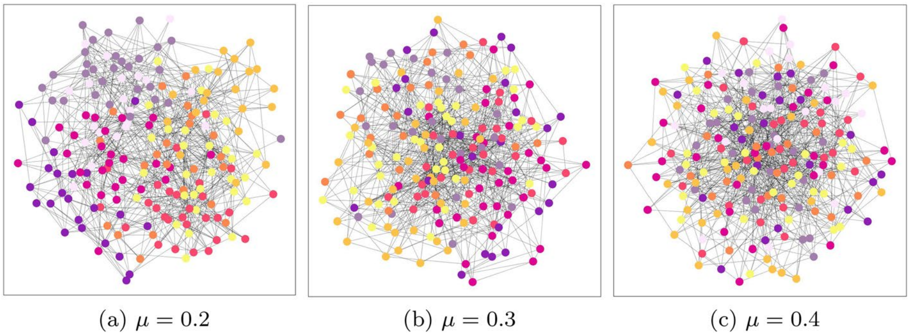

(a) $\mu = 0.2$ (b) $\mu = 0.3$ (c) $\mu = 0.4$

Fig. 2 Example graphs with 200 nodes at the three μ values. Communities shown with colour. Increased mixing parameter increases the prevalence of edges between communities

During training we use 5-fold cross-validation and repeat this for 10 training runs. A permutation importance is calculated for each node or node-pair feature after each of the 50 runs, using the held-out test data. At the end of the 50 cross-validation runs, a mean average of the 50 gathered permutation importance scores is taken for each node or node-pair feature. This gives us its final importance score as generated by this graph. Overall, this results in $|\Gamma|$ permutation importance values for each feature. The full experimental methodology for node features is represented in Algorithm 2. For node-pair features, the algorithm is identical, looping over node-pairs instead of nodes.

Algorithm 2 Node Features Experiment

```
Input: Graphs $\Gamma_\mu$ of mixing parameter $\mu$
Output: Permutation importance matrix $\mathbf{P}_{|\mathbf{\Gamma}_\mu|,\mathbf{f}}$ where $f$ is the number of features in the longlist.
for graph $G = (V, E)$ in $\Gamma_\mu$ do
    Run chosen community finding algorithm 1000 times
    Calculate coassociation matrix, $\mathbf{C_G}$
    Calculate feature vector $X_G = (x_{G,1}, \dots, x_{G,f}))$
    for node $v$ in $V = \{1, \dots, n\}$ do
        Calculate label $y_v$ according to $\mathbf{C_G}$
    end for
    for training run $r = 1$ to $50$ do
        Train random forest using $(X, Y = (y_1, \dots, y_v))$
        Calculate permutation importances for each feature, $P_{G,r} = (p_{G,1,r}, \dots, p_{G,f,r})$
    end for
    Calculate mean permutation importance across the 50 training runs, $P_G = (p_{G,1}, \dots, p_{G,f})$
end for
Output final permutation importance matrix $\mathbf{P} = (P_1, \dots, P_{\Gamma_\mu})$
```

Statistical methodology

For both experiments above, we have distributions of our features over the runs of the experiment. These distributions can be compared to determine statistical significance of the difference between them, and the size of this difference, in order to identify the features of interest. This statistical analysis and the final conclusions drawn are specific to the μ-algorithm-feature type combination of the experiment.

In order to develop an appropriate statistical methodology, we performed a pilot study using 20 graphs at each μ value (giving 60 graphs in total). For each experiment, this gave us 20 values of permutation importance for each feature, on which we carried out Shapiro-Wilk tests. In this pilot study, 67% of the features across all algorithms, feature types and μ values were normally distributed, so we started with a normal assumption. On this basis, the statistical methodology would be as follows:

1. Perform a power analysis with a normal assumption to determine the value of $|\Gamma|$ required to draw statistically significant conclusions.
2. Carry out the experiments to obtain $|\Gamma|$ values for each feature of each experiment.
3. Confirm with a repeat of the Shapiro-Wilk tests that these $|\Gamma|$ values are indeed normally distributed in the majority of cases.
4. If the distributions are normal, perform pairwise t-tests with Bonferroni–Holm corrections using these values. Otherwise, perform pairwise Wilcoxon tests with Bonferroni–Holm corrections using these values.

