Role-equivalent Actors in Networks *
Group Structures Beyond Dense Communities

Ulrik Brandes and Jürgen Lerner**

Department of Computer & Information Science, University of Konstanz

Abstract. Communities in social networks are often defined as groups of densely connected actors. However, members of the same dense group are not equal but may differ largely in their social position or in the role they play. Furthermore, the same positions can be found across the borders of dense communities so that networks contain a significant group structure which does not coincide with the structure of dense groups. This papers gives a survey over formalizations of network-positions with a special emphasis on the use of algebraic notions.

1 Introduction

A common trend in recent years is the emergence of very large networks, i. e., sets of objects (later called *vertices*) together with one or more relations. Examples include networks of humans together with kinship or friendship relations, *affiliation networks* where actors are connected by common participation in events or common membership in organizations, *authorship networks* where researchers are connected by co-authored documents, *citation networks* where articles point to other articles, *customer-product networks* where customers are linked to the products they bought or evaluated, or online discussion groups where users respond to other users. Concrete examples include Amazon's "who purchased this, also purchased that"-network, the WWW where Web-pages link to other pages, and the online encyclopedia Wikipedia where users are co-authoring articles and articles point to other articles.

While it is generally believed that valuable information is contained in these networks, their sheer size makes it a challenging task to extract this information automatically. Network analysis methods (see [28] and [4] for an overview) try to achieve different goals. A well-known issue is the computation of the *importance* or *centrality* of vertices (see, e. g., Chapt. 5 of [28]). For instance, Google's *PageRank* [6] defines the importance of Web-pages relative to a user's query and thereby facilitates seeing the most important results first. Other examples are popularity-rankings of users in social networking sites. A different goal in network analysis is the computation of densely connected groups of vertices (see, e. g., Chapt. 7 of [28]). The hope is that these clusters correspond to natural divisions of the network into, e. g., articles or documents treating similar topics,

* Research supported by DFG under grant Br 2158/2-3

** Corresponding author, lerner@inf.uni-konstanz.de

researchers working on the same problems, or customers interested in the same products. An early definition for densely connected groups is that of a *clique* which is a subset of pairwise connected vertices. Obviously this definition is very strict and will normally result in rather small groups, compare [13]. More stable notions require that clusters do not have to be perfect cliques but have to be more densely connected within the group than to the outside. See, e. g., [14] for different formalizations of this idea.

A third issue in network analysis aims at computing groups of vertices that occupy the same *structural position* or play the same *role* in the network (see, e. g., Chapts. 9–12 of [28], or [19]). In this article we review several formalizations of network-positions with an emphasis on the resulting algebraic structures. Since we believe that the notion of structural position of vertices is not as well-known as the notion of, e. g., dense groups of vertices, we give an intuition of positions or roles in social networks in the following Section 1.1. Figure 1 shows a small network together with an intuitive partition into classes of vertices that occupy the same structural positions.

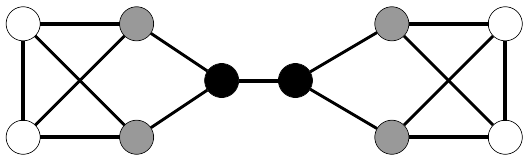

Fig. 1. Example graph taken from [1]. The coloring defines classes of vertices that (intuitively) occupy the same structural position.

1.1 Social Role and Social Position

Formally, methods that compute role assignments partition the vertex set of a graph into several classes. However, in contrast to clustering where these classes have to be densely connected, role assignments try to identify classes of vertices that occupy the same social position, play the same role, or have the same function in the network. To illustrate this distinction, employees of a company that work closely together are likely to form a dense group, e. g., due to having frequent email or face-to-face contact. However, densely connected employees may occupy different positions (like, e. g., manager or secretary) and these differences are reflected in their mutual relations. However, positions are normally not so evident, since social actors do not always occupy *institutionalized* positions (like manager or secretary). In many social networks, actors appear uniform or at least differences in their positions are not known. In this situation we might still discover positions dependent on the relations these actors have with other actors. For instance, Turner et al. [27] analyzed the patterns of user interaction in Usenet

groups and identified several *types* of users (in our notation, groups of users that occupy the same position), among others the *answer person*, *questioner*, *troll* (someone who likes to draw others into useless discussions), *spammer*, and *flame warrior*. Of course, these positions are not institutionalized (nobody enters a Usenet group officially as a questioner—even less as a spammer). Instead, the positions are determined by the pattern of interaction these users have with other users. Moreover, note that these types of users do not correspond to dense groups. It is simply impossible that a dense group of users (users that have frequent interaction) is solely composed of, e. g., questioners.

To summarize these considerations we conclude that in many networks there is a meaningful group structure which does not coincide with the partition into dense clusters and that these groups are defined (or reflected) by similar patterns of interaction to other groups. Since these positions are dependent on the relations, i. e., the graph structure, we will use the term *structural positions*.

1.2 Outline of this Paper

Until now we have only given an intuition of the notion of structural position. In the remainder of this article we review different formalizations of structural positions and analyze the properties of the so-defined sets of role assignments.

