Squillace *et al. Applied Network Science* (2025) 10:39
https://doi.org/10.1007/s41109-025-00726-7

Applied Network Science

RESEARCH Open Access

Approximate regular equivalence by partition refinement

Giuseppe Squillace[1*], Mirco Tribastone[1†], Max Tschaikowski[2†] and Andrea Vandin[3†]

†Mirco Tribastone, Max Tschaikowski and Andrea Vandin have contributed equally to this work.

*Correspondence:
Giuseppe Squillace
giuseppe.squillace@imtlucca.it
[1]IMT School for Advanced Studies Lucca, Lucca, Italy
[2]Aalborg University, Aalborg, Denmark
[3]Sant'Anna School of Advanced Studies, Pisa, Italy

Abstract
Regular equivalence aims to identify nodes that have links to nodes that are themselves equivalent, and is considered to capture key relational properties in networks. Exact equivalences are notoriously difficult to emerge in real-world networks because of the rather stringent criteria required. This has motivated the development of approximate approaches, which, however, do not scale well to large networks. In this paper, we present a new method to compute approximate regular equivalences for weighted networks based on a partition refinement algorithm. This is parameterized by a tolerance ε that determines the extent to which two nodes may be deemed equivalent. We also show an asymptotic result for networks with power-law distribution that analytically provides a partition of approximately equivalent nodes. Using a number of benchmark networks, we show that our method outperforms the state of the art in terms of precision and running time. When the asymptotic partition is used to initialize the partition refinement algorithm for real-world networks, it avoids the problem of aggressive clustering that affects binary networks.

Keywords Networks, Approximate regular equivalence, Partition refinement

Introduction

Rooted in social sciences, the notions of node equivalences are useful tools to uncover and understand roles and relations in networks across a variety of domains, including biology (Luczkovich et al. 2003), economics (Smith and White 1992), social media (Kane et al. 2014), collaboration (Ahuja 2000), and management (Orsenigo et al. 1997). A role summarizes the structural position of a node in a network based on its connection patterns with other nodes. For example, in an organizational network, managers may be assigned the same role due to their common function of supervising employees.

Identifying roles allows practitioners to reinterpret a network in terms of role interactions, rather than individual node identities, making complex graphs easier to analyze and more interpretable. This is extremely useful in the modern era, where the abundance of data gives rise to large and intricate networks that are challenging to analyze (Bedru et al. 2020). Node equivalences are also particularly valuable in role-based machine learning tasks, such as node and link classification (Jin et al. 2019b; Ahmed et al. 2017), user stitching (Jin et al. 2019a), anomaly detection (Akoglu et al. 2015), and many others (Rossi et al. 2020). For instance, the notion of role can be exploited in anomaly detection to identify patterns of interaction associated with anomalous behaviors (Rossi et al. 2013).

Over the years, social sciences developed different notions of node equivalence that are able to capture different aspects of the role of a node in a network. Structural equivalence identifies nodes that are connected to the same neighbors (Lorrain and White 1971). In automorphic equivalence (Borgatti and Everett 1992), nodes are related through graph-theoretic properties such as in-/out-degree and centralities. This is relaxed by regular equivalence, which aims to identify nodes that play the same role in the network even if they do not share neighbors by requiring that any two regularly equivalent nodes are both connected to nodes that are themselves regularly equivalent (White and Reitz 1983). In this paper, we focus on regular equivalence because it represents an advance in capturing key features of the relational role concept (Borgatti and Everett 1993).

Node equivalences can be defined for both binary (unweighted) and weighted networks. In either case, it is recognized that *exact* notions may be too strict to discover useful relations in practice, for example under the presence of noisy data (e.g., Reichardt and White (2007)). This has motivated the development of *approximate* relations that relax the assumptions on when two nodes can be deemed equivalent. These methods can be categorized into indirect and direct, the former being developed earlier. The indirect approach computes a similarity matrix among the nodes, which is then partitioned using hierarchical clustering (White and Reitz 1983; Borgatti and Everett 1993). Direct approaches do not require a similarity matrix. Deterministic blockmodeling sorts the adjacency matrix in blocks (Doreian et al. 2005), each identifying a cluster of nodes in the network, and specifies the links from one cluster to another through an optimization problem. Stochastic blockmodeling is a generative model based on statistical properties of random graphs (Funke and Becker 2019; Peixoto 2019).

Overall, the current state of the art makes both indirect and direct approaches difficult to use in large networks. We present a novel algorithm for the computation of approximate regular equivalences in weighted networks that can scale to larger networks. Our algorithm can be considered as a direct method in that it does not compute a similarity matrix. Differently from blockmodeling, however, it does not require the user to choose the number of clusters. Instead, it is parameterized by a tolerance ε that, roughly speaking, determines the degree with which nodes can be deemed approximately regularly equivalent.

More in detail, our approach builds on a well-understood interpretation of regular equivalence as a bisimulation, a fundamental concept in (theoretical) computer science (Marx and Masuch 2003). Here, in particular, we relate approximate regular equivalence to a notion called ε-backward differential equivalence (ε-BDE) (Cardelli et al. 2018, 2023), originally developed for systems of ordinary differential equations with polynomial right-hand sides. Our intuition is to associate the adjacency matrix of network A with a linear system of differential equations $\dot{x} = Ax$ and establish a formal correspondence between regular equivalence in the former and backward equivalence on the latter.

Notably, this connection ultimately allows us to compute approximate regular equivalences by partition refinement, a widely adopted algorithm that iteratively splits the blocks of a candidate initial partition until the criteria for the desired equivalence are met (Paige and Tarjan 1987; Baier et al. 2000). The algorithm for backward equivalence, however, cannot be directly reused for two reasons. The first is of mathematical nature and concerns the fact that regular equivalence is related to backward equivalence on both the network A and its transpose A^T. The second reason is that the algorithm constructs equivalence classes through a transitive closure of nodes that are pairwise ε-similar, which may lead to aggressive aggregation in the output (cf. Example 1). A similar phenomenon occurs in indirect methods: especially with binary networks, they could fail the analysis by identifying all nodes as approximately regularly equivalent (Borgatti and Everett 1989).

To cope with both issues, we propose an iterative approach where the ε-BDE algorithm executes as an inner step for both A and A^T, while the iterations consider increasingly larger values of ε to avoid aggregating too much. In addition to ε, another important user-tuneable input of the algorithm is the candidate initial partition to be split, as with any partition refinement method. In general, the largest equivalence (i.e., the coarsest partition) is computed by initializing the algorithm with a singleton partition where all nodes are in the same block. In many applications, however, the freedom in choosing an arbitrary initial partition may be used to encode certain requirements or a-priori knowledge, e.g., to isolate a node or to prepartition nodes according to given roles.

In this paper, we exploit this feature to provide a candidate initial partition for the relevant class of Barabasi-Albert (BA) networks (Barabási and Bonabeau 2003), which are well-known to fit real-world datasets appropriately. Intuitively, such networks are particularly challenging for our algorithm because their power-law distributed degrees may lead to relatively low values of ε to collapse many low-degree nodes, with the risk of losing much information in the resulting equivalence. The initial BA partition accounts for it because we prove that, on average, *it already is* an ε-BDE partition for sufficiently large BA networks.

Using a prototypical implementation, we conduct an experimental evaluation of our algorithm on binary and weighted networks from the literature to show the following:

1. Our algorithm provides consistently more accurate partitions than both direct and indirect methods using the same level of granularity (number of clusters), as indicated by statistics on the centralities of approximately regular equivalent nodes.
2. Our algorithm can practically scale to larger networks than direct and indirect methods.
3. Applied to concrete instances of binary networks with skewed distribution, our asymptotic BA partition can avoid excessive clustering and can also be used as an appropriate pre-partition for indirect methods that are known to identify all nodes in the same block (Borgatti and Everett 1989).

The rest of the paper is structured as follows. Section 2 reviews the related work. Instead, Sect. 3 first reviews the notion of regular equivalence, bisimulation, BDE and ε-BDE. Then, after showing that BDE implies regular equivalence, it introduces iterative ε-BDE and establishes its worst-case complexity. The section then concludes by introducing the BA partition and by showing the corresponding asymptotic result. The framework is then evaluated against state-of-the-art approaches on real-world networks in Sect. 4. Finally, Sect. 5 concludes the paper.

Related work

REGE is the first indirect method for regular equivalence based on an iterative point-scoring procedure that builds a similarity matrix for both binary and weighted networks (White and Reitz 1983). Its time complexity is $\mathcal{O}(n^5)$, where n is the number of nodes. CATREGE is an improved version for binary and categorical networks with time complexity $\mathcal{O}(n^3)$ (Borgatti and Everett 1993), which, moreover, allows specifying an initial partition of the nodes to constrain the solution and improve the results. Despite this, the current implementation of CATREGE limits its use to networks with at most a few hundred nodes (Borgatti et al. 2002). To build a partition of approximately regularly equivalent nodes with an indirect approach, typical hierarchical clustering techniques are based on single-link, complete-link and Ward's method (Ziberna 2008).

While stochastic blockmodeling is based on a generative model, deterministic blockmodeling is more similar to indirect approaches in that it performs clustering. Usually, the number of clusters is a parameter set beforehand, and an optimization problem is set up to minimize a certain objective function of discrepancy with respect to ideal blocks by allowing permuting rows and the columns of the adjacency matrix (Doreian et al. 2005). In binary networks, the binary blockmodeling defines the discrepancy in terms of ideal blocks specified as 0–1 patterns (Doreian et al. 2005). For weighted networks, dichotomization is used; given a threshold it sets all values in the matrix under (resp., over) the threshold with 0 (resp., 1), so that blockmodeling for binary networks can be used. Since dichotomization may lead to loss of information, in Žiberna (2007) a new approach, valued blockmodeling, is proposed trying to define ideal blocks in terms of weighted networks (*f-regular* equivalence). This is shown to be more robust than dichotomization, but it requires the choice of a parameter that depends on the strength of the links. Often, the estimation is based on previous knowledge of the network (Žiberna 2007); some general estimations are possible, like the median or the mode of the weights, but no guarantees can be provided (Matjasiĉ et al. 2020). A different approach is the homogeneity blockmodeling proposed in Žiberna (2007). Its aim is to create blocks where a measure of the variability of the links is minimal.

Since exact methods that guarantee globally optimal solutions are computationally expensive (Brucker 1978), heuristic methods of local search are generally employed (Doreian et al. 2005). However, they lack optimality guarantees (Doreian et al. 2005). As shown also by our numerical experiments, such optimization-based approach cannot scale, in practice, to networks larger than a few hundred nodes.

To overcome the limitations of rigid equivalence notions, several studies have introduced relaxed versions of classical role equivalence (Everett 1985; Sailer 1978; Everett et al. 1990) or proposed more flexible approaches to blockmodeling (Brandes and Lerner 2010). While conceptually aligned with our work, these approaches are primarily theoretical and, to the best of our knowledge, lack publicly available implementations.