Table 1 The numbers of communities identified by each algorithm on graphs with 1000 nodes

Number of communities

	Mean	Median	Std.
Infomap, mu 0.2	40.65	41.00	2.45
Infomap, mu 0.3	39.93	40.00	2.33
Infomap, mu 0.4	35.76	36.00	2.65
Louvain, mu 0.2	34.45	34.00	1.83
Louvain, mu 0.3	29.56	29.00	1.79
Louvain, mu 0.4	24.20	24.00	1.61
LPA, mu 0.2	39.14	39.00	2.54
LPA, mu 0.3	36.33	36.00	3.06

The mean, median and standard deviation are calculated across all 120 graphs in each case

Table 2 Statistics on the normalised mutual information scores for two algorithms on graphs of the same μ value

NMI statistics

	Mean	Median	Std.
Infomap and Louvain, mu 0.2	0.977	0.978	0.008
Infomap and Louvain, mu 0.3	0.949	0.949	0.012
Infomap and Louvain, mu 0.4	0.883	0.884	0.021
Infomap and LPA, mu 0.2	0.991	0.992	0.007
Infomap and LPA, mu 0.3	0.962	0.966	0.022
Louvain and LPA, mu 0.2	0.970	0.970	0.012
Louvain and LPA, mu 0.3	0.922	0.925	0.025

For each row of the table, 1000 pairs of partitions were uniformly randomly chosen for each graph. As there are 120 graphs for each μ value, this means 120,000 values contribute to each statistic

Power analysis was conducted with the following parameters: Cohen's effect size of 0.3, significance level of 0.05, and a power of 0.9. The power analysis concluded 119 graphs were necessary for our experiment, which we rounded to 120. At this stage, we generated 360 new graphs (120 at each μ level) for our experiment.

The application of this methodology to our new data set revealed that the distributions of metric values were not normally distributed. Therefore, to determine significance, pairwise Wilcoxon tests with Bonferroni–Holm correction were applied to our data to determine the significant results.

Exp. 1: node feature experiment

Once the experimental data was collected, the 0.4-LPA-node and 0.4-LPA-node-pair experiments were omitted. This is because LPA clustered a majority of nodes into one large community at this μ value, generating features and labels that were not suitable for our experiments. Essentially, LPA was unable to recognise community structure at this high degree of mixing. All other experimental data is reported.

Tables 1 and 2 respectively show statistics on the number of communities detected by each algorithm across graphs of a common μ value, and normalised mutual information (NMI) scores comparing the performance of pairs of algorithms. In the first of these, we can see that communities range in number from 24 to 40, resulting in a mean community size between 25 and 45 nodes. In reality, sizes of communities created using the LFR generator follow a power law, so many will be much larger or smaller than the mean. In the second table, NMI scores are generally high in all cases, although decrease as the μ value increases, as one might expect. Overall these results suggest that there are some communities with very large, stable cores, and that the nodes which frequently change across multiple algorithmic runs are single nodes on the periphery of these large communities, or belong to the much smaller communities.

Experiment

The classification labels for the node feature experiments are calculated for a single graph as follows. The entropy of each node is calculated from the coassociation matrix of the current graph, and k-means clustering of these entropy values is performed to separate the nodes into "easy to cluster" and "hard to cluster" nodes. However, using this process, we have a very low proportion of "hard to cluster" nodes. The proportion of nodes labelled as "hard to cluster" are reported in Table 3. For low mixing parameter values, this can be as low as 9%. This reinforces the finding from Tables 1 and 2 that there are large, central cores to the communities with a small number of nodes on the periphery or in smaller communities. However, the proportion of "hard to cluster" nodes can rise to as high as 25% with an increased mixing parameter, indicating that this is a distinct class of nodes. Due to the low proportions of "hard to cluster" nodes, we propose using undersampling. Rather than undersampling randomly, we propose using the "easiest" nodes to cluster (those with the lowest entropy) until the number of "hard" nodes is 75% that of the number of "easy" nodes. Using this *strategic undersampling* method enables us to identify node features which distinguish between truly separate classes, rather than distinguishing between nodes with an entropy either side of the arbitrary cut-off generated by the k-means clustering.