In Sect. 3 we treat discrete approaches, i. e., formalizations of structural positions which identify two vertices as either equivalent or non-equivalent. These Boolean notions define certain ideal types of role structures and are convenient to understand conceptual differences between various approaches. However, these discrete measures are of limited applicability in the context of empirical, huge, and possibly noisy networks. Major drawbacks are the fact that results are often trivial (very small classes), highly sensitive to small changes in the input data, computationally intractable, and unable to deal with varying importance of vertices or with vertices that do not fit exactly into one of the classes.

To overcome these drawbacks, a series of measures have been proposed that do not require vertices to be equivalent or non-equivalent but that define varying *degrees of similarity* between two vertices. These measures are more robust to the irregularities of application data. We review some of them in Sect. 4.

Furthermore, Sect. 3 and Sect. 4 are both divided into two parts. The first parts (Sects. 3.1 and 4.1) describe formalizations that recognize two vertices as equivalent (similar) if they have identical (largely overlapping) neighborhoods. For instance, two actors in a social network are considered as similar by these measures if they are connected to almost the same other actors. However, this understanding of structural position is a very limited point of view. Coming back to the Usenet example, two users may be equivalent in their role (e. g., may both be answer persons) without replying to any common other user. To overcome this drawback several formalizations have been proposed that recognize vertices as equivalent if their neighborhoods themselves are equivalent (but not necessarily identical). These discrete proposals will be reviewed in Sect. 3.2; their real-valued counterpart (which defines similarity of vertices instead of equivalence or non-equivalence) are introduced in Sect. 4.2. The partition in Fig. 1 is an example

of a role assignment in which equivalent vertices have equivalent (same-colored) but not necessarily identical neighborhoods.

2 Preliminaries

An (*undirected*) *graph* $G = (V, E)$ consists of a set of *vertices* V (the objects or actors) and a symmetric relation $E \subseteq V \times V$, whose elements are called *edges*. For sake of simplicity we will only consider undirected graphs but most notions of this paper can be adapted to directed graphs. If $v \in V$, then $N(v) = \{u \in V\,;\ (u, v) \in E\}$ denotes the *neighborhood* of v. For more on graph theory see, e. g., [9]. If $\sim \subseteq V \times V$ is an *equivalence relation* and $v \in V$ then $[v] = \{u\,;\ u \sim v\}$ is called the *equivalence class* of v. The set of equivalence classes defines a *partition* of V. The mapping $r\colon v \mapsto [v]$ which maps a vertex to its class will be called a *role assignment*. Equivalence relations, partitions, and role assignments are mutually in a canonical one-to-one correspondence. If two vertices $u, v \in V$ are equivalent we say that u and v occupy the same *position*, have the same *status*, or play the same *role*. Two specific and trivial partitions are the *identity partition* in which every vertex occupies a different position (hence the classes of the identity partition are all singleton sets) and the *complete partition* in which all vertices occupy the same position (hence the only class of the complete partition is the complete vertex set). Equivalence relations on V can be partially ordered by $\sim_1 \leq \sim_2 \Leftrightarrow \sim_1 \subseteq \sim_2$ ($\sim_1$ is then called *finer* than $\sim_2$). This partial order is a *lattice* (see, e. g., [17]) where the infimum of two equivalence relations is given by their intersection and the supremum is given by the transitive closure of their union. The identity partition is the minimum element and the complete partition the maximum element of this lattice.

3 Discrete Approaches

3.1 Requiring Neighborhood Identity

If two individuals have the same neighbors they cannot be distinguished from the point of view of any individual in the network. This idea has given rise to two different formalizations of vertex equivalence which will be presented in Sect. 3.1. The two formulations differ in how neighbors within the own class are treated.

Structural Equivalence. The most simple, but also most restrictive requirement of role equivalence has been defined by Lorrain and White [20] who proposed that individuals are role equivalent if they are related to the same individuals.

Definition 1. *Let $G = (V, E)$ be a graph, and r a role assignment on V. Then, r is called* structural *if for all $u, v \in V$*

$$r(u) = r(v) \Longrightarrow N(u) = N(v) \ .$$

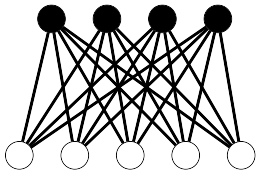

Fig. 2. The bipartition of a complete bipartite graph (indicated by the vertex coloring) defines a structural equivalence.

Trivially, the identity partition is structural for each graph. A slightly less trivial example is the bipartition of complete bipartite graphs, see Fig. 2.

We note some elementary properties of structural equivalence. A class of structurally equivalent vertices is either an independent set (i. e., contains no edges, compare Fig. 2) or a clique. The distance of two structurally equivalent, non-isolated vertices is at most 2, since if u and v are structurally equivalent and u has a neighbor w then w is also a neighbor of v. Thus, structural equivalence can only identify vertices that are close to each other.

It can easily be verified that if $\sim_1$ and $\sim_2$ are two structural equivalences for a graph, then so are their intersection and the transitive closure of their union.