The role identification problem was also explored within the machine learning community (Rossi et al. 2020). In this context, role-based embedding methods, such as Ribeiro et al. (2017), Donnat et al. (2018), Tu et al. (2018) and Nikolentzos and Vazirgiannis (2019), extract salient structural features from graphs with the aim of encoding nodes in a lower dimensional space, where nodes with similar roles are positioned close in the embedding space. These methods are conceptually and methodologically different from our approach. Rather than identifying specific equivalences within a graph, they produce a representation of the nodes into a metric space. Since the focus of this work is the explicit computation and analysis of graph partition, we do not include these methods in our evaluation. A fair comparison with respect to role-based embedding techniques can be found in Squillace et al. (2024), where we show how partitions computed with backward equivalence can be leveraged to build network embeddings.

In Marx and Masuch (2003), regular equivalence is related to the classic notion of bisimulation for transition systems (Milner 1982). In this paper, instead, we related *approximate* regular equivalence to ε-backward differential equivalence (ε-BDE). BDE, its exact counterpart, conservatively generalizes exact lumpability, defined for Markov chains (Buchholz 1994), to systems of ordinary differential equations (ODEs) with polynomial right-hand sides, relating its variables. Exact lumpability, in turn, can be seen as an instance of bisimulation for transition systems labeled with probability distributions on the arcs (and, indeed, is also termed *backward bisimulation* in the literature). ε-BDE relaxes the requirements of BDE by allowing ODE variables to be related under some given tolerance ε. Such tolerance parameter is an input to a partition refinement algorithm that is shown here to run in $\mathcal{O}(n^4)$, where *n* is the number of nodes. Despite this worst-case complexity is higher than that of CATREGE, we show in the experimental section that, in practice, our algorithm is able to scale consistently better than the other methods.

In this direction, we mention the work (Petrov and Tognazzi 2021), where the authors employed the exact and approximate BDE to reduce multilayer networks. More in detail, they propose an ODE system to encode the iterative scheme to compute the eigenvalue centrality on binary undirected multilayer networks. Consequently, they compute the equivalences to find exact and approximate role assignments. The novelty of our approach is the definition of the iterative scheme that avoids aggressive aggregation on single-level networks. This scheme can be applied straightforwardly to the proposed ODE system, extending our approach for multilayer networks.

Methods

Background

We define a network with *n* nodes by its adjacency matrix $A = (a_{i,j}) \in \mathbb{R}^{nxn}$ where each component a_{ij} denotes the weight of the link from node *i* to node *j*; as usual, we call a network *binary* if $a_{i,j} \in \{0, 1\}$ and *undirected* if A is symmetric. Nodes are labelled 1, 2,..., *n*. Intuitively, regular equivalence relates nodes equivalent whenever these have identical links to and from regularly equivalent nodes (White and Reitz 1983). For the purposes of this paper, it is convenient to express it via the classic notion of bisimulation (Marx and Masuch 2003), as recalled next.

Definition 1 For an adjacency matrix $A \in \{0, 1\}^{n\times n}$, we write $i \to j$ whenever $a_{i,j} = 1$.

- A equivalence relation $\mathcal{R}$ is a bisimulation of A if for any $(i, j) \in \mathcal{R}$ and link $i \to i'$, there exists a link $j \to j'$ such that $(i', j') \in \mathcal{R}$
- A relation $\mathcal{R}$ is a regular equivalence of A whenever $\mathcal{R}$ is a bisimulation of A and A^T.
- We set $\mathcal{H}_{\mathcal{R}} = \{1, \ldots, n\}/\mathcal{R}$ for any equivalence relation $\mathcal{R}$.

To illustrate the correspondence of bisimulation and regular equivalence, let us consider the network in Fig. 1 composed by 7 nodes arranged in a tree structure. To make the example more concrete, we assume that edges represent the relationship *"parent of"*. In this context, node 1 is parent of nodes 2 and 3, while nodes 2 and 3 are themselves parents of 4,5 and 6,7, respectively. Let us consider the partition $\mathcal{H}_{\mathcal{R}} = \{\{1\}, \{2,3\}, \{4,5,6,7\}\}$, where nodes within each block satisfy the bisimulation condition. If we interpret each block as a distinct role, the notion of bisimulation aligns with that of regular equivalence. Indeed, the partition shown in the example of Fig. 1 illustrates that two nodes are regularly equivalent if they are connected to nodes that play the same roles, rather than to the same individual nodes.

In this example, the three roles $\{1\}$, $\{2,3\}$, and $\{4,5,6,7\}$ admit a clear social interpretation based on the edge semantics. Specifically, nodes 4,5,6, and 7 can be considered as individuals without children, 2 and 3 are parents because they have at least one child, and finally, 1 is a grandparent because its children are themselves parents.

The definition naturally extends to weighted networks by, essentially, treating every distinct weight as a categorical label and requiring regular equivalences on all such labels (e.g., Ziberna (2008)).

Definition 2 Let $A \in \mathbb{R}^{n\times n}$ be a weighted adjacency matrix with L distinct weights such that $A = \sum_{l=1}^{L} w_l A^l$, where $w^l \in \mathbb{R}$ and $A^l \in \{0,1\}^{n\times n}$. Then, $\mathcal{R}$ is a regular equivalence of A if $\mathcal{R}$ is a regular equivalence of $A^1, \ldots, A^L$.

Regular equivalence allows the same link of a node to match more than one link of a regularly equivalent one. Although regular equivalences can be used to identify roles in a network, they assume that nodes with the same role must follow a precise pattern of interactions. As we will show in the experimental section, this constraint becomes particularly limiting in the case of real networks that are usually affected by noise and uncertainty in the data. This motivates the need of an approximate version of regular equivalence that relaxes the strict definition in order to group nodes that exhibit similar, rather than identical, interaction patterns.

BDE

Our approach recalls the notion of bisimulation for dynamical systems that, informally, relates nodes that have the *same cumulative degree* with respect to blocks of nodes in the same equivalence class. Here, given a matrix $A \in \mathbb{R}^{n\times n}$ one considers an associated linear system of ordinary differential equations (ODEs) in the form $\dot{x} = Ax$, where $\dot{x}$ denotes the time derivative of the solution x. Backward differential equivalence (BDE) is defined for polynomial differential equations (Cardelli et al. 2017a), thus including

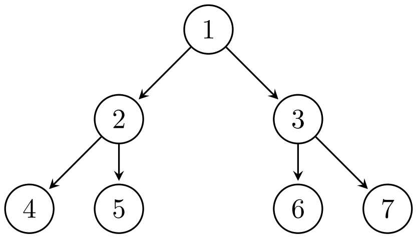

Fig. 1 Example of a network. Edges represent the relationshiop *"parent of"*. Bisimulation, regular equivalence and BDE identifies the same partition $\{\{1\}, \{2,3\}, \{4,5,6,7,8\}\}$ with 3 different roles

the case above. Although it concerns ODE solutions, and thus it does not apply directly to the problem of finding regular equivalences, it allows us to establish a connection between BDE and regular equivalence.

Definition 3 For an adjacency matrix $A \in \mathbb{R}^{n \times n}$, an equivalence relation $\mathcal{R}$ is called backward equivalence (BDE) when

$$\sum_{H' \in \mathcal{H}_{\mathcal{R}}} \left| \sum_{k \in H'} a_{k,i} - \sum_{k \in H'} a_{k,j} \right| = 0$$

for all $H, H' \in \mathcal{H}_{\mathcal{R}}$ and $i, j \in H$.

Let us observe that BDE matches cumulative in-degrees towards equivalence classes (hence the term *backward*). A BDE relation for the transpose adjacency matrix A^T corresponds to a *forward* equivalence that matches out-degrees (Baier et al. 2000; Valmari and Franceschinis 2010; Bacci et al. 2021; Buchholz 1994). We start with a simple yet crucial statement that relates regular equivalence with BDE.[1]

Theorem 1 Given $A = \sum_{l=1}^{L} w^l A^l$ with $A^l \in \{0,1\}^{n \times n}$, assume that $\mathcal{R}$ is a BDE of A^l and $(A^l)^T$, for all $1 \leq l \leq L$. Then, $\mathcal{R}$ is a regular equivalence of A and each $A^1, \ldots, A^L$.

In the example shown in Fig. 1, the BDE definition, like regular equivalence and bisimulation, is suitable to identify the three roles of parenthood. This can be verified by examining the distribution of edges with respect to the equivalence classes. Specifically, nodes 4,5,6, and 7 are characterized by having one incoming edge and no outgoing edges. Nodes 2 and 3 have two outgoing edges toward nodes in block $\{4, 5, 6, 7\}$ and one incoming edge from node 1. Finally, node 1 is defined by having only outgoing edges, which are directed toward nodes in block $\{2, 3\}$.

Unfortunately, Theorem 1 provides only a sufficient condition for regular equivalence, and its assumption cannot be relaxed to $\mathcal{R}$ being the BDE of A and A^T only. Indeed, partition $\{\{1, 2, 3\}, \{4, 5, 6\}\}$ of the network depicted in Fig. 2 can be shown to be a BDE of A and A^T, where $A = A^1 + \ldots + A^4$. At the same time, however, it is not a regular equivalence of A because it is not a regular equivalence of A^1. Indeed, nodes 1 and 3 have black links, while node 2 has no black links. It can also be noted that, in contrast to regular equivalence, BDE requires the rather strict assumption regarding equal degrees of related nodes.

ε-BDE

To cope with the strict assumption imposed by BDE, we define a notion of *approximate* BDE, called ε-BDE, where the equality between the degrees of related nodes in Definition 4 is relaxed by inequalities up to a given tolerance ε. Since approximately related

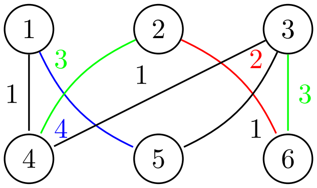

Fig. 2 Example of a network where BDE of *A* and A^T does not imply regular equivalence

---

[1] Proofs are given in the appendix.

nodes will not have equal in- and out-degrees in general, ε-BDE becomes an alternative method to compute an approximate regular equivalence.

In the following, we recast the notion of ε-BDE, developed for polynomial differential equations (Cardelli et al. 2018), to the linear case related to an adjacency matrix.

Definition 4 (ε-*BDE*) For an $A \in \mathbb{R}^{n\times n}$ and a partition $\mathcal{H}$, we write $i \sim_{A,\mathcal{H},\varepsilon} j$ whenever there exists an $H \in \mathcal{H}$ with $i, j \in H$ such

$$\sum_{H'\in\mathcal{H}} \Big| \sum_{k\in H'} a_{k,i} - \sum_{k\in H'} a_{k,j} \Big| \leq \varepsilon.$$

A partition $\mathcal{H}$ is called ε-BDE if $\mathcal{H} = \{1, \ldots, n\}/ \sim^*_{A,\mathcal{H},\varepsilon}$. Here, the asterisk denotes the equivalence closure of a relation.

