Weak Invariants of Actions of the Automorphism Group of a Graph

Fabian Ball and Andreas Geyer-Schulz

Abstract Automorphism groups of graphs may lead to multiple equivalent solutions of graph-clustering algorithms and to a certain degree of arbitrariness in selecting one or more solution(s) as well as to problems with partition comparison measures. Knowledge of the automorphism group is crucial for stability analysis, for evaluating the degree of arbitrariness involved in selecting a solution, as well as for a further classification as congruent solutions or structurally equivalent solutions. For this purpose we identify three weak invariants of group actions of the automorphism group of a graph, namely modularity, partition type, and the Kolmogorov-Sinai entropy. In particular, we extend the Kolmogorov-Sinai entropy for measuring the uncertainty in finite permutation groups and we apply the underlying construction for testing if multiple structurally equivalent solutions exist for a given graph partition.

Fabian Ball
Information Services and Electronic Markets,
Karlsruhe Institute of Technology (KIT) Kaiserstraße 12, D-76131 Karlsruhe, Germany,
✉ ball@kit.edu

Andreas Geyer-Schulz
Information Services and Electronic Markets,
Karlsruhe Institute of Technology (KIT) Kaiserstraße 12, D-76131 Karlsruhe, Germany
✉ andreas.geyer-schulz@kit.edu

ARCHIVES OF DATA SCIENCE (ONLINE FIRST) DOI 10.5445/KSP/1000058749/02
KIT SCIENTIFIC PUBLISHING ISSN 2363-9881
Vol. 2, No. 1, 2017

1 Introduction

Clustering a graph is an algorithmic technique of finding a partition of the node set of a graph from the given data of the graph (node set and list of edges) so that the groups of vertices forming clusters have common properties with regard to their structure or interrelationship. It is used to find structures in real networks (e.g. the Internet, social networks, biological and chemical reaction networks, ...) (Fortunato, 2010). The algorithm-focused view on graph clustering already indicates that the solutions of this undertaking are the solutions of optimization problems that maximize or minimize some local or global criterion (the *clustering criterion*). The optimum can either be unique or there exist several equivalent optima which may be structurally identical, "congruent" (defined later) or a mixture of both. The possibility of multiple equivalent solutions raises the question "What are the true clusters?" (Hennig, 2015). Multiple structurally identical solutions (generated from the automorphism group of the graph) provide yet another example of the pervasive problem of multiplicity which P. Diaconis has identified 30 years ago (Diaconis, 1985).

Finding the automorphism group of a graph is equivalent to the graph isomorphism problem. These problems are computationally hard and expensive (even after more than 35 years of intensive research). McKay's backtracking algorithm `nauty` (based on McKay (1981)) in its most recent incarnation is still one of the most promising practical approaches (McKay and Piperno, 2014). Weisfeiler (1976) studied algorithmic ideas of attacking the graph isomorphism problem: In addition to "almost" brute force enumeration (McKay's approach) he presented the idea of using graph-invariants which can be computed fast for pruning the search space. However, only weak invariants were found and discussed. In the following we present three weak invariants related to graph clustering which hopefully can be used to reduce the problem complexity: Modularity, partition type, and the Kolmogorov-Sinai entropy. Two of these are available directly from a single clustering solution (Sects. 4.1 and 4.2). The Kolmogorov-Sinai entropy (Sinai (1959), Sect. 4.3) is a concept mainly coming from the study of ergodic dynamical systems.

2 Notation and Concepts

As a graph we consider a tuple $G = (V,E)$ with V the set of nodes (or often called vertices/points) and $E \subset V \times V$ the set of edges. In the context of this paper all graphs are undirected, connected (for every pair of nodes a path between them exists), unweighted (every edge has no or at least constant weight set to 1) and has no self edges (loops). Further we define $|V| = n < \infty$ (finite graphs) and $|E| = m$. A partition $P(\Omega)$ of a set Ω is a set of subsets $C_i \subseteq \Omega$ having the properties of being complete ($\bigcup_i C_i = \Omega$), not having empty subsets ($C_i \neq \emptyset$) and being disjoint ($C_i \cap C_j = \emptyset$ for all $i \neq j$). For $\Omega = V$ we call $P(V)$ (or simply P) partition of the graph $G = (V,E)$, the node subsets C_i are called clusters.