Proposition 1. *The set of structural equivalences of a graph is a sublattice of the lattice of all equivalence relations.*

In particular there exists always a maximum structural equivalence (MSE) for a graph. The property of being structural is preserved under refinement which implies that the whole set of structural equivalences is completely described by the MSE.

Proposition 2. *If $\sim_1 \leq \sim_2$ and $\sim_2$ is a structural equivalence, then so is $\sim_1$.*

Modular Decomposition. We now present another compatibility requirement that yields a convenient decomposition of a graph and is widely used in graph theory and combinatorics. Here it is required that all vertices that lie in one class must have identical neighborhoods *outside* their own class, whereas structural equivalence required identical neighborhoods without any restriction. We do not know of any article that proposed modular decomposition for defining role assignments in social networks. Gagneur *et al.* [15] used modular decomposition to identify *functional complexes* in protein-protein interaction networks. The following is mostly adapted from [23].

Let $G = (V, E)$ be a graph. A subset of vertices $M \subseteq V$ is called a *module* if for any $v \in V \setminus M$, either v is adjacent to every member of M, or v is adjacent to no member of M. It is easy to see that V and the singleton subsets are always modules, called *trivial modules*. A partition $\mathcal{P}$ of V is called a *congruence partition* if every class of $\mathcal{P}$ is a module. Thus, the complete partition and the

identity partition are congruence partitions for all graphs. The *quotient* $G/\mathcal{P}$ of a graph G modulo a congruence partition $\mathcal{P}$ is defined to be the subgraph induced by a set of representatives of each class of $\mathcal{P}$. This is well-defined since each class is a module, i. e., a different choice of representatives yields an isomorphic subgraph.

We say that two sets *overlap* if they intersect and neither of them contains the other and call a module *strong* if it overlaps no other module. Figure 3 shows a small affiliation network (vertices connected if they participate in the same events) together with its strong modules.

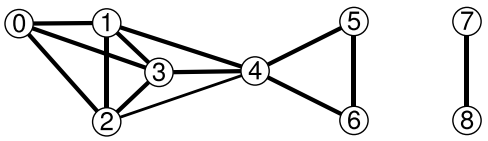

Fig. 3. Affiliation network G resulting from the affiliations $W = \{0, 1, 2, 3\}$, $X = \{1, 2, 3, 4\}$, $Y = \{4, 5, 6\}$, and $Z = \{7, 8\}$. The strong modules of G are: $\{1, 2, 3\}$, $\{5, 6\}$, $\{7, 8\}$, $\{0, 1, 2, 3, 4, 5, 6\}$, and the trivial modules (singleton sets and the complete vertex set). For instance, $\{\{0\}, \{1, 2, 3\}, \{4\}, \{5, 6\}, \{7, 8\}\}$ is a congruence partition for G, which reveals the vertices that participate in the same set of affiliations.

Let $\mathcal{M}$ be the set of strong modules. We say that a strong module M_1 is a (direct) *child* of another strong module M_2 if $M_1 \subset M_2$ and there is no $M \in \mathcal{M}$ such that $M_1 \subset M \subset M_2$. This child-relation on $\mathcal{M}$ defines a rooted tree whose nodes are the elements of $\mathcal{M}$, whose root is V, and that has one leaf for each vertex $v \in V$. A node of this tree corresponds to the element of $\mathcal{M}$ that contains exactly the leaf descendants of that node. Such a tree will be called a *union tree* on V. The union tree represents $\mathcal{M}$ in $\mathcal{O}(|V|)$ space. A central observation is that all modules of G can be represented by the union tree defined by its strong modules (and hence in $\mathcal{O}(|V|)$ space). A node U is called *degenerate* if the union of any subfamily of the direct children of U is a module; it is called *prime* if no union of any nontrivial subfamily of the direct children of U is a module.

Theorem 1 ([23]). *A strong module is either degenerate or prime.*

All modules of G may thus be represented by constructing the union tree of its strong modules and labeling the nodes as degenerate or prime. This labeled union tree is called the *modular decomposition tree* (*MD-tree*) of G. The modular decomposition tree of a graph $G = (V, E)$ can be computed in $\mathcal{O}(|V| + |E|)$ time by the algorithm from [23]. See Fig. 4 for the MD-tree of the graph from Fig. 3.

A node X in the MD-tree induces a graph whose vertices are the (direct) children of X (which are strong modules of G). Two such children are adjacent if nodes contained in them are adjacent. Thus, the graph induced by a node of the MD-tree is the quotient of a subgraph of G.