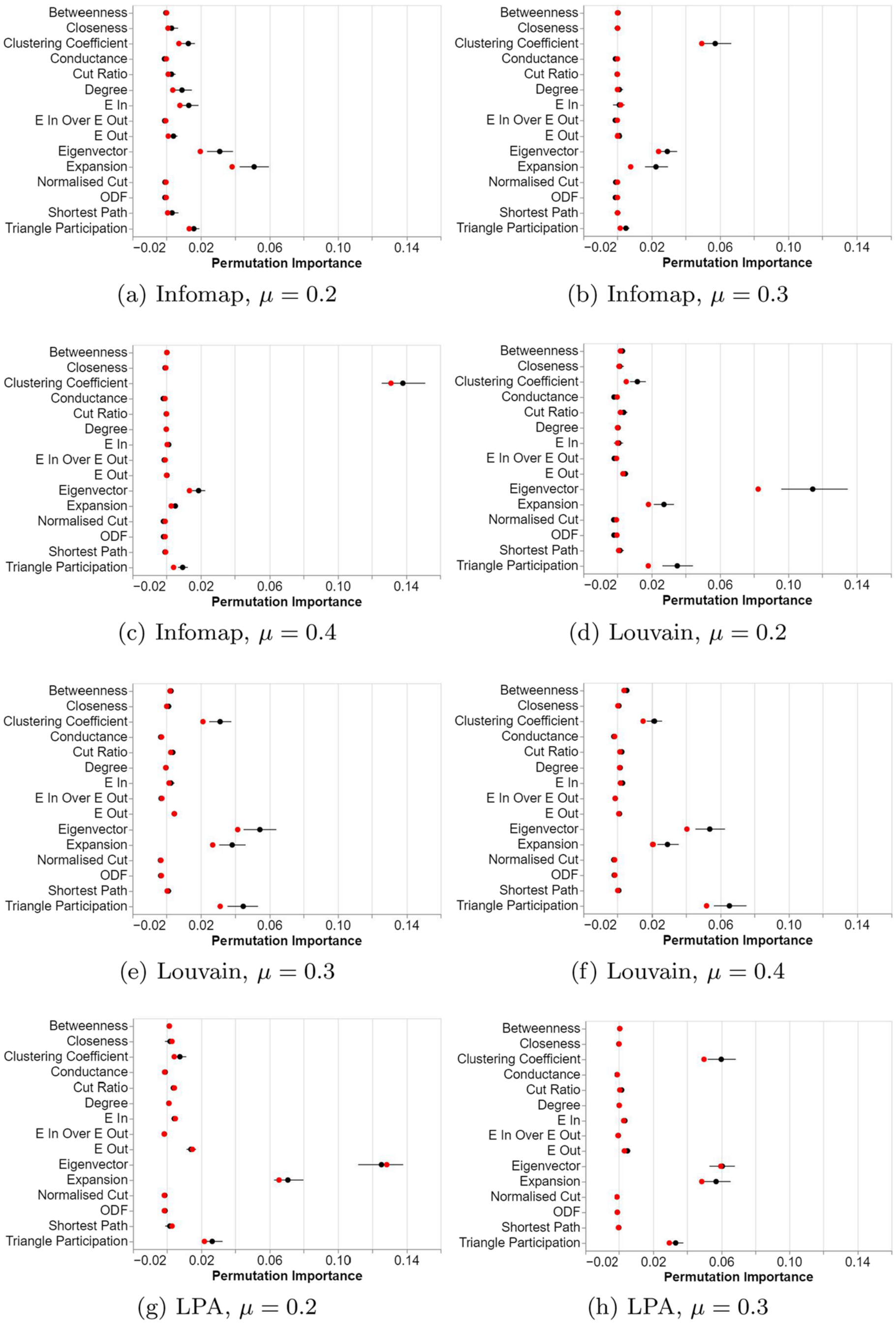
Results

From these three experiments (displayed in Fig. 3), we see that four of the features consistently have a non-zero permutation importance: clustering coefficient,

Table 3 Statistics on the proportion of nodes labelled as "hard to cluster" after running each algorithm on graphs of varying μ value

Proportion of "hard to cluster" nodes			
	Mean	**Median**	**Std.**
Infomap, mu 0.2	0.087	0.060	0.072
Infomap, mu 0.3	0.076	0.067	0.044
Infomap, mu 0.4	0.192	0.180	0.070
Louvain, mu 0.2	0.169	0.152	0.101
Louvain, mu 0.3	0.199	0.175	0.120
Louvain, mu 0.4	0.227	0.221	0.081
LPA, mu 0.2	0.242	0.250	0.117
LPA, mu 0.3	0.253	0.250	0.115

The mean, median and standard deviation are calculated across all 120 graphs in each case



(a) Infomap, $\mu = 0.2$

(b) Infomap, $\mu = 0.3$

(c) Infomap, $\mu = 0.4$

(d) Louvain, $\mu = 0.2$

(e) Louvain, $\mu = 0.3$

(f) Louvain, $\mu = 0.4$

(g) LPA, $\mu = 0.2$

(h) LPA, $\mu = 0.3$

Fig. 3 Results of the node feature experiments. Plots are of permutation importance of the metrics. Mean indicated as a black dot and median as a red dot. Lines indicate 95% bootstrapped confidence intervals

eigenvector centrality, expansion and triangle participation. We focus on reporting the significant differences for these features and provide full results in the supplementary material. Across all experiments at all μ levels, our pairwise Wilcoxon tests confirmed that these four features were significantly more important than the rest of the features, with the following exceptions:

- For Louvain at $\mu = 0.2$, clustering coefficient was not significantly different from betweenness centrality, cut ratio, or E_{out}.
- For Infomap at $\mu = 0.2$, clustering coefficient was not significantly different from degree, E_{in}, E_{out} or shortest path. Triangle participation was not significantly more important than degree or E_{in}.
- For Infomap at $\mu = 0.3$, clustering coefficient was not significantly different from closeness centrality, degree, E_{in} or shortest path.
- For LPA at $\mu = 0.2$, clustering coefficient was not significantly different from any of betweenness centrality, closeness centrality, cut ratio, degree, E_{in} or average shortest path.

These exceptions align with what can be seen qualitatively: at the lowest μ level, some other features appear to be important such as degree, E_{in} and perhaps even closeness centrality, cut ratio, E_{out} and average shortest path. However, the effect size for all of these is much smaller than for the four most prominent features, and their significance vanishes at the two higher μ levels.