The approximate definition enables the identification of regular equivalences that elude detection by the exact BDE method. To illustrate this capability, consider the example network depicted in Fig. 3. Here, we consider all the edges with the same unitary weight. The target regular equivalence is $\{\{1\}, \{2, 3\}, \{4, 5, 6, 7, 8\}\}$. In this context, the exact BDE method fails to recognize this partition because node 2 has two connections to the group $\{4, 5, 6, 7, 8\}$, while node 3 has three connections. In contrast, the ε-BDE approach successfully identifies the regular equivalence by setting ε to 1. This tolerance not only enables the recovery of exact regular equivalences, but also supports the identification of approximately regular equivalent nodes, accounting for the noise and uncertainty commonly present in real-world networks.

Iterative ε-BDE

As discussed in Sect. 1, an attractive feature of BDE, both in its exact and approximate variant, is that it can be computed by partition refinement (Cardelli et al. 2017a, 2018), where a given candidate initial partition of nodes is iteratively refined until the BDE criteria are satisfied. In ε-BDE, we obtain equivalence relations by closing under transitivity pairs of variables satisfying Definition 4. This transitive closure could lead to inconvenient associations. Indeed, if both pairs of nodes *i*, *j* and *j*, *k* are ε-BDE equivalent, their pairwise difference is less than ε; however, this does not mean that the difference between *i* and *k* is less than ε. To show this, let us consider the following example.

Example 1 Pick $\varepsilon = 0.3$ and consider the adjacency matrix

$$A = \begin{pmatrix} 0.4 & 0.1 & 0.5 & 0.7 \\ 0.1 & 0.5 & 0.6 & 0.7 \\ 0.5 & 0.6 & 0.3 & 0.8 \\ 0.7 & 0.7 & 0.8 & 0.3 \end{pmatrix}$$

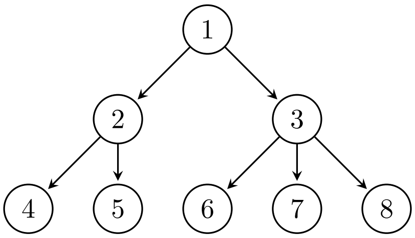

Fig. 3 Example of a network where BDE fails while ε-BDE find the regular equivalence $\{\{1\}, \{2, 3\}, \{4, 5, 6, 7, 8\}\}$ imposing ε equal to 1

Then, $3 \sim_{A,\mathcal{H},\varepsilon} 4$ for $\mathcal{H} = \{\{1,2,3,4\}\}$ because

$$|0.5 + 0.6 + 0.3 + 0.8 - (0.7 + 0.7 + 0.8 + 0.3)| = 0.3$$

Instead, $1 \not\sim_{A,\mathcal{H},\varepsilon} 3$ for $\mathcal{H} = \{\{1,2,3,4\}\}$ since

$$|0.4 + 0.1 + 0.5 + 0.7 - (0.5 + 0.6 + 0.3 + 0.8)| = 0.5$$

At the same time, $1 \sim^*_{A,\mathcal{H},\varepsilon} 3$ because $1 \sim_{A,\mathcal{H},\varepsilon} 2$ and $2 \sim_{A,\mathcal{H},\varepsilon} 3$. We infer that $\{1,\ldots,4\}/\sim^*_{A,\mathcal{H},\varepsilon} = \mathcal{H}$, showcasing that ε-BDE can aggregate too much.

To cope with this problem, we propose an iterative scheme where nodes are related by invoking the ε-BDE algorithm with increasingly larger values of ε, using an appropriate choice of initial partitions at each iteration. We call this approach *iterative ε-BDE* (Iε-BDE).

Let us discuss its pseudocode shown in Algorithm 1. It requires an adjacency matrix A, an initial partition $\mathcal{H}_{in}$ to keep track of the blocks discovered in the previous iterations, an initial tolerance ε_0, a step size δ, and a maximum tolerance Δ. At every iteration, the ε-BDE algorithm is invoked with the current tolerance, starting from ε_0. The algorithm refines the current partition with respect to A and A^T (lines 6-7) until no refinement is possible. The result is a partition satisfying Definition 4 on A and A^T according to the current ε. Afterward, the algorithm joins all trivial blocks of size one present in $\mathcal{H}_{\varepsilon'}$ (line 9). The intuition is to attempt node aggregation for smaller values of ε first. If that fails, i.e., nodes are eventually outputted as singleton blocks, the merging of such nodes is used to attempt aggregation for the larger ε values in the next iterations. The so-obtained partition $\mathcal{H}_\varepsilon$ is then used to refine the original input partition $\mathcal{H}_{in}$. Specifically, line 10 computes the coarsest partition that refines both $\mathcal{H}_{\varepsilon'}$ and $\mathcal{H}_{in}$, That is, a partition with a minimal number of blocks such that each of its blocks is a subset of a block in $\mathcal{H}_{\varepsilon'}$ and a block in $\mathcal{H}_{in}$. Thereafter, line 11 and 12 update and increase, respectively, $\mathcal{H}_{\varepsilon'}$ and ε. The algorithm then iterates until the user-defined maximum tolerance Δ is reached.

```
Require: Adjacency matrix $A$, initial partition of nodes $\mathcal{H}_{in}$, an initial tolerance $\varepsilon_0$,
    step size $\delta \geq 0$, maximum tolerance $\Delta \geq 0$.
1:  $\varepsilon \longleftarrow \varepsilon_0$
2:  $\mathcal{H}_{\varepsilon'} \longleftarrow \mathcal{H}_{in}$
3:  while $\varepsilon \leq \Delta$ do
4:      repeat
5:          $\mathcal{H}_\varepsilon \longleftarrow \mathcal{H}_{\varepsilon'}$
6:          $\mathcal{H}_\varepsilon \longleftarrow \{1,\ldots,n\}/\sim^*_{A,\mathcal{H}_\varepsilon,\varepsilon}$ via Algorithm 2
7:          $\mathcal{H}_{\varepsilon'} \longleftarrow \{1,\ldots,n\}/\sim^*_{A^T,\mathcal{H}_\varepsilon,\varepsilon}$ via Algorithm 2
8:      until $\mathcal{H}_\varepsilon = \mathcal{H}_{\varepsilon'}$
9:      $\mathcal{H}_{\varepsilon'} \longleftarrow$ joinSingletons($\mathcal{H}_{\varepsilon'}$)
10:     $\mathcal{H}_{in} \longleftarrow$ coarsestRefinement($\mathcal{H}_{in}$,$\mathcal{H}_{\varepsilon'}$)
11:     $\mathcal{H}_{\varepsilon'} \longleftarrow \mathcal{H}_{in}$
12:     $\varepsilon \longleftarrow \varepsilon + \delta$
13: end while
14: return $\mathcal{H}_\varepsilon$
```

Algorithm 1 Iterative ε-BDE.

```
**Require:** Adjacency matrix $A$, partition of nodes $\mathcal{H}$, tolerance $\varepsilon$
1:  $\mathcal{H}_\varepsilon \longleftarrow \mathcal{H}$
2:  **repeat**
3:      $\mathcal{H}' \longleftarrow \mathcal{H}_\varepsilon$
4:      $w[i, H] \longleftarrow 0$ for all $H \in \mathcal{H}'$, $1 \leq i \leq n$
5:      **for all** $H \in \mathcal{H}'$ **do**
6:          **for all** $i \in H$ **do**
7:              **for all** $j$ with $a_{i,j} \neq 0$ **do**
8:                  $w[j, H] \longleftarrow w[j, H] + a_{i,j}$
9:              **end for**
10:         **end for**
11:     **end for**
12:     $D_i \longleftarrow \emptyset$ for $1 \leq i \leq n$
13:     **for all** $1 \leq i \leq n$ **do**
14:         **for all** $1 \leq j \leq n$ **do**
15:             **if** $\sum_{H \in \mathcal{H}'} |w[i, H] - w[j, H]| \leq \varepsilon$ **then**
16:                 insert $j$ in $D_i$
17:             **end if**
18:         **end for**
19:     **end for**
20:     $\mathcal{H}_\varepsilon \longleftarrow$ **refine** $\mathcal{H}'$ using $D$ to compute $\sim^*_{\mathcal{H},A,\varepsilon}$
21: **until** $\mathcal{H}_\varepsilon = \mathcal{H}'$
22: **return** $\mathcal{H}'$
```

Algorithm 2 Routine for computing $\{1, \ldots, n\}/\sim^*_{\mathcal{H},A,\varepsilon}$

Complexity

The repeat until loop of Algorithm 1 requires at most $\mathcal{O}(n^5)$ steps. Additionally, the total number of steps performed by Algorithm 1 is at most $\mathcal{O}(\lceil \Delta/\delta \rceil n^5)$. We begin by noting that the while loop performs $\lceil \Delta/\delta \rceil$ iterations. Instead, each repeat until loop has at most *n* iterations because any partition may have at most *n* refinements. It thus suffices to show that Algorithm 2, inspired by the (non-approximative) forward equivalence algorithm (Valmari and Franceschinis 2010), runs in at most $\mathcal{O}(n^4)$. To see this, we first note that lines 5-11 of Algorithm 2, computing the number of connections toward nodes in the block *H*, can be computed in $\mathcal{O}(n^2)$. Then, Algorithm 2 computes in lines 13-19 the equivalence relation $\sim_{\mathcal{H},A,\varepsilon}$, where D_i is the list of nodes that are ε-BDE equivalent to node *i*. This portion of the code can be computed in $\mathcal{O}(n^3)$. This complexity arises due to the necessity of verifying the ε-BDE condition for every pair (*i*, *j*), requiring $\mathcal{O}(n^2)$ comparisons. The verification is performed in line 15 of Algorithm 2, involving the evaluation of differences. The cost of computing these differences is proportional to the number of blocks that, in the worst case, is $\mathcal{O}(n)$. In line 20 Algorithm 2 computes the blocks of the partition $\mathcal{H}_\varepsilon$ induced by the transitive closure $\sim^*_{\mathcal{H},A,\varepsilon}$. To do this, we consider the undirected graph induced by D_i, where node *i* is connected to node *j* if and only if *i* and *j* are ε-BDE equivalent. The blocks of the partition $\mathcal{H}_\varepsilon$ correspond to the strongly connected components of the graph and thus can be computed in $\mathcal{O}(n + m)$, where *m* is the number of edges. In the worst case, this implies that $\sim^*_{\mathcal{H},A,\varepsilon}$ can be computed in $\mathcal{O}(n^2)$. Finally, the main loop is repeated until no more refinement is possible. Since the partition is composed of *n* nodes, the number of possible refinements is bounded by *n*. To summarize, the overall computational complexity of Algorithm 2 is characterized by at most *n* iterations, thus giving rise to $\mathcal{O}(nn^3) = \mathcal{O}(n^4)$.