To investigate the problems raised by multiple solutions, we must give some foundations of permutation groups (see Wielandt, 1964) and derive the definition of graph automorphisms for permutation groups: A permutation is a bijective function $f : \Omega \to \Omega$ mapping each point $\alpha_i \in \Omega$ onto a point $\beta_i \in \Omega$:

$$\begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \beta_1 & \beta_2 & \dots & \beta_n \end{pmatrix}.$$

For brevity, the cycle notation $f = c_1 c_2 \dots c_k$ is often used, each $c_i = (\gamma_{i_1}\ \gamma_{i_2}\ \dots\ \gamma_{i_r})$ is a cycle that maps γ_{i_k} onto $\gamma_{i_{k+1}}$ and γ_{i_r} onto γ_{i_1}. Single cycles (γ_{j_1}) are omitted in notation. The composition of permutations $f \circ g = h$ is again a permutation and the mapping is performed from right to left (i.e. $h(\alpha) = f(g(\alpha))$). We call a permutation id that maps each point in Ω onto itself the identity and a permutation f^{-1} for that $f \circ f^{-1} = f^{-1} \circ f = id$ the inverse of f. Furthermore $f^k := f \circ \dots \circ f$ (k times f composed with f) and $f^0 := id$. A set of permutation functions H is called a permutation group if

1. $\forall f, g \in H : f \circ g \in H$ (Closure),
2. $id \in H$ (Existence of unit element),
3. $\forall f \in H : f^{-1} \in H$ (Existence of inverse elements) and
4. $\forall f, g, h \in H : f \circ (g \circ h) = (f \circ g) \circ h$ (Associativity)

holds.

When a permutation group H acts on a graph G, we write $G^f = (V^f, E^f), f \in H$ where $V^f := \{f(v) \mid v \in V\}$) and $E^f := \{\{f(u), f(v)\} \mid \{u,v\} \in E\}$. We say "$f$ acts on G". A permutation group is the automorphism group of G (we write $Aut(G)$) if and only if $G^f = G, \forall f \in Aut(G)$ and $Aut(G)$ must be maximal

($\nexists f' \notin Aut(G) : G^{f'} = G$). A permutation acting on a set Ω also acts on diverse combinatorial constructions of Ω, such as partitions.

A metric for a space S (with $s, t, u \in S$) is a distance function from $S \times S$ into $\mathbb{R}^+$ with symmetry ($d(s,t) = d(t,s)$), identity ($d(s,t) = 0$ if and only if $s = t$), and where the triangle inequality $d(s,u) \leq d(s,t) + d(t,u)$ holds. A pseudometric space S^*, d has a relaxed identity condition ($d(s,s) = 0$) and, instead of the space S, the space S^* which consists of equivalence classes of subsets of S is used (see Doob, 1994, p. 5).

3 Problems Caused by Multiple Structurally Equivalent Solutions

In this section we illustrate two problems of the presence of multiple structurally equivalent (MSE) solutions for graph clustering:

1. Partition-comparison measures are nonconstant when comparing solutions from a set of MSE solutions with a cardinality larger than 2.
2. What is the true cluster structure in the MSE solution set? Does the MSE solution set contain a true cluster structure at all?

3.1 The Problem of Nonconstant Partition-Comparison Measures

Example 1. The *Variation of Information* (`VI`, Meilă, 2003) measures the distance of pairs of partitions and is proven to be a metric by the author. It is defined as $VI(P,P') = H(P) + H(P') - 2MI(P,P')$ where $H(P) = -\sum_{i=1,\ldots,|P|} \frac{|C_i|}{n} \log \frac{|C_i|}{n}$ is the entropy of a partition and $MI(P,P') = \sum_{C \in P} \sum_{C' \in P'} \frac{|C \cap C'|}{n} \log n \frac{|C \cap C'|}{|C||C'|}$ is the mutual information between both partitions.