Discussion

With respect to our research question RQ1 and in contradiction to our hypothesis H1, we observed that the four prominent features for predicting whether a node was difficult to classify did not depend on the algorithm used: clustering coefficient, triangle participation, eigenvector centrality, and expansion. In relation to the first two, this is not unexpected as these characteristics have previously been shown to be broadly indicative of good community structure (Harenberg et al. 2014). One could conjecture that nearly all community finding approaches would try to preserve cliques (accounting for the importance of clustering coefficient and triangle participation). In fact, cliques have often been used as seeds for constructing larger communities in the context of quite different community detection algorithms (Palla et al. 2005; Lee et al. 2010). Meanwhile, it seems reasonable that a node with many links to nodes in other communities relative to the number of nodes in its own community would be harder to classify, as it likely lies on the periphery of its community, close to one or more others (accounting for the importance of expansion).

At a surface level, the prominence of eigenvector centrality is more surprising, especially given the level of its performance. This centrality measure has similarities to PageRank (Page et al. 1999), where high values correspond to nodes at short distances from many high degree nodes. Nodes within a community's core are more likely to have high degree and to be a short distance from other high degree nodes, with edges that connect other nodes within the community. The relationship between eigenvector centrality and regions of high density within the core versus periphery of a network was recently highlighted by Jayne and Bonacich (2021). Thus, in our case high values of eigenvector centrality might correspond to an increased chance that this node forms a part of the stable community core, rather than being an unstable node on a community's periphery.

The results of our experiment with regards to changing mixing parameters μ (RQ3) indicate that these four features remain prominent. There is some evidence as well that the other features diminish in prominence as μ increases and the communities become more difficult to find. Thus, the same features are involved for explaining why a node is part of a stable core or changes communities between runs and all become statistically significant at higher mixing parameters.

Further investigation is required to find out why the important features consistently performed the best and the relative differences between them across other community finding algorithms.

Exp. 2: pairwise community membership

As mentioned in "Exp. 1: node feature experiment" section, the 0.4-LPA-node-pair experiment is omitted here as LPA classified the entire graph as one community on a number of occasions at the higher mixing parameter level.

Experiment

As with the node feature experiments, labelling all pairs directly as "same community" or "different community" results in imbalanced classes. However, we have vastly more data for the pairs of nodes than for the single nodes. Therefore, we propose undersampling both classes by randomly selecting the same number of "same community" and "different community" pairs from the available data. We choose to undersample randomly here rather than "strategically" since there are no pairs of nodes close to the threshold of 0.5 between "same" and "different" community, but choosing the highest and lowest values leads to a classification problem which is *too* easy. We select 1000 training examples for each class.

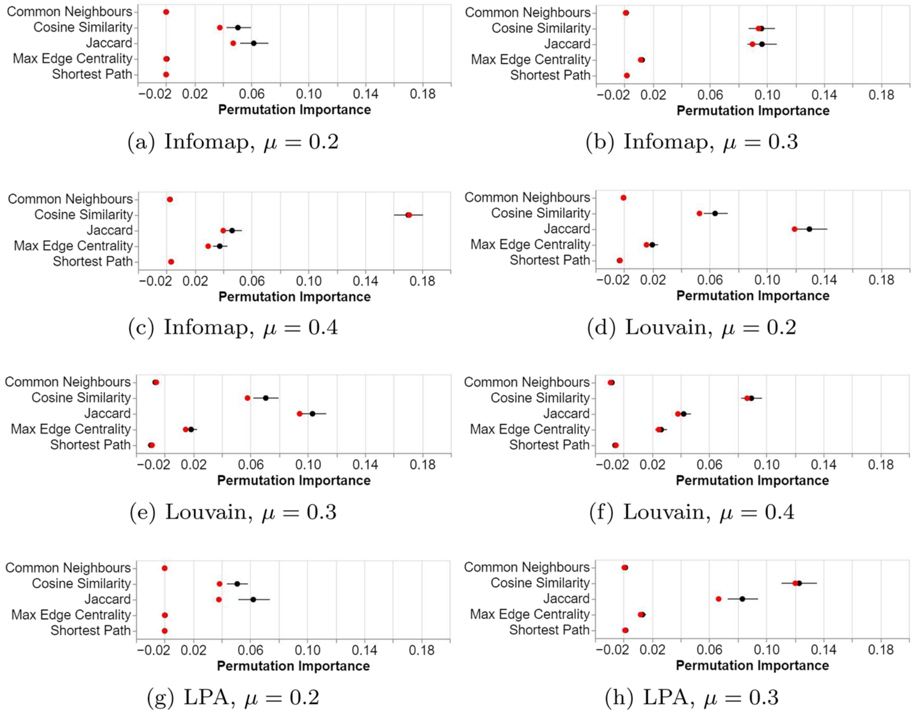
Results

As with our previous experiment, we found that two features were consistently important across the three community finding algorithms: cosine similarity and the Jaccard coefficient (displayed below in Fig. 4). We also found that the maximum edge centrality along the shortest path became more important at higher mixing parameter levels. This varied a little by algorithm; for Louvain it became important even at the lowest mixing parameter level of 0.2, however for Infomap and LPA it didn't become important until the mixing parameter of 0.3.