Remark 1 In all experiments from Sect. 4, repeat-until loop of Algorithm 1 executed at most twice.

Example 2 We next present Algorithm 1 on Example 1 with $\varepsilon_0 = 0$, $\Delta = 0.3$, $\delta = 0.1$ and $\mathcal{H}_{in} = \big\{\{1, 2, 3, 4\}\big\}$. For $\varepsilon = 0$, the repeat loop returns $\mathcal{H}_{\varepsilon'} = \big\{\{1\}, \{2\}, \{3\}, \{4\}\big\}$, making `joinSingletons return` $\mathcal{H}_{in}$ in line 9. The second iteration of the while loop with $\varepsilon = 0.1$ is therefore computed for the original $\mathcal{H}_{in}$. A similar statement can be made about $\varepsilon = 0.1$, meaning that the algorithm initiates its third while loop iteration with $\varepsilon = 0.2$ and the original $\mathcal{H}_{in}$. This time, nodes 1 and 2 are aggregated during the repeat loop, giving rise to $\mathcal{H}_{\varepsilon'} = \big\{\{1, 2\}, \{3\}, \{4\}\big\}$. At this point, it is worth noting that block $\{1, 2\}$ will not be split in any future iteration of the algorithm because the aggregation of nodes is monotonic in ε. The algorithm then executes `joinSingletons` in line 9 which yields $\mathcal{H}_{\varepsilon'} = \big\{\{1, 2\}, \{3, 4\}\big\}$. Since $\mathcal{H}_{\varepsilon'}$ is a refinement of $\mathcal{H}_{in}$, `coarsestRefinement` in line 10 does not change $\mathcal{H}_{\varepsilon'}$. After setting $\mathcal{H}_{in}$ to $\mathcal{H}_{\varepsilon'}$, the algorithm begins its final while loop with $\varepsilon = 0.3$ and $\mathcal{H}_{in} = \big\{\{1, 2\}, \{3, 4\}\big\}$. In it, the repeat loop aggregates nodes 3 and 4 and returns $\mathcal{H}_{\varepsilon} = \big\{\{1, 2\}, \{3, 4\}\big\}$, the final result of the algorithm.

The above example showcases why Algorithm 1 helps avoid unnecessary aggregations. Indeed, as discussed in Example 1, 0.3-BDE returns one block, while Algorithm 1 returns two blocks as outlined in Example 2. This is because aggregations arising for smaller ε-values are separated from these, which require larger ε-values. Intuitively, δ accounts for the granularity with which Algorithm 1 aggregates nodes. A large δ may result in over aggressive aggregations, while a small value may lead to prohibitively small blocks. In the experimental section, we evaluate Algorithm 1 on a number of benchmark networks from different fields.

Asymptotics for BA networks

In this section, we introduce a partition for a BA network (Barabási 2012) of size $\mathcal{O}(n)$ which is approximately BDE equivalent.

Definition 5 *(BA Model)* For a given size *n*, the BA model is described by the stochastic process $(G^t)_{0 \le t \le n}$, where G^t describes an undirected graph with nodes $\{1, \ldots, n\}$. Given G^{t-1}, we form G^t by adding node *t* and link node *t* to node *i*, where *i* is chosen randomly with

$$\mathbb{P}(i = j) = \begin{cases} \dfrac{d_{G^{t-1}}(j)}{2t - 1} & , \ 1 \le j \le t - 1 \\ \dfrac{1}{2t - 1} & , \ j = t \end{cases}$$

Here, $d_{G^{t-1}}(j)$ denotes the degree of node *j* in graph G^{t-1}.

Although not all real-world networks are strictly scale-free, we consider the BA model because it captures a key structural property observed in many such networks: the presence of highly connected nodes and a heavy-tailed degree distribution. This structure is commonly found in social and citation domains, as well as in other networks that exhibit a skewed degree distribution, even if they are not strictly scale-free. As we will show in the experimental section, the BA partition can serve as an effective pre-partitioning strategy for both Iε-BDE and CATREGE. We are now ready to introduce the

BA partition, which, inspired by the theory of the BA model, divides the nodes into two groups: *celebrities* and *followers*.

Definition 6 *(BA partition)* Fix a partition threshold $0 < \xi < 1$ and generate a sample run of the stochastic process $(G^t)_{0 \leq t \leq n}$. From this sample run, partition the set of graph nodes $\{1, \ldots, n\}$ into blocks of *celebrities* $\mathcal{C}$ and *followers* $\mathcal{F}$, that is, let

$$\mathcal{H} = \mathcal{C} \cup \mathcal{F} = \{C_1, \ldots, C_\kappa\} \cup \{F_1, \ldots, F_\kappa\},$$

where $\kappa = 1/\xi$ and[2]:

- celebrity nodes comprise $1, \ldots, \sqrt{n}$, that is, we have:

$$C_1 \cup \ldots \cup C_\kappa = \{1, \ldots, \sqrt{n}\};$$

- follower nodes constitute $\sqrt{n} + 1, \ldots, n$, meaning that

$$F_1 \cup \ldots \cup F_\kappa = \{\sqrt{n} + 1, \ldots, n\};$$

- $|C_\nu| = \xi\sqrt{n}$ and $C_\nu \leq C_{\nu'}$ elementwise for $\nu \leq \nu'$;
- $|F_\nu| = \xi(n - \sqrt{n})$ and $F_\nu \leq F_{\nu'}$ elementwise for $\nu \leq \nu'$.

For the BA process, computing the partition is straightforward because the node identifier corresponds to its age. In the case of an instance of a network with power-law distribution, without access to its underlying generative process, we can estimate the node's age by considering its degree and exploiting that older nodes have, on average, higher degrees than younger ones (Bollobás and Riordan 2004; Barabási 2012). For this reason, we can sort the nodes in decreasing order of their degrees and consider the first $\sqrt{N}$ as celebrities and the remaining $N - \sqrt{N}$ as followers. Then, these two groups of nodes will be split following the definition.

We next formally justify the choice of the partition from Definition 6. To this end, we recall the big-$\mathcal{O}$, big-Ω, and big-Θ notations.

Definition 7 For two functions $f, g : \mathbb{N}_0 \to \mathbb{R}_{\geq 0}$, we define

$$\begin{aligned} f = \mathcal{O}(g) &:\Longleftrightarrow \limsup_{n \to \infty} \tfrac{f(n)}{g(n)} < \infty \\ f = \Omega(g) &:\Longleftrightarrow \liminf_{n \to \infty} \tfrac{f(n)}{g(n)} > 0 \\ f = \Theta(g) &:\Longleftrightarrow f = \mathcal{O}(g) \text{ and } f = \Omega(g) \end{aligned}$$

The following key auxiliary result estimates the probability of finding links between nodes of a BA model and sharpens (Bollobás and Riordan 2004) by studying the error terms in the proof of (Bollobás and Riordan 2004, Lemma 2).

Lemma 1 *In a BA model $(G^t)_{0 \leq t \leq n}$, let g_j with $j \leq n$ denote the node to which node j connects to. Then for all $i < j < k$, we have*

---

[2] To simplify presentation, we assume that n is a square and that $\xi\sqrt{n}, 1/\xi \in \mathbb{N}$. The assumption can be dropped by rounding up.

$$\mathbb{P}(g_j = i) = \frac{1}{2}\frac{1}{\sqrt{ij}} + \mathcal{O}\Big(\frac{1}{ij}\Big)$$
$$\mathbb{P}(g_j = i, g_k = i) = \frac{1}{2}\frac{1}{i\sqrt{jk}} + \mathcal{O}\Big(\frac{1}{i\sqrt{ijk}}\Big)$$

Moreover, it holds that

$$\mathbb{P}(g_j = i, g_k = i) - \mathbb{P}(g_j = i)\mathbb{P}(g_k = i) = \frac{1}{4i\sqrt{jk}} + \mathcal{O}(\frac{1}{i\sqrt{ijk}})$$

For a block H of some given partition of nodes $\mathcal{H}$, we shall next study the number of H-in-degree of a node i. Formally, this is captured by the random variable $\sum_{j\in H} X^{i,j}$, where

$$X^{i,j} = \begin{cases} 1, & g_j = i \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Revisiting Lemma 1, we note that the in-degree of a node can be approximated by a sequence of Bernoulli trials. However, while the first identity of Lemma 1 suggests to approximate $X^{i,j}$ by a Bernoulli variable with success probability $1/2\sqrt{ij}$, the third statement shows that variables $X^{i,j}$ and $X^{i,k}$ are positively correlated, hence stochastically dependent. This prevents direct applications of the law of large numbers or the central limit theorem.

The next result studies the number of links between the blocks of $\mathcal{H}$.

Theorem 2 *Let $(G^t)_{0\le t\le n}$ be a BA model and $\mathcal{H}$ as in Definition 6. Then, the number of links between celebrities and followers is of order $\sqrt{\xi}\sqrt[4]{n}$, that is*

$$\mathbb{E}[\deg(i,H)] = \Theta(\sqrt{\xi}\sqrt[4]{n}), \quad H \in \mathcal{F},\ i = \gamma\sqrt{n} \in C_\nu,\ C_\nu \in \mathcal{C}$$

where $\deg(i,H)$ is the number of edges from *i* to nodes in block $H \in \mathcal{F}$ and $0 < \gamma < 1$; in all other cases, the number of links is $\mathcal{O}(1)$, i.e., negligible because it does not grow with *n*.

After studying the connectivity between the blocks of $\mathcal{H}$, we are ready to state our main result.

Theorem 3 *Let $G = (G^t)_{0\le t\le n}$ be a BA model and $\mathcal{H}$ as in Definition 6. Then, for large n, partition $\mathcal{H}$ is on average a $\sqrt{\xi}$-BDE of the scaled network $G/\sqrt[4]{n}$. Specifically, for any $0 < \gamma, \gamma' < 1$ and $H \in \mathcal{H}$, for large $n \ge 1$, we have*

$$\frac{1}{\sqrt[4]{n}}|\mathbb{E}[\deg(i,H)] - \mathbb{E}[\deg(i',H)]| = \mathcal{O}(\sqrt{\xi}|\gamma' - \gamma|),$$

where

- either $i = \gamma\sqrt{n}$ and $i' = \gamma'\sqrt{n}$ such that $i, i' \in \bar{H} \in \mathcal{C}$;
- or $i = \gamma n$ and $i' = \gamma' n$ such that $i, i' \in \bar{H} \in \mathcal{F}$.