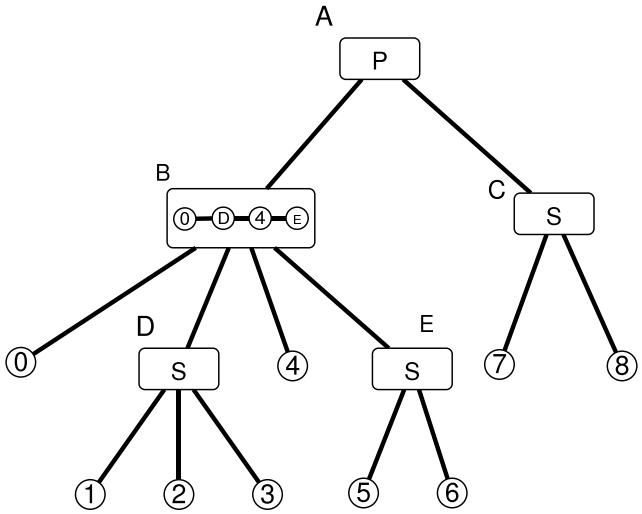

Fig. 4. MD-tree T of the graph G shown in Fig. 3. The leaves of T are the vertices of G. The internal nodes (A, B, C, D, and E) of T correspond to the strong modules that are not singleton sets. They are labeled by S if the induced graph is complete, by P if the induced graph is edgeless, and by the induced graph if it is prime. For instance, the node B induces a prime graph that is a path connecting the the modules $\{0\}$, $D = \{1, 2, 3\}$, $\{4\}$, and $E = \{5, 6\}$.

If a node of the modular decomposition tree is prime, it induces a graph that has no nontrivial modules. Such a graph is called *prime*. If a node of the modular decomposition tree is degenerate, it induces a graph in which every subset of nodes is a module. Such a graph is called *degenerate*. A degenerate graph is either complete or edgeless [23].

A degenerate node that induces a complete graph is called an S-node, a degenerate node that induces an edgeless graph is called a P-node. If we label nodes in the MD-tree as S or P if they are degenerate and with the induced subgraph if they are prime, then the MD-tree encodes not only the set of modules of G but it is also possible to reconstruct G from this labeled MD-tree, compare Fig. 4.

Limitations of requiring identical neighborhoods. Several researchers have pointed out that the requirement of identical neighborhoods does not meet the intuition of structural position (see, e. g., [26, p.78] or [21, p.304]). We illustrated this in the context of Usenet groups at the end of Sect. 1.2. As an abstract example, consider Fig. 5, where the white vertices are indistinguishable in terms of the graph's structure (since they are automorphic images of each other). However, two white vertices connected to different black vertices

have disjoint neighborhoods and hence would not be recognized as similar by measures that require (almost) identical neighborhoods.

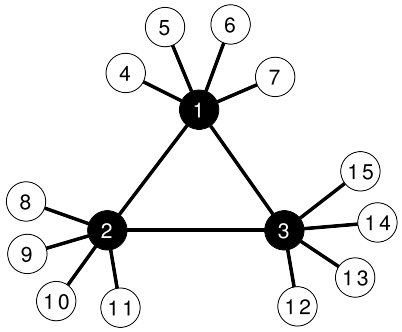

Fig. 5. Graph with a vertex partition indicated by the coloring. Vertices with identical neighborhoods (like $\{4, 5, 6, 7\}$) can be recognized by equivalences defined in Sect. 3.1 (or by measures from Sect. 4.1). However, the fact that all white vertices are structurally indistinguishable is not discovered by these formalizations.

3.2 Requiring Neighborhood Equivalence

To overcome the limitations mentioned above, role assignments have been formalized by measures that require *equivalent* neighborhoods for equivalent vertices.

Automorphic Equivalence. Two vertices that are automorphic images of each other are indistinguishable with respect to structural properties. Automorphic equivalence defines that exactly these vertices occupy the same structural position.

Definition 2 ([11]). *Let $G = (V, E)$ be a graph, $u, v \in V$. Then u and v are said to be* automorphically equivalent *if there is an automorphism φ of G with $\varphi(u) = v$.*

Let $G = (V, E)$ be a graph and H a subgroup of G's automorphism group (not necessarily proper). An *orbit* of the action of H on V is a subset of vertices of the form $\{\varphi(v)\,;\ \varphi \in H\}$, for a $v \in V$. It is well known that the orbits of a group of automorphisms define a partition of V, called an *orbit partition*. For example, the coloring in Fig. 5 defines the orbit partition of the (entire) automorphism group. It is easy to see that structurally equivalent vertices are automorphically equivalent. No efficient algorithms are known to compute automorphic equivalence. Furthermore, classes of automorphically equivalent vertices will be rather small for most empirical networks.

Equitable Partitions. The partition shown in Fig. 5 has the property that every white vertex has exactly one black neighbor and zero white neighbors and every black vertex has exactly four white neighbors and two black neighbors. Hence, it satisfies the condition of the following definition.

Definition 3 ([16]). *Let $G = (V, E)$ be a graph and $c\colon V \to \{1, \ldots, k\}$ a coloring of the vertex set. Then c defines an* equitable partition *if $c(u) = c(v)$ implies that for every color c_0 the neighborhoods of u and v contain the same number of vertices colored c_0.*

Equitable partitions are called *divisors of graphs* in [8]. In the context of social network analysis partitions of this type have been called *exact colorations* [11]. Equitable partitions are indeed a relaxation of automorphic equivalence.