The three partitions of the C_9 (shown in Fig. 1) $P = \{\{v_0, v_1, v_2\}, \{v_3, v_4, v_5\}, \{v_6, v_7, v_8\}\}$, $R = \{\{v_1, v_2, v_3\}, \{v_4, v_5, v_6\}, \{v_7, v_8, v_0\}\}$ and $S = \{\{v_2, v_3, v_4\}, \{v_5, v_6, v_7\}, \{v_8, v_0, v_1\}\}$ are easily identified as **structurally equivalent** by "just looking at the graph" and they have the same optimal modularity (see Sect. 4.1).

We expect that measures which can identify structurally equivalent partitions (e.g. P and R above) have a constant distance $d(P,R) = 0$. The entropy H for all three partitions is $-3 \cdot (\frac{3}{9} \log_2 \frac{3}{9}) = -\log_2 \frac{1}{3} = \log_2 3$ and the mutual information is $3 \cdot (\frac{2}{9} \log_2 9 \cdot \frac{2}{3 \cdot 3} + \frac{1}{9} \log_2 9 \cdot \frac{1}{3 \cdot 3}) = \frac{2}{3} \log_2 2 = \frac{2}{3}$ (each cluster of one partition

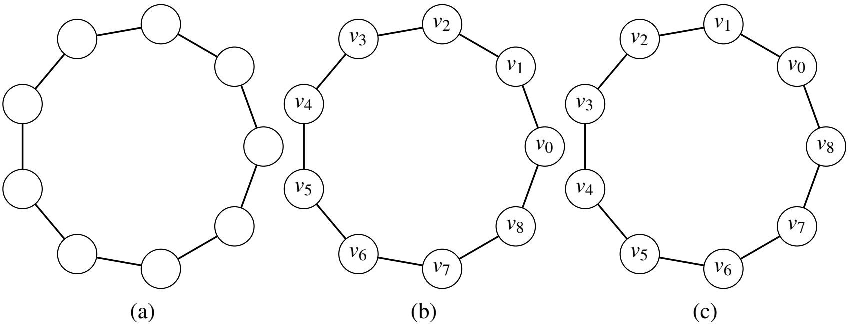


Fig. 1 A 2-regular cyclic graph C_9. k-regularity means each node $v_i \in V$ has degree k, for undirected graphs the degree of a node is the number of incident edges $\deg(v_i) := |\{e \in E \mid v_i \in e\}|$. Every 2-regular (connected!) graph is automatically a cyclic graph, w.l.o.g. $C_n = (\{v_0, \ldots, v_{n-1}\}, \{\{v_i, v_{i+1}\} \mid i, (i+1) \in \mathbb{Z}/n\mathbb{Z}\})$. The permutation $f = (v_0\ v_1\ \ldots\ v_8)$ maps the representation in (b) to the representation in (c)

compared to the three clusters of the other partition has once overlap two and once overlap one). However, the distances measured by `VI` yield $d(P,P) = 0.0$, but $d(P,R) = 1.837$, $d(P,S) = 1.837$, and $d(R,S) = 1.837$. For a C_n with larger n, `VI` will vary with the degree of overlap between clusters.

If d' is a pseudometric distance function, $d'(P,P) = d'(P,R) = d'(P,S) = \ldots = d'(S,S) = 0.0$ would hold, as $P,R,S \in [P]$ ($[P]$ is the equivalence class).

`VI` is not a suitable measure for identifying structurally equivalent solutions. All known distance-, similarity-, or dissimilarity-measures routinely used by practitioners of cluster analysis are unsuitable for identifying structurally equivalent solutions.