A strengthening of the above result which would ensure that the difference of degrees (rather than their expected values) is with high probability of order $\mathcal{O}(|\gamma - \gamma'|)$ is difficult. Indeed, as stated in the next result, the variance of difference $(\deg(i,H) - \deg(i',H))/\sqrt[4]{n}$ does not vanish as *n* increases.

Theorem 4 *In a BA model* $(G^t)_{0\le t\le n}$. *Then, for any* $H \in \mathcal{H}$ *and* $i \neq i'$, *it holds that:*

$$\mathbb{V}[\deg(i,H) - \deg(i',H)] \geq \mathbb{V}[\deg(i,H)] + \mathbb{V}[\deg(i',H)]$$

and

$$\mathbb{V}[\deg(i,H)] = \begin{cases} \mathcal{O}(1), & i \in C_\nu,\ H \in \mathcal{C} \\ \Theta(\sqrt{n}), & i \in C_\nu,\ H \in \mathcal{F} \\ \mathcal{O}(1), & i \in F_\nu,\ H \in \mathcal{C} \\ \mathcal{O}(1), & i \in F_\nu,\ H \in \mathcal{F} \end{cases}$$

We proved that the BA partition is asymptotically an ε-BDE partition for binary networks with power-law distribution of the nodes. By identifying important (celebrity) and less important nodes (followers), practitioners can use this result as a starting point in the analysis of the network. The BA partition can be used to avoid an aggressive reduction of the network, a well-known issue for regular equivalence on binary networks (Borgatti and Everett 1989). In the experimental section, we show how Iε-BDE and CATREGE can employ the BA partition as a pre-partitioning strategy to improve the results on binary networks.

Experiments

We apply our framework to a number of benchmark networks, both weighted and binary, to show its effectiveness and compare it against the state of the art. As indirect approaches we consider REGE and CATREGE; for direct approaches, we consider different variants of blockmodeling. To make a fair comparison, we analyzed the algorithms by keeping the same level of granularity for all. This is controlled by incomparable parameters δ/Δ for Iε-BDE and the number of clusters for the competing techniques. Thus, we first fixed Iε-BDE parameters in a uniform manner across all networks, as detailed next; then we chose the number of clusters of direct and indirect approaches equal to the number of partition blocks returned by Iε-BDE.

Experimental set-up

We consider benchmark networks of different sizes as listed in Table 1. The networks are divided into binary and weighted networks. For binary networks, we specify whether they are directed (D) or undirected (U), since this affects how regular equivalence is computed, as discussed below.

Iterativeε–BDE

We implemented the Iε-BDE in a prototype that uses the implementation of ε-BDE already available in the software tool ERODE (Cardelli et al. 2017b, 2025). The two parameters for Iε-BDE are the maximum tolerance Δ and the step size δ. For an unbiased, model-independent choice of these parameters, we considered the following heuristic: since ε-BDE relates nodes with similar row-sums of the adjacency matrix A, we set up Δ by picking a value roughly equal to the average sum of the rows of A. For δ, instead, we took a value one order of magnitude smaller than the average value of the non-zero entries of A, considering only the weights greater than 0. When this value is less than the minimum non-zero entry in A, we set up δ equal to this minimum. The values of δ and Δ are listed in Table 1. With this choice, Iε-BDE was run using $\varepsilon_0 = 0$ and, unless

Table 1 Parameters and results for iterative ε-BE

Weighted networks						**Iε-*BDE***		
Model	**References**	**n**	**REGE**	**0-BDE**	**Size**	**δ**	**Δ**	**Ratio**
EIES	Freeman and Freeman (1979)	32	32	32	19	10^0	300	0.59
Windsurfers	Kunegis (2013)	43	43	43	20	10^0	60	0.47
Ecosystem	Kunegis (2013)	128	128	128	37	10^{-1}	30	0.29
FaoTrade	De Domenico et al. (2015)	214	214	214	170	10^2	$2 \cdot 10^4$	0.79
WTN	Gaulier and Zignago (2010)	226	226	226	171	10^4	10^7	0.75
CElegans	Watts and Strogatz (1998)	306	286	286	207	10^0	40	0.67
USairport	Colizza et al. (2007)	500	500	500	223	10^4	10^6	0.45
FB	Opsahl and Panzarasa (2009)	1899	–	1857	1310	10	1000	0.69
						Average ratio:		0.59

Binary networks						**Iε-*BDE***				**Iε-BDE / BA**	
Model	**References**	**Type**	**n**	**CATREGE**	**0-BDE**	**Size**	**δ**	**Δ**	**Ratio**	**Size**	**Ratio**
Karate	Kunegis (2013)	U	34	30	30	16	10^0	4	0.47	27	0.79
GD	Batagelj and Mrvar (2006)	U	73	54	54	17	10^0	2	0.23	34	0.47
Revolution	Kunegis (2013)	U	136	56	56	11	10^0	2	0.11	46	0.34
Email	Kunegis (2013)	D	167	166	166	14	10^0	30	0.08	45	0.27
Physician	Kunegis (2013)	D	241	241	241	3	10^0	5	0.01	195	0.80
FilmTrust	Kunegis (2013)	D	874	–	673	238	10^0	2	0.27	371	0.42
BlogCatalog	Zafarani and Liu (2009)	U	10312	–	10106	5455	10^0	60	0.53	5490	0.53
Youtube	Zafarani and Liu (2009)	U	15088	–	12691	4760	10^0	10	0.32	6766	0.45
						Average ratios:			0.25		0.51

otherwise stated, with the singleton initial partition considering all nodes. We use this output to compare against the other approaches.

REGE

For weighted networks, we compare against the indirect method of REGE as implemented in the R package presented in Matjasiĉ et al. (2020). Following (Žiberna 2009), we set up a number of iterations for every model equal to 100. From REGE's similarity matrix, we retrieved the partition with the same number of blocks as Iε-BDE's partition using the dendrogram associated to hierarchical clustering, which was computed with the scikit-learn library (Pedregosa et al. 2011). In this case, following the literature on this subject (Ziberna 2008), we consider two linkages for hierarchical clustering, i.e., single and complete links (Ziberna 2008) (denoted by *REGE+SL* and *REGE+CL* in the forthcoming tables, respectively).

CATREGE

For binary networks, we considered the indirect method of CATREGE, since it achieves superior performance with respect to REGE (Borgatti and Everett 1993). We used the implementation in UCINET (Borgatti et al. 2002). CATREGE allows specifying an initial partition. In the first iteration, it divides the nodes following the given partition instead of considering all of them in the same block. The nodes in different blocks cannot be associated in the following iterations. To avoid CATREGE identifying all nodes as equivalent (Borgatti and Everett 1993), as discussed, we employed the BA partition in all binary networks. As with REGE, hierarchical clustering was applied to the resulting similarity matrix.

Blockmodeling

For direct approaches, we compare against the binary, valued, and homogeneity blockmodeling approaches (Žiberna 2007), using the R package by Matjasiĉ et al. (2020). Since these techniques are based on a local optimization algorithm, we set 1000 repetitions/different starting partitions to check. Valued and homogeneity blockmodeling look for f-regular equivalence, where the function f was set to max, which is the common setting for regular equivalence (Matjasiĉ et al. 2020). Valued blockmodeling requires specifying a parameter m that distinguishes between prominent and non-prominent weights. The best way to determine m is to have prior knowledge about the network. In absence of this, it is possible to choose the value of m considering the distribution of the links. Following Matjasiĉ et al. (2020) and Žiberna (2007), we set m equal to the median and the mode of the nonzero weights.

Error metric

To compare the precision of the different methods in computing approximately equivalent nodes, we define a notion of similarity based on PageRank, following Tu et al. (2018). Specifically, for each approximately regularly equivalent block, we compute the maximum PageRank difference across all pairs of its nodes. Then, we compute the minimum, the average, and the maximum value across all blocks. Thus, lower values of such indices correspond to more similar (in terms of regular equivalence) blocks.

Timeout

Throughout all experiments we set a 3 h timeout for each analysis.

Results

Preliminary analysis

We computed the number of regularly equivalent node blocks using REGE and CATREGE. These blocks contain the nodes that strictly satisfy the regular equivalence definition. We show that the occurrence of regular equivalent nodes is rare, especially for the weighted networks. Table 1 shows the numbers of regularly equivalent nodes. In all weighted networks but CElegans, no nontrivial blocks were found, proving that no regular equivalent nodes can be found. For CElegans, REGE returned 286 blocks. CATREGE has an appreciable number of equivalent nodes; larger networks could not be analyzed because CATREGE supports networks with at most 256 nodes. For comparison, we also analyzed the regular equivalences that can be found using Iε-BDE by setting $\Delta = 0$ (shortened 0-BDE in the table column); this corresponds to computing the condition according to Theorem 1. Interestingly, although Theorem 1 is only a sufficient condition for regular equivalence, the analyzed networks do not distinguish the two notions since the algorithms return the same number of blocks.

Weighted networks

Table 1 shows the number of approximately equivalent blocks of nodes identified by Iε-BDE with the chosen parameters, highlighting that it halves, on average, the network size. The comparison against REGE and blockmodeling for weighted networks is reported in Table 2 (left). Iε-BDE proved generally more accurate and can yield errors up to one order of magnitude smaller. We also observe that homogeneity blockmodeling