Proposition 3 ([10]). *Orbit partitions are equitable.*

The set of orbit equivalences forms a proper subset of the set of all equitable partitions. For example, the complete partition for the graph in Fig. 1 is equitable but not an orbit partition. The coloring in Fig. 1 defines an orbit partition and hence an equitable partition.

The compatibility requirement for equitable partitions is still too strict for empirical graphs, as the defined partitions will be almost trivial (having very small classes).

Regular Equivalence. The term regular equivalence has been introduced by White and Reitz [29]. It is closely related to the notion of *bisimulation* (see [24] and [22]). A coloring defines a regular equivalence if vertices that are colored the same have the same colors in their neighborhoods. In contrast to equitable partitions, multiple occurrence of a color makes no difference for regular equivalence. If r is a role assignment on V and $U \subseteq V$ then $r(U) = \{r(u)\,;\ u \in U\}$ is the set of classes (or colors) that members of U have.

Definition 4. *Let $G = (V, E)$ be a graph. A role assignment r on V is called* regular *if for all $u, v \in V$*

$$r(u) = r(v) \Rightarrow r(N(u)) = r(N(v)) \ .$$

An example of a regular equivalence can be seen in Fig. 6. Note that, in contrast to equitable partitions, regularly equivalent vertices can have different degrees.

In empirical graphs equitable partitions (and hence orbit partitions and structural equivalence) frequently lead to partitions whose classes are very small. In contrast, the maximal regular equivalence is close to the complete partition (which is also trivial):

Proposition 4. *The maximal regular equivalence of an undirected graph is the division into isolated and non-isolated vertices.*

Also for directed graphs the complete partition is regular if every vertex has at least one in-coming and one out-going edge. Since the maximal regular equivalence is mostly trivial, it is worthwhile to consider the structure of the set of regular equivalence to identify non-trivial members of it.

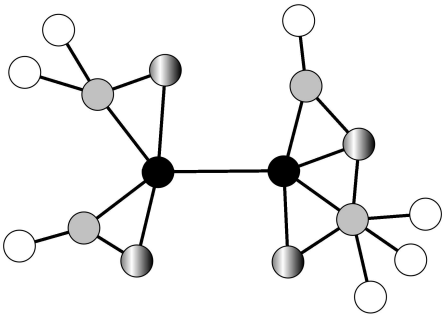

Fig. 6. Graph with a non-trivial regular equivalence indicated by the vertex coloring. Note that the maximal regular equivalence of this graph is the complete partition.

Theorem 2 ([1]). *The set of all regular equivalences of a graph G forms a lattice, where the supremum operation is identical to the supremum in the lattice of all equivalences.*

Figure 7 shows a small graph together with its lattice of regular equivalences.

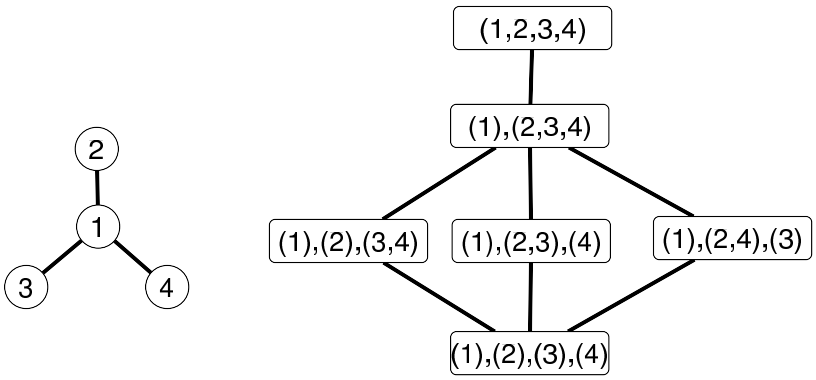

Fig. 7. Graph (*left*) and its lattice of regular equivalence relations (*right*).

Although the supremum in the lattice of regular equivalences coincides with the supremum in the lattice of all equivalences, the infimum is different.

Proposition 5. *The intersection of two regular equivalences is not necessarily regular.*

Proof. Consider the graph in Fig. 8. The intersection of the two regular partitions $\mathcal{P}_1$ and $\mathcal{P}_2$ is $\mathcal{P} = \{\{A, C\}, \{B, D\}, \{E\}\}$, which is not regular. □

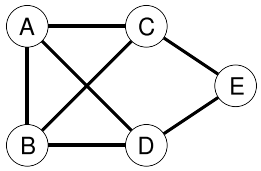

Fig. 8. The intersection of the regular equivalences $\mathcal{P}_1 = \{\{A, C, E\}, \{B, D\}\}$ and $\mathcal{P}_2 = \{\{A, C\}, \{B, D, E\}\}$ is not regular for this graph.

The fact that the supremum in the lattice of regular equivalences coincides with the supremum in the lattice of all equivalences implies the existence of a maximum regular equivalence which lies below a given (arbitrary) equivalence. Thus it is possible to partition a network (e. g., using some *a priori* knowledge about different types of vertices) and then "regularizing" this partition by splitting up some classes.