3.2 The Problem of the True Cluster Structure in a Set of MSE Solutions

Example 2. The graph shown in Fig. 1 has a non trivial automorphism group $|Aut(C_9)| > 1$ (i.e. not only the identity acts on the graph) and e.g. $f = (v_0\ v_1\ \ldots\ v_8) \in Aut(C_9)$. One can easily check that $P = S^f = R^{f^2}$ (see Example 1). The solutions are structurally equivalent as automorphic mappings define structural equivalency. The automorphism group of the graph is implicitly derived from the graph's structure and it is important to keep in mind that this means the properties are present in the graph data and not because they were subsequently introduced by any assumptions we added.

The existence of a set of MSE solutions poses the question how the cluster structure found by the cluster algorithm relates to the true cluster structure and in the case of the example above where the automorphism group acts nontrivially on all nodes if there exists a true cluster structure at all or if we may pick one of the solutions P, R, S arbitrarily.

Graph automorphisms are the common cause of both multiplicity problems illustrated above:

- For partition comparison measures used for the evaluation of cluster solutions, they violate the axiom that $d(x,y) = 0$ implies that $x \equiv y$, (d is a measure between two points x and y of some space S). In measure theory, e.g. Doob (1994) handles this problem formally by replacing this axiom of a metric space by the axiom of pseudometric space which states that $d(x,y) = 0$ does not imply $x \equiv y$.
  In applications, the transformation from a metric space to a pseudometric space requires the identification of the equivalence classes defined by the automorphism groups and the replacement of members of an equivalence class by the "canonical" representative of the class before evaluating the solution of the cluster problem.
- For the interpretation of the existence and/or nature of the true cluster structure in a graph we also need to know the automorphism group of the graph. At least in the natural sciences, group actions are linked to natural laws.

Progress with regard to both problems requires finding the automorphism group of a graph. Unfortunately, this is equivalent to solving the graph isomorphism problem which is still considered a hard problem.

In Sect. 4 we identify three weak invariants of graph automorphisms which are necessary but not sufficient for structurally equivalent partitions. We suggest to use them for detecting candidates for non-trivial automorphism functions.

4 Weak Invariants of Graph Automorphisms

An invariant is a condition which is always true up to some transformations of the input. In our context it is some function $m : X \to Y$ for which $m(x) = m(x^f)\, \forall f \in Aut(G), x \in X$ holds. X is the space of some combinatorial construction of the node set V of G, for our purpose X is the space of all partitions of V. Such a measure is considered weak if a transformation $f \notin Aut(G)$ with $m(x^f) = m(y)$ exists (for any G).

4.1 Modularity: A Measure for Partition Quality

The modularity introduced by Newman and Girvan (2004) is a measure of quality of some partition P of a graph G. It is defined as the summation over all clusters $C_i \in P$ of the difference of the edge fraction and the expected edge fraction if the edges would be rewired randomly:

$$Q(G,P) = \sum_{i=1,\dots,|P|} (e_{ii} - a_i^2). \tag{1}$$

The two components are defined as

$$e_{ii} = \frac{|\{\{u,v\} \in E | u,v \in C_i\}|}{|E|} \tag{2}$$

and

$$a_i = e_{ii} + \frac{|\{\{u,v\} \in E | u \in C_i, v \notin C_i\}|}{2|E|}. \tag{3}$$

The factor $\frac{1}{2}$ in Eq. 3 is important because every edge connecting clusters is counted once for each incident cluster, therefore twice all in all. Due to the possibility of iterative computation of Q, starting from a partition of singletons (clusters consisting of exactly one node), modularity can be used as cluster criterion (e.g. Ovelgönne et al, 2010). See Fortunato and Barthélemy (2007); Brandes et al (2008) for additional details.