Table 2 Comparison on weighted (left) and binary (right) networks

Weighted networks				
Method	**Errors**			**Times (s)**
	Min	**Avg**	**Max**	
EIES				
Iε-BDE	**2.23E−4**	**3.31E−3**	**9.34E−3**	2.62
REGE+CL	1.05E−3	8.56E−3	3.15E−2	**0.23**
REGE+SL	1.50E−3	1.26E−2	3.72E−2	**0.23**
Blockmod. Hom.	1.50E−3	5.30E−3	8.27E−3	319.86
Blockmod. Val. median	3.15E−2	4.65E−2	6.15E−2	142.74
Blockmod. Val. mode	7.12E−4	2.06E−2	8.30E−2	115.23
Windsurfers				
Iε-BDE	**5.66E−6**	**4.46E−3**	1.50E−2	1.47
REGE+CL	1.31E−4	6.36E−3	3.85E−2	**0.46**
REGE+SL	2.10E−4	7.41E−3	3.85E−2	**0.46**
Blockmod. Hom.	1.93E−3	5.67E−3	**1.41E−2**	532.80
Blockmod. Val. median	1.99E−3	3.78E−2	7.83E−2	291.55
Blockmod. Val. mode	1.31E−3	1.86E−2	3.85E−2	300.61
Ecosystem				
Iε-BDE	**6.89E−8**	**7.61E−4**	**2.89E−3**	**18.13**
REGE+CL	6.89E−8	1.27E−2	2.18E−1	18.20
REGE+SL	6.89E−8	1.70E−3	8.16E−3	18.20
FaoTrade				
Iε-BDE	**0.0**	**3.47E−4**	**1.38E−3**	**23.53**
REGE+CL	0.0	2.68E−3	2.89E−2	503.58
REGE+SL	0.0	5.23E−4	2.89E−2	503.58
WTN				
Iε-BDE	**2.42E−7**	**7.55E−5**	**4.35E−4**	**154.95**
REGE+CL	1.53E−5	1.90E−3	1.33E−2	857.40
REGE+SL	1.15E−5	4.93E−3	3.08E−2	857.40
CElegans				
Iε-BDE	**0.0**	**5.61E−4**	3.46E−3	**3.54**
REGE+CL	0.0	6.47E−4	**2.06E−3**	75.30
REGE+SL	0.0	7.56E−4	3.40E−3	75.30
USairport				
Iε-BDE	**1.25E−7**	**1.92E−4**	**3.45E−3**	**13.73**
REGE+CL	1.32E−6	1.30E−3	3.27E−2	408.60
REGE+SL	1.52E−7	1.64E−3	4.02E−2	408.60
FB				
Iε-BDE	**0.0**	**3.21E−5**	**5.25E−4**	**47.46**
Binary networks				
Method	**Errors**			**Times (s)**
	Min	**Avg**	**Max**	
Karate				
BA partition	2.37E−5	2.90E−3	7.62E−3	–
Iε-BDE	0.0	8.85E−4	5.48E−3	**0.28**
CATREGE+CL	**0.0**	**6.31E−4**	**2.41E−3**	1.00
CATREGE+SL	0.0	1.23E−3	2.61E−3	1.00
Blockmodeling	0.0	9.99E−4	4.99E−3	84.6
GD				
BA partition	0.0	3.09E−3	6.06E−3	–
Iε-BDE	**0.0**	**1.04E−3**	**4.89E−3**	**0.12**
CATREGE+CL	0.0	2.34E−3	6.06E−3	1.00
CATREGE+SL	0.0	2.12E−3	6.06E−3	1.00

Table 2 (continued)

Binary networks				
Method	**Errors**			**Times (s)**
	Min	**Avg**	**Max**	
Blockmodeling	4.54E−3	2.18E−2	9.92E−2	1441.8
Revolution				
BA partition	1.63E−4	1.38E−3	1.01E−2	–
Iε-BDE	**0.0**	**1.02E−4**	2.63E−3	**0.32**
CATREGE+CL	0.0	5.71E−4	1.01E−2	1.00
CATREGE+SL	0.0	5.57E−4	**1.01E−3**	1.00
Email				
BA partition	0.0	1.23E−3	2.96E−3	–
Iε-BDE	**0.0**	**6.71E−4**	**2.96E−3**	2.11
CATREGE+CL	0.0	9.99E−4	2.96E−3	**1.00**
CATREGE+SL	0.0	9.94E−4	2.96E−3	**1.00**
Physician				
BA partition	0.0	3.95E−3	1.38E−2	–
Iε-BE	**0.0**	**4.92E−4**	**3.28E−3**	**0.77**
CATREGE+CL	0.0	3.12E−3	1.38E−2	1.00
CATREGE+SL	0.0	3.32E−3	6.93E−3	1.00
FilmTrust				
BA partition	0.0	3.25E−3	1.19E−2	–
Iε-BDE	**0.0**	**1.79E−4**	**3.41E−3**	**1.46**
BlogCatalog				
BA partition	1.33E−5	4.47E−4	2.29E−3	–
Iε-BDE	**0.0**	**2.86E−6**	**2.50E−5**	**1250.0**
Youtube				
BA partition	5.83E−5	5.05E−4	1.83E−3	–
Iε-BDE	**0.0**	**5.77E−6**	**1.21E−4**	**114.04**

Best results in bold; methods that timed out are not listed

performs better than REGE and valued blockmodeling. Blockmodeling requires a considerable amount of time, making it applicable in these examples to networks up to 43 nodes within the given threshold. REGE's implementation is faster than Iε-BDE for analyzed networks with fewer than 306 nodes. We remark that FaoTrade and WTN, despite being similar in size, are characterized by considerably different runtimes. This is attributed to the density of the network. For larger networks, Iε-BDE proved faster, justifying the differences in asymptotic cost complexity of the two algorithms; in practice, REGE could not analyze the FB network within the timeout.

Binary networks

With the settings of Table 1, Iε-BDE identifies nodes more aggressively, on average, in binary networks than in weighted networks (ratios 0.25 and 0.59, respectively). We now consider Iε-BDE initialized with the BA partition, following Definition 6. Here we set $\xi = 0.1$ for all networks, always leading to 20 blocks. With this setting, and using the same values of δ and Δ, the average ratio using the BA partition becomes comparable to that of weighted networks (last two columns of Table 1).

Table 2 (right) shows the comparison with CATREGE and (binary) blockmodeling. These results were obtained when initializing Iε-BDE with the BA partition for a fair analysis against CATREGE. For reference, we also report the error statistics directly computed on the BA partition. Since, in all cases, both the CATREGE and the Iε-BDE results refine the BA partition, their error metrics are consistently improved. However, we remark that, for small networks, the BA partition already provides errors within the same order of magnitude as the iterative algorithms; for larger networks, the error statistics of the BA pre-partition obviously deteriorate because, by fixing the number of blocks, it clusters increasingly more nodes.

Overall, in the networks where the comparison is possible, Iε-BE yields superior precision than CATREGE and blockmodeling except for the Karate network where CATREGE+CL achieves better results. CATREGE may be faster than Iε-BDE in some cases, but it does not support networks larger than 256 nodes, as discussed. The blockmodeling results confirm the scalability issues observed in weighted networks, timing out already with 136 nodes.

Discussion

Our method is designed to produce interpretable node partitions of approximately regular equivalent nodes. While many role extraction techniques, such as REGE and CATREGE, rely on structural similarity matrices followed by clustering, our approach directly computes a discrete partition satisfying the ε-BDE condition. On the other side, blockmodeling computes the partition by swapping nodes among the blocks, but does not provide guarantees on the optimality of the resulting partition.

Across the diverse set of networks we tested, our method consistently outperforms REGE, CATREGE, and blockmodeling both in accuracy and computational efficiency. Our approach does not depend on strong assumptions about the network structure and performs well even in complex, noisy real networks. Furthermore, the scalability demonstrated in our experiments enables Iε-BDE to handle networks of sizes that the other methods cannot process efficiently.

A current limitation of our approach lies in the difficulty of estimating the exact number of blocks in the final partition. This number depends on the tolerance parameter ε and on the topology of the input network. However, at the current stage of development, it is not possible to predict the resulting partition size a priori for a given ε. While the resulting number of blocks is theoretically explainable within the framework of our method, practitioners who need to explicitly control the number of roles or partitions may prefer REGE, CATREGE, or blockmodeling, which allow finer control over the output dimensionality.

Conclusion

We have presented Iε-BDE, a new method to compute approximate regular equivalences for networks based on a partition refinement algorithm. In most examples, it showed superior precision and performance with respect to the state of the art, enabling the analysis of networks that are beyond the scope of applicability of currently available direct and indirect approaches. The asymptotic result for binary networks that present skewed distributions provides a pre-partitioning heuristic for practical models that avoid the problem of aggressive clustering of nodes. A possible extension would be to develop a similar result for other classes of distributions.

Appendix A: Proofs

Proof of Theorem 1 Follows via the if-then direction of (Chen et al. 2012, Lemma 1), applied on each A^l. $\square$

For the proof of Lemma 1, we will need the following.

Lemma 2 For any $0 < s < t$, it holds that

$$\prod_{i=s}^{t}\left(1+\frac{1}{2i-1}\right)=\sqrt{\frac{t}{s}}+\mathcal{O}(s^{-1})$$

Proof As suggested in (Bollobás and Riordan 2004, Lemma 2), we approximate the product by applying the logarithm

$$\log\left(\prod_{i=s}^{t}\left(1+\frac{1}{2i-1}\right)\right)=\sum_{i=s}^{t}\log\left(1+\frac{1}{2i-1}\right)=\sum_{i=s}^{t}\log\left(\frac{2i}{2i-1}\right)$$

and noting that the integral convergence test ensures

$$\left|\sum_{i=s}^{t}\log\left(\frac{2i}{2i-1}\right)-\int_{s}^{t}\log\left(\frac{2x}{2x-1}\right)dx\right|\leq\log\left(\frac{2s}{2s-1}\right)$$

Moreover, $\log(2x/(2x-1))=\log(2x)-\log(2x-1)$, while integration by substitution yields

$$\int_{s}^{t}\log(2x)dx=\tfrac{1}{2}\int_{2s}^{2t}\log(x)dx$$
$$\int_{s}^{t}\log(2x-1)dx=\tfrac{1}{2}\int_{2s-1}^{2t-1}\log(x)dx$$

Hence, by Taylor's theorem, there exist $\xi,\xi'\in(-1;0)$ with

$$\begin{aligned}\int_{s}^{t}2\log\left(\frac{2x}{2x-1}\right)dx&=\left[\int_{2t-1}^{2t}\log(x)dx-\int_{2s-1}^{2s}\log(x)dx\right]\\&=\log(2t)+\frac{1}{2t}\cdot\xi'-\log(2s)-\frac{1}{2s}\xi'\\&=\log(t/s)+\mathcal{O}(s^{-1})\end{aligned}$$

Recalling that $\frac{1}{2}\log(a)=\log(\sqrt{a})$, we thus obtain

$$\begin{aligned}\Big|\prod_{i=s}^{t}\left(1+\frac{1}{2i-1}\right)-\sqrt{\frac{t}{s}}\Big|&\leq\frac{2s}{2s-1}\cdot\exp(\mathcal{O}(s^{-1}))\\&\leq\big(1+\mathcal{O}(1/s))(1+\mathcal{O}(1/s)),\end{aligned}$$

yielding the claim. $\square$

We next prove Lemma 1.