We focus on the significant difference between these features. For a complete set of results, we refer to the supplementary material. The pairwise Wilcoxon tests confirmed that all three were significantly different across all experiments, including for max edge centrality at the $\mu = 0.2$ level despite the small effect size.

Discussion

In contradiction to hypothesis H2, all algorithms performed similarly, with the most important features being Jaccard and cosine similarity. Both features compare the neighbourhoods of the selected nodes. Their importance is supported by the *local consistency assumption* (Zhou et al. 2004), frequently discussed in the context of instance-based classification, which asserts that neighboring instances will frequently share similar characteristics. In the context of unsupervised community detection, this corresponds to neighbouring nodes belonging to the same community, rather than having the same class label. This result



(a) Infomap, $\mu = 0.2$

(b) Infomap, $\mu = 0.3$

(c) Infomap, $\mu = 0.4$

(d) Louvain, $\mu = 0.2$

(e) Louvain, $\mu = 0.3$

(f) Louvain, $\mu = 0.4$

(g) LPA, $\mu = 0.2$

(h) LPA, $\mu = 0.3$

Fig. 4 Results of the pair feature experiments. Plots are of permutation importance of the metrics. Mean indicated as a black dot and median as a red dot. Lines indicate 95% bootstrapped confidence intervals

is also congruous with the result of the first experiment, which found similar features to be important.

In response to RQ4, maximum edge centrality proved increasingly important as the μ -level of the generated data increased. This measure, which is the maximum edge centrality measure along the shortest path between the two nodes, could be indicative of important edges that bridge two communities, i.e. the weak ties, and has been used in the past for divisive methods of community detection (Girvan and Newman 2002). The increased importance at higher values for the mixing parameter could be explained by how important local information is to determining if two nodes are within the same community. If μ is low, communities are well separated and local information almost completely describes if two nodes are in the same community. However, as μ increases, the value of local information decreases in importance. Instead, global information, such as determining if the path between the two nodes likely contains an edge that lies between communities, becomes critical in determining whether nodes belong to the same community.

Further investigation is required to find out why these features consistently performed the best and the relative differences between them across other community finding algorithms.

Discussion and limitations

General results discussion

Our hypothesis for both experiments was that the important metrics for would be dependent on the community finding algorithm, however, the same metrics were identified consistently. This indicates that there are common metrics that can be used to explain these phenomena, at least when producing explanations on the same dataset for the three algorithms tested. As our study is limited to networks generated by the LFR algorithm, the common metrics of importance could be indicative of structure produced by this method. Further work should be carried out on other datasets to see how this affects the metrics of importance. If the community finding algorithm can be taken into account, then these important metrics can also be weighted in a way that is in line with the algorithm and degree of mixing of the communities.

To validate the results of these experiments, we also ran them using Shapley values in place of permutation importance. Shapley values are well known among the explainability community, and are known to have mathematically desirable properties for producing explanations. Since our power analysis was performed using values of permutation importance, we report these as our main results, and report the results with Shapley values in the supplementary material.

In the case of the node experiments, we see clearly that for Louvain and LPA, the same four node features are more important with increasing mu value for Shapley values as we saw with permutation importance: clustering coefficient, expansion, eigenvector centrality, and triangle participation. With Infomap, there is some variation; the same four features are seen to be important, though not at all mu values. In addition, E In is shown to be important, as are Degree and Closeness Centrality at the lower mu values.

In the case of the node-pair experiments, the same trends are seen for Shapley values as for permutation importance across all algorithms. Jaccard and cosine similarity are consistently the most important, with max edge centrality increasing in importance with rising mu value.

The features used in both experiments, which are well understood by the network analysis community, can be used to gain a greater understanding of community structure in online social networks. For public health applications, these metrics may be used to understand phenomena such as social contagion (Brown et al. 2017; Valente and Yon 2020) and to plan interventions (Valente 2012), but their practical use remains future work. In future studies, we plan on integrating visualisation methods and performing studies to ensure that approaches such as the ones proposed here have impact in explaining network phenomena to address real-world user needs. We envisage these results could be used with a visualisation system where the communities assigned by an algorithm can be explored by selecting individual nodes or pairs of nodes to understand their community assignment. When a node flips between different communities on consecutive runs of the community finding algorithm, important feature values such as those identified in this work could be visually reported and compared relative to other nodes in the same or in different communities for further study by an expert.

Consensus clustering (Lancichinetti and Fortunato 2012b) is a way of dealing with nodes that are difficult to classify: run the community finding algorithm many times and determine the average result of these runs. Given that this study indicates that the metrics to determine if nodes are easy or hard to cluster by community finding algorithms are consistent across algorithm, the results of consensus clustering approaches could be augmented with these metrics to help determine which community these nodes should be clustered into, but this remains an area of future work. Also, values for these metrics could be used to seed the stable core of a community and then find other nodes that are less easy to cluster. This approach could lead to other partitioning algorithms or potentially overlapping community finding algorithms where "hard to cluster" nodes are partially contained by multiple communities. However, the effectiveness of such an approach would still need to be evaluated.