Definition 5 ([3]). *Let G be a graph and $\sim$ an equivalence relation on its vertex set. An equivalence relation $\sim_1$ is called the* regular interior *of $\sim$ if $\sim_1$ is regular, $\sim_1 \leq \sim$, and for all $\sim_2$ satisfying the former two conditions it holds $\sim_2 \leq \sim_1$.*

Another name for regular interior is *coarsest regular refinement*. The maximal regular equivalence of a graph is the regular interior of the complete partition.

Corollary 1. *Let G be a graph and $\sim$ an equivalence relation on its vertex set. Then the regular interior of $\sim$ exists. On the other hand there is in general no minimum regular equivalence above a given equivalence (called* regular closure *or* regular hull*).*

Proof. The first part has been shown in [3]. For the second part recall the example in the proof of Prop. 5 shown in Fig. 8. It is easy to verify that the regular partitions $\mathcal{P}_1$ and $\mathcal{P}_2$ are both above the (non-regular) partition $\mathcal{P}$ and are both minimal with this property. □

Corollary 2 ([3]). *The infimum in the lattice of regular equivalence relations is given by the regular interior of the intersection.*

Computation of regular equivalences. The regular interior of a given input partition (and hence the maximal regular equivalence) can be computed quite efficiently. The algorithm CATREGE [2] is the most well-known in the social network literature. It runs in time $\mathcal{O}(n^3)$ on a graph with n vertices. Paige and Tarjan [25] presented a sophisticated algorithm for the *relational coarsest partition*

problem which is essentially equivalent to computing the regular interior. Their algorithm runs in $\mathcal{O}(m \log n)$ time on a graph with n vertices and m edges. This is a significant improvement especially for sparse graphs whose number of edges is linear in n.

However, as we have seen above, the maximal regular equivalence is often trivial. Hence we might wish to compute a regular equivalence with a given number of equivalence classes. Unfortunately, this task is $\mathcal{NP}$-hard in general [12]. In particular, this implies that computing the whole set of regular equivalences is (probably) not possible in polynomial time. For some graphs this follows already from the fact that the size of this set is not polynomially bounded.

4 Relaxations of Discrete Approaches

The Boolean definitions for structural positions presented in Sect. 3 do often not fit well to empirical data since in real world applications vertices are not necessarily either equivalent or non-equivalent but might be just be "similar". In Sect. 4 we introduce real-valued solutions that assign similarity values to pairs of vertices. These methods are more robust to the irregularities of application data.

4.1 Similarity by Neighborhood Overlap

Two vertices are structurally equivalent (see Def. 1) if they have exactly the same neighbors. Frequently, vertices are defined as similar if their neighborhoods have large overlap. The notion of neighborhood overlap can be normalized in many different ways. We present only one of these measures and refer the reader to Chapt. 9 of [28] for additional possibilities.

A simple measure is to count how many neighbors two vertices have in common and normalize this over the size of the union of the two neighborhoods.

$$\sigma(u, v) = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|} \ .$$

The notion of neighborhood overlap can also be extended to the overlap of higher order neighborhoods (e. g., the second order neighborhood of a vertex v consists of those vertices that have distance two to v). Examples include [7, 18], where vertices are defined as similar if they are connected by "many" paths of length k, and where k goes through all integers (for structural equivalence k must be two). The approaches differ in how the number of connecting paths is normalized, and how paths of length k are counted. However, as with structural equivalence, these measures do not assign high similarity to vertices that lie in different dense clusters, even if these vertices cannot be distinguished in terms of the graph's structure. For instance, these measures will recognize Vertices 1 and 4 in Fig. 5 as more similar than, e. g., 4 and 8. Therefore, these measures do not overcome the limitations described at the end of Sect. 3.1. Furthermore, it seems to be likely that these measures are biased to detecting dense subgraphs and thus do not distinguish conceptually from density based graph clustering.

4.2 Requiring Neighborhood Similarity

In this section we define similarity of vertices, under the requirement that similar vertices have to be connected to vertices that are themselves similar. This is the "real-valued" version of the equivalences from Sect. 3.2. We only sketch the most important definitions and theorems and refer to [5] for a more exhaustive treatment.

We represent an equivalence relation $\sim$ on a set of n vertices is by its *characteristic matrix* S, which is defined to be the $n \times n$ matrix

$$S_{uv} = \begin{cases} 1/r & \text{if } u \sim v \text{ and } r \text{ is the size of } v\text{'s equivalence class} \\ 0 & \text{if } u \not\sim v \ . \end{cases}$$