From Eqs. 2 and 3 it is clear that modularity is computed only on adjacency information. Because every automorphism $f \in Aut(G)$ preserves adjacency by definition, $Q(G,P) = Q(G^g,P^g)$ holds and proves modularity to be an invariant measure. However, modularity is only a weak invariant measure, because for many graphs partitions exist which are structurally different but result in the same value for Q. For example, consider the clusters $C_1 = \{u_1,u_2,u_3,u_4\}$, $C_2 = \{w\}$, and $C_3 = \{v_1,v_2,v_3,v_4\}$ which partition the graph shown in Fig. 2. C_1 and C_3 are congruent.

We call subgraphs **congruent** if they have the same number of nodes and inner edges plus the same connection structure with the rest of the graph.

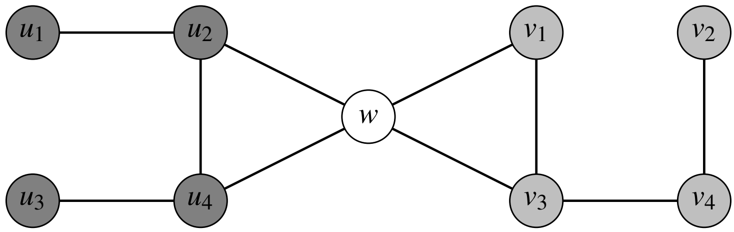


Fig. 2 A "nearly point-symmetric" graph (around w) where (1) the two clusters formed of the gray nodes are similar but structurally different and (2) the two two-cluster-partitions formed of the gray nodes and the center node w (once part of the darker gray, once part of the light gray cluster) have the same modularity and integer partition but cannot be mapped onto each other by an automorphism of the graph.

4.2 The Integer Partition

A partition of an integer $n \in \mathbb{N}$ is some series $a_1, a_2, a_3, \ldots$ where $\sum_i a_i = n$. The definition of a partition from Sect. 1 was based on sets but if we let $a_i = |C_i|$, every partition of a set has an associated integer partition which is unique up to permutation of the objects. For faster comparison we write the integer partition as sequence $(a_{i_1}, a_{j_2}, a_{k_3}, \ldots)$ partially ordered by $\geq$ and for brevity we compact the sequence by writing $(a^{\#a}, b^{\#b}, c^{\#c}, \ldots)$ ($\#x$ is the count of occurrences of integer x). E.g. the singleton partition with n elements is denoted by (1^n).

Let now t be a function that maps a partition of a set to its integer partition. We call this the type of the partition and can of course also apply t to partitions of graphs.

We can make use of the same argumentation as in the previous subsection because graph automorphisms do not change the adjacency (and are permutations), the type of a partition does not change: $\forall f \in Aut(G) : t(P) = t(P^f)$ (see also James, 1978, p. 6). Again, counterexamples can be found where partition type **and** modularity are equal but there does not exist an automorphism that maps the partitions onto each other. This proves the partition type to be only a weak invariant measure. For example, consider the following two partitions of the graph shown in Fig. 2: $\hat{C}_1 = \{u_1, u_2, u_3, u_4, w\}$, $\hat{C}_2 = \{v_1, v_2, v_3, v_4\}$ and $\bar{C}_1 = \{u_1, u_2, u_3, u_4\}$, $\bar{C}_2 = \{v_1, v_2, v_3, v_4, w\}$.

4.3 The Kolmogorov-Sinai Entropy

In this subsection we introduce the Kolmogorov-Sinai entropy, one of the concepts for which Sinai received the Abel Prize 2014 (Sinai, 1959). The Kolmogorov-Sinai entropy is a weak invariant of graph automorphisms. We show that the Kolmogorov-Sinai entropy (and its computation) provides a diagnostic of the stability of a graph with regard to its automorphism group $Aut(G)$. Computation of the entropy of a point allows the classification of the nodes of the graph in stable (fixed by the permutations of $Aut(G)$) and unstable nodes (moved by the permutations of $Aut(G)$) with regard to the automorphism group of the graph.