Although partitioning algorithms are usually used to find communities in networks in public health settings, overlapping community finding algorithms (Lancichinetti et al. 2011; Pallaand et al. 2005) can assign a single node to multiple communities. The studies that we present here suggest metrics used in social network analysis can be used to explain partitioning algorithms. In future work, it would be interesting to see if these, or other metrics, extend to overlapping community finding.

Limitations and future work

Although there were benefits to our use of synthetic LFR data, these networks are ultimately an approximation for real data. As discussed previously, the use of real data for this analysis would have been tricky due to the lack of large datasets of networks with consistent structure over which we could draw statistically significant conclusions. Additionally, we would not have been able to vary parameters such as the mixing parameter μ to observe their effect on the results. Nevertheless, it is a limitation of this study that we are only able to confirm these results across a dataset of synthetic networks based on the parameters specified in an important community finding experiment (Lancichinetti and Fortunato 2009) on three community finding algorithms. In future work, the performance of these informative features could be verified on a smaller number of real networks, to investigate whether this affects the metrics of greatest importance. Even in the case where the metrics of greatest importance are heavily dependent on the dataset, the methodology presented here could beneficially be applied to new settings in order to gain insight into complex networks relevant to different applications, such as social, biological, and financial (Avdjiev et al. 2019; Giudici et al. 2017) networks.

Another area for future work would be to consider applying the more recent Shapley-Lorenz approach to explainability (Giudici and Raffinetti 2021) in place of using Shapley values or permutation importance; this approach is well-suited to settings such as ours, where the response variable is categorical.

Conclusion

This paper presents the results of two experiments designed to move towards explainable community finding and explainable network analysis in general. Despite the different methods used by the algorithms in our study, consistent social network analysis metrics can be used to explain community structure in a post-hoc manner for these three algorithms on LFR networks. The results of our study indicate that commonly understood metrics used for network analysis can be used by an expert to explain community structure, bringing benefits to application areas where network data is prevalent, from computational social science (Lazer et al. 2009) to public health studies (Luke and Harris 2007).

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1007/s41109-022-00515-6.

Additional file 1. Supplementary material is made available containing detailed descriptions and results plots of statistical analysis.

Acknowledgements

We thank the extensive support provided by the AIMLAC CDT team and its administrators, in addition to the computer science department at Swansea University's Computational Foundry.

Author contributions

SS proposed the methods, implemented the code, and performed the experiments. SS, DG and DA wrote, read, and approved the manuscript.

Funding

This work is supported by the UKRI AIMLAC CDT, funded by Grant EP/S023992/1 and by the UKRI EPSRC Grant EP/V033670/1. The research was also partly supported by Science Foundation Ireland (SFI) under Grant No. SFI/12/RC/2289_P2.

Availability of data and materials

Data and code implementations for the main experiments are available in the GitHub repository at https://github.com/sophiefsadler/community_finding. A zip file containing results and python scripts forstatistical analysis can also be found on OSF at: https://osf.io/g4bwt/.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 25 April 2022 Accepted: 31 October 2022
Published online: 08 December 2022

References

Adadi A, Berrada M (2018) Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access 6:52138–52160

Avdjiev S, Giudici P, Spelta A (2019) Measuring contagion risk in international banking. J Financ Stab. https://doi.org/10.1016/j.jfs.2019.05.014

Bach S, Binder A, Montavon G, Klauschen F, Müller K-R, Samek W (2015) On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS ONE 10(7):1–46. https://doi.org/10.1371/journal.pone.0130140

Ben-David S, Pál D, Simon HU (2007) Stability of k-means clustering. In: Bshouty NH, Gentile C (eds) Learning theory, pp 20–34

Blondel VD, Guillaume J-l, Lefebvre E (2008) Fast unfolding of communities in large networks, pp 1–12. arXiv:0803.0476v2

Bonacich P (1986) Power and centrality: a family of measures. Am J Sociol 92(5):1170–1182

Bothorel C, Brisson L, Lyubareva I (2020) How to choose community detection methods in complex networks: the case study of Ulule crowdfunding platform

Brandes U (2001) A faster algorithm for betweenness centrality. J Math Sociol 25(2):163–177

Brown RC, Fischer T, Goldwich AD, Keller F, Young R, Plener PL (2017) #cutting: non-suicidal self-injury (NSSI) on Instagram. Psychol Med 48(2):337–346. https://doi.org/10.1017/s0033291717001751

Chakraborty T, Srinivasan S, Ganguly N, Bhowmick S, Mukherjee A (2013) Constant communities in complex networks. Nat Sci Rep 3(1):1825. https://doi.org/10.1038/srep01825

Chakraborty T, Dalmia A, Mukherjee A, Ganguly N (2017) Metrics for community analysis: a survey. ACM Comput Surv. https://doi.org/10.1145/3091106