The normalization by r seems to be arbitrary at the moment. However, this normalization simplifies formulas later. In any case the so-defined characteristic matrices are equivalent to the more usual Boolean matrices (which have only zero or one as entries) since there is a unique one-to-one correspondence between them. The characteristic matrices of equivalence relations satisfy the identities $S^{\mathrm{T}} = S$ and $S^2 = S$ (due to the symmetry and transitivity of equivalence relations) and have the property that, after appropriate reordering of the rows and columns, they have block-diagonal form with constant diagonal blocks (since vertices are "equally equivalent" to all members of their class and not equivalent to members outside their class). See Fig. 1 for a vertex partition and Fig. 9 for its characteristic matrix. To define relaxations of equivalence relations we drop

$$S = \begin{bmatrix} 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 1/4 \\ 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 1/4 \\ 0 & 0 & 1/4 & 1/4 & 0 & 0 & 1/4 & 1/4 & 0 & 0 \\ 0 & 0 & 1/4 & 1/4 & 0 & 0 & 1/4 & 1/4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/4 & 1/4 & 0 & 0 & 1/4 & 1/4 & 0 & 0 \\ 0 & 0 & 1/4 & 1/4 & 0 & 0 & 1/4 & 1/4 & 0 & 0 \\ 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 1/4 \\ 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 1/4 \end{bmatrix}$$

Fig. 9. Characteristic matrix of the vertex partition shown in Fig. 1. The matrix S defines a structural similarity for this graph.

the restriction on the block-diagonal form but keep the requirements $S^{\mathrm{T}} = S$ and $S^2 = S$.

Definition 6 ([5]). *Let V be a set of n vertices. A real $n \times n$ matrix S is called a* similarity *if it is symmetric, i. e. $S^{\mathrm{T}} = S$, and idempotent, i. e. $S^2 = S$.*

The real value in row u and column v is called the *similarity* of vertices u and v. We call a similarity structural if it is compatible with the graph structure.

In Sect. 3.2 it is required that equivalent vertices have equivalent (e. g., same colored) neighborhoods. Here we require that vertices which are similar have neighbors that are themselves similar. This is ensured by the condition in the next definition. If $G = (V, E)$ is a graph on n vertices then its *adjacency matrix* is defined to be the $n \times n$ matrix A whose rows and columns are indexed by the vertices of G and where $A_{uv} = 1$ if $(u, v) \in E$ and $A_{uv} = 0$ else.

Definition 7 ([5]). *Let $G = (V, E)$ be a graph with adjacency matrix A and S a similarity on V. Then S is called* structural *(for G) if $AS = SA$.*

The next theorem shows that structural similarities indeed ensure that similar vertices have similar neighbors.

Theorem 3 ([5]). *Let $G = (V, E)$ be a graph and S the characteristic matrix of an equivalence relation $\sim$ on V. Then S is structural if and only if $\sim$ is an equitable partition. In particular, automorphic equivalences induce structural similarities.*

For instance the similarity shown in Fig. 9 is structural for the graph in Fig. 1. However, structural similarities are not limited to strict equitable partitions (which often lead to very small classes of vertices). In the next theorem, we derive an equivalent condition for a similarity to be structural. This condition yields a compact description of the set of structural equivalences of a graph and shows that non-trivial structural similarities exist for all graphs.

Theorem 4 ([5]). *Let $G = (V, E)$ be a graph with adjacency matrix A and S a similarity on V. Then S is structural if and only if* $\ker S$ *is generated by eigenvectors of A if and only if* $\operatorname{img} S$ *is generated by eigenvectors of A.*

For instance the image of the similarity shown in Fig. 9 is generated by eigenvectors of the graph in Fig. 1 associated to the eigenvalues 3, 1 and -2.

Lattice of Structural Similarities. Theorem 4 not only yields an efficient method for computing structural similarities and a compact description of the set of all structural similarities of a graph—it also provides means to prove a lattice structure on this set.

If $\sim_1$ and $\sim_2$ are two equivalence relations on a set V and S_1 respectively S_2 the associated similarities, then $\sim_1$ is finer than $\sim_2$ ($\sim_1 \leq \sim_2$) if and only if $\ker S_1 \subseteq \ker S_2$. We will use this correspondence to generalize the partial order of equivalence relations to a partial order on the set of similarities. We establish first a connection between similarities and subspaces.

Lemma 1 ([5]). *Let $\mathcal{U} \subseteq \mathbb{R}^n$ be a subspace. Then there is a unique similarity S such that $\mathcal{U} = \ker S$.*

By the above lemma, we get a bijection between the set of subspaces of $\mathbb{R}^n$ and the set of similarities on V. Via this bijection the set of similarities can be supplied with a partial order $\leq$, where we say that a similarity S_1 is *finer* than a similarity S_2, denoted by $S_1 \leq S_2$, if $\ker S_1 \subseteq \ker S_2$.

The set of all subspaces of $\mathbb{R}^n$ is a complete lattice: If $\mathcal{M} = \{\mathcal{U}_i\,;\ i \in I\}$ is a set of subspaces of $\mathbb{R}^n$ (not necessarily finite), then

$$\inf(\mathcal{M}) = \{v \in \mathbb{R}^n\,;\ \forall i \in I \text{ it is } v \in \mathcal{U}_i\}$$

is the infimum of $\mathcal{M}$. Conversely, the subspace

$$\sup(\mathcal{M}) = \{v \in \mathbb{R}^n\,;\ \exists k \in \mathbb{N},\, v_1 \in \mathcal{U}_i, \ldots, v_k \in \mathcal{U}_k\colon\ v = v_1 + \cdots + v_k\}$$

is the supremum of $\mathcal{M}$. It follows that the set of similarities on a vertex set V is a complete lattice too.