4.3.1 Definition

Sinai (1959) defined the Kolmogorov-Sinai entropy in the setting of M, a Lebesgue measure space with σ-algebra S of measurable subsets of M, and a probability measure μ with an arbitrary automorphism T of M. He investigated the properties of applying T t-times on a finite partition $\mathscr{P} = \{\mathscr{C}_1, \ldots, \mathscr{C}_k\}$ of M and what happens to the entropy of $h(\mathscr{P}) = -\sum_{i=1}^{k} \mu(\mathscr{C}_i) \log_2 \mu(\mathscr{C}_i)$ in the limit $k \to \infty$ (Sinai, 2010, p. 3):

Definition 1. The Kolmogorov-Sinai entropy of an automorphism T is the supremum of $h_T(\mathscr{P})$ over all finite partitions $\mathscr{P}$: $h_T = \sup_{\mathscr{P}} h_T(\mathscr{P})$.

Building on Shannon's well known entropy definition in Sect. 4.3.2, we show in detail in Sects. 4.3.3 and 4.3.4 how the Kolmogorov-Sinai entropy is constructed and computed for the automorphism group $Aut(G)$ of a graph.

4.3.2 Shannon's Entropy on Symbols and Blocks

In his seminal article on information theory Shannon (1948) measured the amount of information transmitted over a communication channel as the entropy of the string of symbols $\ldots x_0 x_1 x_2 \ldots$ where each x_i is an element of a fixed alphabet $A = \{a_1, \ldots, a_k\}$: If $\mathbb{P}(a_i)$ is the probability of receiving a_i, for all $i = 1, \ldots, k$, then the amount of information transmitted per symbol on average is defined by the entropy $H = -\sum_{i=1}^{k} \mathbb{P}(a_i) \log_2 \mathbb{P}(a_i)$.

For a general source (e.g. an English text), the probability of receiving a given symbol depends on what other symbols have already been received. This dependency structure is captured by blocking symbols into groups: For each $k = 1,2,3,\ldots$ let B_k denote the family of all blocks of k-symbols from A. Each block B in B_k has a certain probability $\mathbb{P}(B)$ to be received. The average information per symbol in a transmission of length k is then

$$H_k = -\frac{1}{k} \sum_{B \in B_k} \mathbb{P}(B) \log_2 \mathbb{P}(B) \tag{4}$$

and the entropy of the source for an infinite symbol stream exists (Petersen, 1983, p. 231) and is

$$h = \lim_{k \to \infty} -\frac{1}{k} \sum_{B \in B_k} \mathbb{P}(B) \log_2 \mathbb{P}(B). \tag{5}$$

4.3.3 Kolmogorov-Sinai Entropy for Aut(G)

To better point out what happens when automorphisms act on a graph and how nodes (identified by their labels v_i) wander through a partition, we additionally assign to each node an arbitrary color $\upsilon_i \in \mathscr{V}$. The function $col : V \to \mathscr{V}$ outputs the color of a node, the function $lab : V \to \mathscr{L}$ the label (or name of a node). Automorphisms of a graph can be seen as permutations of the labels or as permutations of the colors. This is a consequence of the internal representation of a node as a pair of identifiers: Its color and its label. If we fix the colors, we apply permutation functions from $Aut(G)$ only to the labels (and the other way round). This can be imagined as a concrete representation of the graph on a two dimensional plane where the circles that represent nodes have a fixed position (identified by their color) and $Aut(G)$ permutes the labels under the conditions we defined in Sect. 2.

A measure-preserving system $(X, \mathscr{X}, \mu, T)$ consists of a probability space $(X, \mathscr{X}, \mu)$ and a measure-preserving transformation T on it: $T : X \to X$ is a measurable transformation on the probability space for which $\forall \mathscr{C} \in \mathscr{X} : \mu(T^{-1}(\mathscr{C})) = \mu(\mathscr{C})$ holds.