Proof of Lemma 1 In the proof of (Bollobás and Riordan 2004, Lemma 2), the authors show that

$$\mathbb{P}(g_j = i) = \frac{1}{2j-1} \prod_{k=i}^{j-1} \left(1 + \frac{1}{2k-1}\right),$$

With this, Lemma 2 implies

$$\mathbb{P}(g_j = i) = \frac{1}{2j-1} \left(\frac{(j-1)^{1/2}}{i^{1/2}} + \mathcal{O}(i^{-1}) \right) = \left(\frac{1}{2j} + \mathcal{O}(j^{-2}) \right) \left(\frac{j^{1/2}}{i^{1/2}} + \mathcal{O}(i^{-1}) \right)$$

because $|\sqrt{j} - \sqrt{j-1}| \leq |\frac{1}{2}(j-1)^{-1/2}| \cdot 1 \leq i^{-1/2}$ by the mean value theorem. For the second and third statement, we note using the notation from (Bollobás and Riordan 2004, Lemma 2) that

$$\mathbb{E}(d_{k,i} I_{g_j=i} \mid G^{k-1}) = d_{k-1} I_{g_j=i} + \frac{d_{k-1}}{2k-1} I_{g_j=i} = \left(1 + \frac{1}{2k-1}\right) d_{k-1,i} I_{g_j=i},$$

which in turn implies

$$\mathbb{E}(d_{k,i} I_{g_j=i}) = \prod_{s=j}^{k} \left(1 + \frac{1}{2s-1}\right) \mathbb{E}(d_{j,i} I_{g_j=i})$$

as postulated in the proof of (Bollobás and Riordan 2004, Lemma 2). With this, the discussion from the aforementioned lemma ensures that

$$\begin{aligned} \mathbb{P}(g_j = i, g_k = i) &= \frac{1}{2k-1} \prod_{s=j}^{k-1} \left(1 + \frac{1}{2s-1}\right) \frac{1}{2j-1} \frac{2j-1}{2i+1} \frac{4i+2}{2i-1} \\ &= \frac{1}{2k-1} \frac{1}{2i-1} \prod_{s=j}^{k-1} \left(1 + \frac{1}{2s-1}\right) \cdot 2, \end{aligned}$$

where, thanks to Chamberland and Straub (2013) and $\Gamma(x+1) = x\Gamma(x)$ for any $x > 0$, we have that

$$\begin{aligned} \frac{2j-1}{2i+1} \frac{4i+2}{2i-1} = \mu^{(2)}_{j-1,i} &= \prod_{s=i+1}^{j-1} \left(1 + \frac{1}{2s-1}\right) \mu^{(2)}_{i,i} = \frac{\Gamma(i+\frac{1}{2})}{\Gamma(j-\frac{1}{2})} \frac{\Gamma(j+\frac{1}{2})}{\Gamma(i+\frac{3}{2})} \mu^{(2)}_{i,i} \\ &= \frac{j-\frac{1}{2}}{i+\frac{1}{2}} \mu^{(2)}_{i,i} = \frac{2j-1}{2i+1} \mu^{(2)}_{i,i} \end{aligned}$$

and

$$\mu^{(2)}_{i,i} = \frac{2i-2}{2i-1} \cdot (1^2 + 1) + \frac{1}{2i-1} \cdot (2^2 + 2) = \frac{4i+2}{2i-1}$$

This, in turn, allows us to conclude that

$$
\begin{aligned}
\mathbb{P}(g_j = i, g_k = i) - \mathbb{P}(g_j = i)\mathbb{P}(g_k = i) = \\
&= \frac{1}{2k-1}\frac{1}{2i-1}\prod_{s=j}^{k-1}\left(1+\frac{1}{2s-1}\right)\cdot 2 - \frac{1}{2k-1}\prod_{s=i}^{k-1}\left(1+\frac{1}{2s-1}\right) \\
&\quad \cdot \frac{1}{2j-1}\prod_{s=i}^{j-1}\left(1+\frac{1}{2s-1}\right) \\
&= \frac{1}{2k-1}\prod_{s=j}^{k-1}\left(1+\frac{1}{2s-1}\right)\left[\frac{2}{2i-1} - \frac{1}{2j-1}\prod_{s=i}^{j-1}\left(1+\frac{1}{2s-1}\right)^2\right] \\
&= \frac{1}{4i\sqrt{jk}} + \mathcal{O}\left(\frac{1}{i\sqrt{ijk}}\right),
\end{aligned}
$$

where the last identity follows by invoking several times Lemma 2. This establishes the third statement which, together with the first statement, implies the second statement. ☐

Proof of Theorem 2 Pick $H \in \mathcal{H}$ with $H = \{a, a+1, \ldots, b-1, b\}$, an $i \in \{\gamma\sqrt{n}, \gamma n\}$ such that $i \leq a$ and let $X^{i,j}$ be as in (1). Then

$$
\begin{aligned}
\mathbb{E}\left[\sum_{j\in H} X^{i,j}\right] &= \sum_{j\in H}\mathbb{E}[X^{i,j}] \\
&= \sum_{j\in H}\frac{1}{2\sqrt{ij}} + \sum_{j\in H}\mathcal{O}(1/ij) \\
&= \sum_{j\in H}\frac{1}{2\sqrt{ij}} + \mathcal{O}((\log(b)+1)/i) \\
&= \int_a^b \frac{1}{2\sqrt{ij}}dj + \mathcal{O}(\log(b)/i + 1/i + 1/\sqrt{ia}) \\
&= \sqrt{b/i} - \sqrt{a/i} + \mathcal{O}(\log(b)/i + 1/i + 1/\sqrt{ia}),
\end{aligned}
$$

where the first identity is due to the linearity of the expected value, the second due to the first identity of Lemma 1 and the logarithmic growth of harmonic numbers, the third due to the integral convergence test, and the fourth due follow via integration. This implies

$$
\mathbb{E}[\sum_{j\in H} X^{i,j}] = \begin{cases}
\mathcal{O}(1), & i \in C_\nu, H \in \mathcal{C} \\
\Theta(\sqrt{\xi}\sqrt[4]{n}), & i \in C_\nu, H \in \mathcal{F} \\
\mathcal{O}(1), & i \in F_\nu,\ H \in \mathcal{C} \\
\mathcal{O}(1), & i \in F_\nu,\ H \in \mathcal{F}
\end{cases}
$$

The statement follows because $|\deg(i, H) - \sum_{j\in H} X^{i,j}| \leq 1$. ☐

Proof of Theorem 3 From the proof of Theorem 2, we infer for $i < i'$:

$$
\mathbb{E}\left[\sum_{j\in H} X^{i,j}\right] - \mathbb{E}\left[\sum_{j\in H} X^{i',j}\right] = (\sqrt{b} - \sqrt{a})\left(\frac{1}{\sqrt{i}} - \frac{1}{\sqrt{i'}}\right) + \mathcal{O}(\log(b)/i + 1/i + 1/\sqrt{ia})
$$

We consider the case $i, i' \in C_\nu$ and $H \in \mathcal{F}$ as the other cases are straightforward. By Taylor's theorem, we infer for $0 < x_0 < x_1$:

$$|x_0^{-1/2} - x_1^{-1/2}| \leq |\tfrac{1}{2}x_0^{-3/2}| \cdot |x_1 - x_0|$$

Setting $x_1 = \gamma_1 \sqrt[2]{n}$ and $x_0 = \gamma_0 \sqrt[2]{n}$, the above formula yields

$$\left| \frac{1}{\sqrt{x_0}} - \frac{1}{\sqrt{x_1}} \right| \leq |\gamma_1 - \gamma_0| / \sqrt[4]{n}\gamma_0^{3/2}$$

Noting that $\sqrt{b} - \sqrt{a} = \mathcal{O}(\sqrt{\xi n})$, we obtain

$$\left| (\sqrt{b} - \sqrt{a}) \left(\frac{1}{\sqrt{i}} - \frac{1}{\sqrt{i'}} \right) \right| \leq \sqrt{\xi}\sqrt[4]{n}\frac{|\gamma_1 - \gamma_0|}{\gamma_0^{3/2}}$$

The statement follows because $|\deg(i, H) - \sum_{j \in H} X^{i,j}| \leq 1$. ☐

Proof of Theorem 4 Letting j, j' ranging over H, we first note that

$$\begin{aligned}
&\mathbb{V}\left[\sum_j X^{i,j} - \sum_j X^{i',j} \right] \\
&\quad = \mathbb{V}\left[\sum_j X^{i,j} \right] + \mathbb{V}\left[\sum_j X^{i',j} \right] - 2Cov\left[\sum_j X^{i,j}, \sum_j X^{i',j} \right] \\
&\quad = \mathbb{V}\left[\sum_j X^{i,j} \right] + \mathbb{V}\left[\sum_j X^{i',j}] - 2 \sum_j \sum_{j'} Cov[X^{i,j}, X^{i',j'} \right]
\end{aligned}$$

The first statement follows by noting that (Bollobás and Riordan 2004, Lemma 3) ensures for any $i \neq i'$ and $j, j' \in H$ that

$$Cov[X^{i,j}, X^{i',j'}] = \mathbb{E}[X^{i,j}X^{i',j'}] - \mathbb{E}[X^{i,j}]\mathbb{E}[X^{i',j'}] \leq 0.$$

To see the second statement, we note that Lemma 1 implies for $j, j' \in H$ with $j \neq j'$:

$$\begin{aligned}
Cov(X^{i,j}, X^{i,j'}) &= \mathbb{E}[X^{i,j}X^{i,j'}] - \mathbb{E}[X^{i,j}]\mathbb{E}[X^{i,j'}] \\
&= \mathbb{P}(g_j = i, g_k = i) - \mathbb{P}(g_j = i)\mathbb{P}(g_k = i) \\
&= \frac{1}{4i\sqrt{jk}} + \mathcal{O}\left(\frac{1}{i\sqrt{ijk}} \right)
\end{aligned}$$

With this, we obtain

$$\begin{aligned}
\mathcal{E} &:= \sum_{j \in H} \sum_{j' \in H \setminus \{j\}} \left(\mathbb{E}[X^{i,j}X^{i,j'}] - \mathbb{E}[X^{i,j}]\mathbb{E}[X^{i,j'}] \right) \\
&= \sum_{j \in H} \sum_{j' \in H \setminus \{j\}} \left(\frac{1}{4i\sqrt{jj'}} + \mathcal{O}\left(\frac{1}{i\sqrt{ijj'}} \right) \right) \\
&= \begin{cases} \mathcal{O}(1), & i \in C_\nu,\ H \in \mathcal{C} \\ \Theta(\sqrt{n}), & i \in C_\nu,\ H \in \mathcal{F} \\ \mathcal{O}(1), & i \in F_\nu,\ H \in \mathcal{C} \\ \mathcal{O}(1), & i \in F_\nu,\ H \in \mathcal{F} \end{cases}
\end{aligned}$$

Moreover, one can observe

$$
\begin{aligned}
\mathbb{V}\left(n^{-1/4}\sum_{j\in H}X^{i,j}\right) &= n^{-1/2}\left(\sum_{j\in H}\mathbb{V}[X^{i,j}]+\mathcal{E}\right)\\
&= n^{-1/2}\sum_{j\in H}\mathbb{V}[X^{i,j}]+n^{-1/2}\mathcal{E}\\
&\le n^{-1/2}\sum_{j\in H}\left[\frac{1}{2\sqrt{ij}}-\frac{1}{4ij}\right]+n^{-1/2}\mathcal{E}\\
&= \mathcal{O}\left(n^{-1/2}\mathbb{E}\left[\sum_{j\in H}X^{i,j}\right]\right)+n^{-1/2}\mathcal{E}
\end{aligned}
$$

The statement follows because $|\deg(i,H)-\sum_{j\in H}X^{i,j}|\le 1.$ ☐

Acknowledgements
Not applicable.