Theorem 5 ([5]). *The set of structural similarities of a graph is a complete sublattice of the lattice of all similarities on its vertex set.*

Theorem 5 implies (see [5]) that given a similarity S that is not necessarily structural, there is always a finest structural similarity above S and a coarsest structural similarity below S. Note, that this does not hold for regular equivalence relations, where only the regular interior exists but not the regular hull (see Corollary 1).

5 Conclusion

We reviewed several proposals for role-equivalence or role-similarity of actors in networks. Some formalizations (like structural equivalence, automorphic equivalence, or equitable partitions) are likely to yield very small classes of equivalent vertices in irregular application data. Other approaches (like regular equivalence or structural similarities) define larger sets of possible role-assignments. An important (and in general unsolved) task is the identification of "good" or "meaningful" elements of these sets. It is yet to be explored whether their lattice structure can help in solving this task.

References

1. S. P. Borgatti and M. G. Everett. The class of all regular equivalences: Algebraic structure and computation. *Social Networks*, 11(1):65–88, 1989.
2. S. P. Borgatti and M. G. Everett. Two algorithms for computing regular equivalence. *Social Networks*, 15(4):361–376, 1993.
3. J. P. Boyd and M. G. Everett. Relations, residuals, regular interiors, and relative regular equivalence. *Social Networks*, 21(2):147–165, Apr. 1999.
4. U. Brandes and T. Erlebach, editors. *Network Analysis*, volume 3418 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.
5. U. Brandes and J. Lerner. Structural similarity in graphs. In *Proceedings of the 15th International Symposium on Algorithms and Computation (ISAAC'04)*, volume 3341 of *Lecture Notes in Computer Science*, pages 184–195, 2004.
6. S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

7. R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences*, 102(21):7426–7431, 2005.
8. D. M. Cvetković, M. Doob, and H. Sachs. *Spectra of Graphs*. Johann Ambrosius Barth, 1995.
9. R. Diestel. *Graph Theory*. Springer-Verlag, New York, 2000.
10. M. G. Everett and S. P. Borgatti. Role colouring a graph. *Mathematical Social Sciences*, 21:183–188, 1991.
11. M. G. Everett and S. P. Borgatti. Regular equivalence: General theory. *Journal of Mathematical Sociology*, 18(1):29–52, 1994.
12. J. Fiala and D. Paulusma. The computational complexity of the role assignment problem. In *Proceedings of the 30th International Colloquium on Automata, Languages, and Programming (ICALP'03)*, pages 817–828. Springer-Verlag, 2003.
13. L. C. Freeman. Cliques, galois lattices, and the structure of human social groups. *Social Networks*, 18:173–187, 1996.
14. M. Gaertler. Clustering. In Brandes and Erlebach [4], pages 187–215.
15. J. Gagneur, R. Krause, T. Bouwmeester, and G. Casari. Modular decomposition of protein-protein interaction networks. *Genome Biology*, 5(8):R57, 2004.
16. C. Godsil and G. Royle. *Algebraic Graph Theory*. Graduate Texts in Mathematics. Springer-Verlag, 2001.
17. G. Grätzer. *General Lattice Theory*. Birkhäuser Verlag, 1998.
18. E. A. Leicht, P. Holme, and M. E. J. Newman. Vertex similarity in networks. *Physical Review E*, 73, 2006.
19. J. Lerner. Role assignments. In Brandes and Erlebach [4], pages 216–252.
20. F. Lorrain and H. C. White. Structural equivalence of individuals in social networks. *Journal of Mathematical Sociology*, 1:49–80, 1971.
21. J. J. Luczkovich, S. P. Borgatti, J. C. Johnson, and M. G. Everett. Defining and measuring trophic role similarity in food webs using regular equivalence. *Journal of Theoretical Biology*, 220(3):303–321, 2003.
22. M. Marx and M. Masuch. Regular equivalence and dynamic logic. *Social Networks*, 25:51–65, 2003.
23. R. M. McConnel and J. P. Spinrad. Modular decomposition and transitive orientation. *Discrete Mathematics*, 201:189–241, 1999.
24. R. Milner. *A Calculus of Communicating Systems*. Springer Verlag, Berlin, 1980.
25. R. Paige and R. E. Tarjan. Three partition refinement algorithms. *SIAM Journal on Computing*, 16(6):973–983, 1987.
26. L. D. . Sailer. Structural equivalence: Meaning and definition, computation and application. *Social Networks*, 1:73–90, 1978.
27. T. C. Turner, M. A. Smith, D. Fisher, and H. T. Welser. Picturing usenet: Mapping computer-mediated collective action. *Journal of Computer-Mediated Communication*, 10(4), 2005.
28. S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.
29. D. R. White and K. P. Reitz. Graph and semigroup homomorphisms on networks of relations. *Social Networks*, 5:193–234, 1983.