Dao VL, Bothorel C, Lenca P (2020) Community structure: a comparative evaluation of community detection methods. Netw Sci 8(1):1–41. https://doi.org/10.1017/nws.2019.59

Flake GW, Lawrence S, Giles CL (2000) Efficient identification of web communities. In: Proceedings of the Sixth ACM SIGKDD international conference on knowledge discovery and data mining (KDD '00), pp 150–160

Fong R, Vedaldi A (2017) Interpretable explanations of black boxes by meaningful perturbation. CoRR. arXiv:1704.03296

Fortunato S (2010) Community detection in graphs. Phys Rep 486(3–5):75–174

Francisco AP, Oliveira AL (2011) On community detection in very large networks. In: da Costa FL, Evsukoff A, Mangioni G, Menezes R (eds) Complex networks. Springer, Berlin, pp 208–216

Freeman LC (1979) Centrality in networks: I. conceptual clarification. Soc Netw 1:215–239

Gesell SB, Barkin SL, Valente TW (2013) Social network diagnostics: a tool for monitoring group interventions. Implement Sci. https://doi.org/10.1186/1748-5908-8-116

Ghasemian A, Hosseinmardi H, Clauset A (2019) Evaluating overfit and underfit in models of network community structure. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/tkde.2019.2911585

Girvan M, Newman MEJ (2002) Community structure in social and biological networks. Proc Natl Acad Sci U S A 99(12):7821–7826. https://doi.org/10.1073/pnas.122653799arXiv:01121

Giudici P, Raffinetti E (2021) Shapley-Lorenz explainable artificial intelligence. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2020.114104

Giudici P, Sarlin P, Spelta A (2017) The interconnected nature of financial systems: Direct and common exposures. J Bank Finance. https://doi.org/10.1016/j.jbankfin.2017.05.010

Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D (2018) A survey of methods for explaining black box models. ACM Comput Surv (CSUR) 51(5):1–42

Harenberg S, Bello G, Gjeltema L, Ranshous S, Harlalka J, Seay R, Padmanabhan K, Samatova N (2014) Community detection in large-scale networks: a survey and empirical evaluation. Wiley Interdiscip Rev Comput Stat 6(6):426–439

Hunter RF, McAneney H, Davis M, Tully MA, Valente TW, Kee F (2015) hidden social networks in behavior change interventions. Am J Public Health 105(3):513–516. https://doi.org/10.2105/AJPH.2014.302399

Hunter RF, de la Haye K, Murray JM, Badham J, Valente TW, Clarke M, Kee F (2019) Social network interventions for health behaviours and outcomes: a systematic review and meta-analysis. PLoS Med 16(9):1–25. https://doi.org/10.1371/journal.pmed.1002890

Jaccard P (1912) The distribution of flora in the alpine zone. New Phytol 11(2):37–50

Jayne Bienenstock E, Bonacich P (2021) Eigenvector centralization as a measure of structural bias in information aggregation. J Math Sociol 46:1–19

Keane MT, Kenny EM (2019) How case-based reasoning explains neural networks: a theoretical analysis of XAI using post-hoc explanation-by-example from a survey of ANN-CBR twin-systems. In: Proceedings of international conference on case-based reasoning (ICCBR'19). Springer, pp 155–171

Lancichinetti A, Fortunato S (2009) Community detection algorithms: a comparative analysis. Phys Rev E - Stat Nonlinear Soft Matter Phys 80(5):1–12. https://doi.org/10.1103/PhysRevE.80.056117arXiv:0908.1062

Lancichinetti A, Fortunato S (2012a) Consensus clustering in complex networks. Sci Rep 2(1):1–7

Lancichinetti A, Fortunato S (2012b) Consensus clustering in complex networks. Nat Sci Rep 2:336

Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. Phys Rev E - Stat Nonlinear Soft Matter Phys 78(4):1–6. https://doi.org/10.1103/PhysRevE.78.046110arXiv:0805.4770

Lancichinetti A, Radicchi F, Ramasco JJ, Fortunato S (2011) Finding statistically significant communities in networks. PLoS ONE 6(4):1–18. https://doi.org/10.1371/journal.pone.0018961

Lazer D, Pentland AS, Adamic L, Aral S, Barabasi AL, Brewer D, Christakis N, Contractor N, Fowler J, Gutmann M et al (2009) Life in the network: the coming age of computational social science. Science 323(5915):721

Lee A, Archambault D (2016) Communities found by users—not algorithms. In: Proceedings of the 2016 CHI conference on human factors in computing systems, pp 2396–2400. https://doi.org/10.1145/2858036.2858071

Lee C, Reid F, McDaid A, Hurley N (2010) Detecting highly overlapping community structure by greedy clique expansion. In: Proceedings of the 4th international workshop on social network mining and analysis (SNA-KDD), pp 33–42

Lipton ZC (2018) The mythos of model interpretability: in machine learning, the concept of interpretability is both important and slippery. Queue 16(3):31–57