Author contributions
All the authors contributed equally to this work.

Funding
This work was partially supported by Poul Due Jensen Foundation grant no. 883901, the Villum Investigator Grant S4OS, the project SERICS (PE00000014), the Fsc regional Tuscan project AISLEA2 J54D23000780005, the project Tuscany Health Ecosystem (THE), CUP: B83C22003920001, the project SMaRT COnSTRUCT (CUP J53C24001460006), within FAIR (CUP I53C22001380006). The last two projects are under the MUR National Recovery and Resilience Plan funded by the EU - NextGenerationEU.

Data availability
No datasets were generated or analysed during the current study.

Declarations

Competing interest
The authors declare no conflict of interest.

Received: 7 November 2024 / Accepted: 18 July 2025

References

Ahmed NK, Rossi RA, Willke TL, Zhou R (2017) Edge role discovery via higher-order structures. Pacific-Asia conference on knowledge discovery and data mining. Springer, Cham, pp 291–303
Ahuja G (2000) Collaboration networks, structural holes, and innovation: a longitudinal study. Adm Sci Q 45(3):425–455
Akoglu L, Tong H, Koutra D (2015) Graph based anomaly detection and description: a survey. Data Min Knowl Disc 29:626–688
Bacci G, Bacci G, Larsen KG, Tribastone M, Tschaikowski M, Vandin A (2021) Efficient local computation of differential bisimulations via coupling and up-to methods. In: 2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS). pp 1–14
Baier C, Engelen B, Majster-Cederbaum M (2000) Deciding bisimilarity and similarity for probabilistic processes. J Comput Syst Sci 60(1):187–231
Barabási A-L (2012) The science of networks. Perseus, Cambridge MA
Barabási A-L, Bonabeau E (2003) Scale-free networks. Sci Am 288(5):60–69
Batagelj V, Mrvar A (2006) Pajek datasets
Bedru HD, Yu S, Xiao X, Zhang D, Wan L, Guo H, Xia F (2020) Big networks: a survey. Computer science review 37:100247
Bollobás B, Riordan O (2004) The diameter of a scale-free random graph. Combinatorica 24(1):5–34
Borgatti SP, Everett MG (1989) The class of all regular equivalences: algebraic structure and computation. Social networks 11(1):65–88
Borgatti SP, Everett MG (1992) Notions of position in social network analysis. Soc Methodol 1–35
Borgatti SP, Everett MG (1993) Two algorithms for computing regular equivalence. Soc Netw 15(4):361–376
Borgatti SP, Everett MG, Freeman LC (2002) Ucinet for windows: software for social network analysis. Harvard MA analytic technologies 6:12–15
Brandes U, Lerner J (2010) Structural similarity: spectral methods for relaxed blockmodeling. J Classif 27(3):279–306
Brucker P (1978) On the complexity of clustering problems. Optimization and Operations Research: Proceedings of a Workshop Held at the University of Bonn, October 2–8, 1977. Springer, Berlin, pp 45–54
Buchholz P (1994) Exact and ordinary lumpability in finite Markov chains. J Appl Probab 31(1):59–75
Cardelli L, Tribastone M, Tschaikowski M, Vandin A (2017a) Maximal aggregation of polynomial dynamical systems. Proc Natl Acad Sci 114(38):10029–10034

Cardelli L, Tribastone M, Tschaikowski M, Vandin A (2017b) ERODE: A tool for the evaluation and reduction of ordinary differential equations. In: TACAS

Cardelli L, Tribastone M, Tschaikowski M, Vandin A (2018) Guaranteed error bounds on approximate model abstractions through reachability analysis. In: McIver A, Horvath A (eds) Quantitative Evaluation of Systems. Springer, Cham, pp 104–121

Cardelli L, Squillace G, Tribastone M, Tschaikowski M, Vandin A (2023) Formal lumping of polynomial differential equations through approximate equivalences. J Log Algebr Methods Programm 134:100876

Cardelli L, Squillace G, Tribastone M, TchaikowskiM, Vandin A Evaluation, Reduction, and Approximation of Dynamical Systems and Networks withERODE International Symposium on Automated Technology for Verification and Analysis (ATVA)2025 (accepted)

Chamberland M, Straub A (2013) On gamma quotients and infinite products. Adv Appl Math 51(5):546–562

Chen D, Breugel F, Worrell J (2012) On the complexity of computing probabilistic bisimilarity. Foundations of software science and computational structures: 15th International Conference,, FOSSACS 2012, Held as Part of the European joint conferences on theory and practice of software, ETAPS 2012, Tallinn, Estonia, March 24–April 1, 2012. Proceedings 15. Springer, Berlin, pp 437–451

Colizza V, Pastor-Satorras R, Vespignani A (2007) Reaction-diffusion processes and metapopulation models in heterogeneous networks. Nat Phys 3(4):276–282

De Domenico M, Nicosia V, Arenas A, Latora V (2015) Structural reducibility of multilayer networks. Nat Commun 6(1):1–9

Donnat C, Zitnik M, Hallac D, Leskovec J (2018) Learning structural node embeddings via diffusion wavelets. In: Proceedings of the 24th ACM SIGKDD International conference on knowledge discovery & data mining. pp 1320–1329

Doreian P, Batagelj V, Ferligoj A (2005) Generalized blockmodeling, vol 25. Cambridge University Press, Cambridge

Everett MG (1985) Role similarity and complexity in social networks. Soc Netw 7(4):353–359

Everett MG, Boyd JP, Borgatti SP (1990) Ego-centered and local roles: a graph theoretic approach. J Math Sociol 15(3–4):163–172

Freeman SC, Freeman LC (1979) The networkers network: a study of the impact of a new communications medium on sociometric structure. School of Social Sciences University of Calif..???

Funke T, Becker T (2019) Stochastic block models: a comparison of variants and inference methods. PLoS ONE 14(4):0215296

Gaulier G, Zignago S (2010) BACI: International trade database at the product-level. the 1994-2007 version. Working Papers 2010–23, CEPII. http://www.cepii.fr/CEPII/fr/publications/wp/abstract.asp?NoDoc=2726

Jin D, Heimann M, Rossi RA, Koutra D (2019a) Node2bits: Compact time-and attribute-aware node representations for user stitching. Joint European conference on machine learning and knowledge discovery in databases. Springer, Cham, pp 483–506

Jin D, Heimann M, Safavi T, Wang M, Lee W, Snider L, Koutra D (2019b) Smart roles: Inferring professional roles in email networks. In: Proceedings of the 25th ACM SIGKDD International conference on knowledge discovery & data mining. pp 2923–2933

Kane GC, Alavi M, Labianca G, Borgatti SP (2014) What s different about social media networks? a framework and research agenda. MIS Q 38(1):275–304

Kunegis J (2013) Konect: the koblenz network collection. In: Proceedings of the 22nd International Conference on World Wide Web. pp 1343–1350

Lorrain F, White HC (1971) Structural equivalence of individuals in social networks. J Math Soc 1(1):49–80

Luczkovich JJ, Borgatti SP, Johnson JC, Everett MG (2003) Defining and measuring trophic role similarity in food webs using regular equivalence. J Theor Biol 220(3):303–321

Marx M, Masuch M (2003) Regular equivalence and dynamic logic. Soc Netw 25(1):51–65

Matjasiĉ M, Cugmas M, Ẑiberna A (2020) Blockmodeling: an R package for generalized blockmodeling. Adv Methodoland Stat 17(2):49–66

Milner R (1982) A Calcul Commun Syst. Springer, Berlin, Heidelberg

Nikolentzos G, Vazirgiannis M (2019) Learning structural node representations using graph kernels. IEEE Trans Knowl Data Eng 33(5):2045–2056

Opsahl T, Panzarasa P (2009) Clustering in weighted networks. Soc Netw 31(2):155–163

Orsenigo L, Pammolli F, Riccaboni M, Bonaccorsi A, Turchetti G (1997) The evolution of knowledge and the dynamics of an industry network. J Manag Govern 1:147–175

Paige R, Tarjan R (1987) Three partition refinement algorithms. SIAM J Comput 16(6):973–989

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830

Peixoto TP (2019) Bayesian stochastic blockmodeling. In: Advances in network clustering and blockmodeling. pp 289–332

Petrov T, Tognazzi S (2021) Exact and approximate role assignment for multi-layer networks. J Complex Netw 9(5):027. https://doi.org/10.1093/comnet/cnab027

Reichardt J, White DR (2007) Role models for complex networks. Eur Phys J B 60:217–224

Ribeiro LF, Saverese PH, Figueiredo DR (2017) struc2vec: Learning node representations from structural identity. In: Proceedings of the 23rd ACM SIGKDD International conference on knowledge discovery and data mining. pp 385–394

Rossi RA, Gallagher B, Neville J, Henderson K (2013) Modeling dynamic behavior in large evolving graphs. In: Proceedings of the sixth ACM international conference on web search and data mining. pp 667–676

Rossi RA, Jin D, Kim S, Ahmed NK, Koutra D, Lee JB (2020) On proximity and structural role-based embeddings in networks: misconceptions, techniques, and applications. ACM Trans Knowl Discov Data (TKDD) 14(5):1–37

Sailer LD (1978) Structural equivalence: meaning and definition, computation and application. Soc Netw 1(1):73–90

Smith DA, White DR (1992) Structure and dynamics of the global economy: network analysis of international trade 1965–1980. Soc Forces 70(4):857–893

Squillace G, Tribastone M, Tschaikowski M, Vandin A (2024) Efficient Network Embedding by Approximate Equitable Partitions, 2024 IEEE International Conference on Data Mining (ICDM), Abu Dhabi, United Arab Emirates, pp. 440–449.

Tu K, Cui P, Wang X, Yu PS, Zhu W (2018) Deep recursive network embedding with regular equivalence. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & data mining. pp 2357–2366

Valmari A, Franceschinis G (2010) Simple $O(m \log n)$ time markov chain lumping. In: Esparza, J., Majumdar, R. (eds.) Tools and Algorithms for the Construction and Analysis of Systems, 16th International Conference, TACAS, vol. 6015, pp. 38–52

Watts DJ, Strogatz SH (1998) Collective dynamics of small-world networks. Nature 393(6684):440–442

White DR, Reitz KP (1983) Graph and semigroup homomorphisms on networks of relations. Soc Netw 5(2):193–234

Zafarani R, Liu H (2009) Social Computing Data Repository at ASU. http://socialcomputing.asu.edu

Žiberna A (2007) Generalized blockmodeling of valued networks. Soc Netw 29(1):105–126

Ziberna A (2008) Direct and indirect approaches to blockmodeling of valued networks in terms of regular equivalence. J Math Soc 32(1):57–84

Žiberna A (2009) Evaluation of direct and indirect blockmodeling of regular equivalence in valued networks by simulations. Adv Methodol Stat 6(2):99–134

Publisher's Note