Loyola-Gonzalez O, Gutierrez-Rodríguez AE, Medina-Pérez MA, Monroy R, Martínez-Trinidad JF, Carrasco-Ochoa JA, Garcia-Borroto M (2020) An explainable artificial intelligence model for clustering numerical databases. IEEE Access 8:52370–52384

Luke DA, Harris JK (2007) Network analysis in public health: history, methods, and applications. Annu Rev Public Health 28:69–93

Lundberg SM, Lee S-I (2017) A unified approach to interpreting model predictions. In: Proceedings of the 31st international conference on neural information processing systems. NIPS'17. Curran Associates Inc., Red Hook, pp 4768–4777

Morichetta A, Casas P, Mellia M (2019) EXPLAIN-IT: towards explainable AI for unsupervised network traffic analysis. In: Proceeedings of 3rd ACM CoNEXT workshop on big data, machine learning and artificial intelligence for data communication networks, pp 22–28

Page L, Brin S, Motwani R, Winograd T (1999) The pagerank citation ranking: bringing order to the web. Technical Report 1999-66, Stanford InfoLab

Palla G, Derényi I, Farkas I, Vicsek T (2005) Uncovering the overlapping community structure of complex networks in nature and society. Nature 435(7043):814–818

Pallaand G, Derényi I, Farkas I, Vicsek T (2005) Uncovering the overlapping community structure of complex networks in nature and society. Nature 435:814–818. https://doi.org/10.1038/nature03607

Park M, Lawlor MC, Solomon O, Valente TW (2020) Understanding connectivity: the parallax and disruptive-productive effects of mixed methods social network analysis in occupational science. J Occup Sci. https://doi.org/10.1080/14427591.2020.1812106

Peel L, Larremore DB, Clauset A (2017) The ground truth about metadata and community detection in networks. Sci Adv. https://doi.org/10.1126/sciadv.1602548

Radicchi F, Castellano C, Cecconi F, Loreto V, Parisi D (2004) Defining and identifying communities in networks. PNAS 101(9):2658–2663

Raghavan UN, Albert R, Kumara S (2007) Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E 76:036106. https://doi.org/10.1103/PhysRevE.76.036106

Ribeiro MT, Singh S, Guestrin C (2016) "Why should I trust you?" Explaining the predictions of any classifier. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, pp 1135–1144. https://doi.org/10.1145/2939672.2939778. arXiv:1602.04938v3

Rosvall M, Bergstrom CT (2008) Maps of random walks on complex networks reveal community structure. Proc Natl Acad Sci U S A 105(4):1118–1123. https://doi.org/10.1073/pnas.0706851105arXiv:0707.0609

Rudin C (2019) Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat Mach Intell 1(5):206–215

Saarela M, Jauhiainen S (2021) Comparison of feature importance measures as explanations for classification models. SN Appl Sci 3(2):1–12

Shapley LS (2016) In: Kuhn HW, Tucker AW (eds) 17. A value for n-person games. Princeton University Press, pp 307–318. https://doi.org/10.1515/9781400881970-018

Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22(8):888–905

Strehl A (2002) Relationship-based clustering and cluster ensembles for high-dimensional data mining. Master's Thesis, The University of Texas at Austin

Sundararajan M, Taly A, Yan Q (2017) Axiomatic attribution for deep networks. In: Proceedings of the 34th international conference on machine learning—volume 70. ICML'17, pp 3319–3328

Valente TW (2012) Network interventions. Science 337(6090):49–53. https://doi.org/10.1126/science.1217330

Valente TW, Yon GGV (2020) Diffusion/contagion processes on social networks. Health Educ Behav 47(2):235–248. https://doi.org/10.1177/1090198120901497

Valente TW, Fujimoto K, Unger JB, Soto DW, Meeker D (2013) Variations in network boundary and type: a study of adolescent peer influences. Soc Netw 35(3):309–316. https://doi.org/10.1016/j.socnet.2013.02.008

Valente TW, Palinkas LA, Czaja S, Chu K-H, Brown CH (2015) Social network analysis for program implementation. PLoS ONE. https://doi.org/10.1371/journal.pone.0131712

von Luxburg U (2010) Clustering stability: an overview. Found Trends Mach Learn 2(3):235–274. https://doi.org/10.1561/2200000008

Wachter S, Mittelstadt B, Russell C (2017) Counterfactual explanations without opening the black box: automated decisions and the GDPR. Harv J Law Technol 31:841

Watts DJ, Strogatz SH (1998) Collective dynamics of small-world networks. Nature 393:440–442

Ying R, Bourgeois D, You J, Zitnik M, Leskovec J (2019) GNNExplainer: a tool for post-hoc explanation of graph neural networks. CoRR. arXiv:1903.03894

Yuan H, Tang J, Hu X, Ji S (2020) XGNN: Towards model-level explanations of graph neural networks. CoRR. arXiv:2006.02587

Zhou D, Bousquet O, Lal TN, Weston J, Schölkopf B (2004) Learning with local and global consistency. In: Advances in neural information processing systems, pp 321–328

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.