The measure-preserving system $(X, \mathscr{X}, \mu, T)$ for the cyclic graph C_4 in Fig. 3 and $g = (v_1\, v_2\, v_3\, v_4)$ is set up in the context of graph automorphisms by the following:

1. The set X is the set V of nodes,

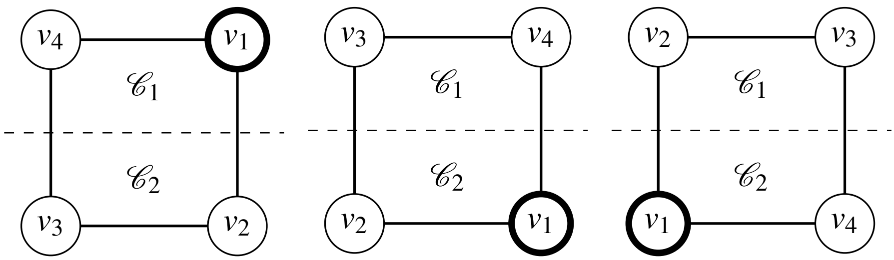
2. the sigma-algebra $\mathscr{X}$ is the set of all subsets of V. We denote a subset of $\mathscr{X}$ as $\mathscr{C}_i$,
3. $\mu(\mathscr{C}_i)$ is the probability that an element $v \in V$ is in subset $\mathscr{C}_i$, and
4. $T : X \to X$ is the graph automorphism g.

An element $v \in V$ is in subset $\mathscr{C}_i$ exactly when $col(v) = \mathscr{C}_i$.

Figure 3 shows how we attach a finite state machine fs to the automorphism g of the graph C_4 which is partitioned as $\mathscr{P} = \{\mathscr{C}_1, \mathscr{C}_2\}$. We tacitly assume that partitions have a canonical labeled representation (which we did by the coloring of nodes), the technical details of canonical labeling (called painting by Rudolph) can be found e.g. in Rudolph (1990).

The elements of the finite alphabet A are the names (symbols) of the subsets $\mathscr{C}_1$ and $\mathscr{C}_2$ in the partition $\mathscr{P}$. The function $sym(subset)$ prints the name of the subset. The state space of the finite state machine fs (constructed for $\mathscr{P}$ and g) is A. The state transition function of fs is defined as follows: At time t the finite state machine fs prints out symbol $\mathscr{C}_i$ if node v is in partition $\mathscr{C}_i$ after t applications of g, short:

$$fs(v^{g^t}) = sym(\mathscr{C}_i), \quad \texttt{if} \quad v^{g^t} \in \mathscr{C}_i. \tag{6}$$



(a) State 0 ($C_4^{g^0}$):
Output symbol $fs(v_1^{g^0}) = \mathscr{C}_1$
Symbol stream $\mathscr{S}_{0,0} = \mathscr{C}_1$

(b) State 1 ($C_4^{g^1}$):
Output symbol $fs(v_1^{g^1}) = \mathscr{C}_2$
Symbol stream $\mathscr{S}_{0,1} = \mathscr{C}_1\mathscr{C}_2$

(c) State 2 ($C_4^{g^2}$):
Output symbol $fs(v_1^{g^2}) = \mathscr{C}_2$
Symbol stream $\mathscr{S}_{0,2} = \mathscr{C}_1\mathscr{C}_2\mathscr{C}_2$

Fig. 3 A C_4 with the group action $g = (v_1 v_2 v_3 v_4)$ and the partition $\{\mathscr{C}_1, \mathscr{C}_2\}$ with a finite state machine fs which reports the symbol of the cluster a node is member of after $0, 1, 2$ applications of g. The symbol stream we can observe starting from time 0 to t is denoted $\mathscr{S}_{0,t}$

Of course, we can compute the probability $\mathbb{P}(v_j \in \mathscr{C}_i)$ that a certain element v_j is in a certain subset $\mathscr{C}_i$ of the finite partition $\mathscr{P}$ after t applications of the group action. In Fig. 3 and Table 1 we write $\mathbb{P}(\cdot \mid \mathscr{S}_{0,t})$ to point out that the

t	0	1	2	3	4	5	6	7	8	…	∞
Action	v_1g^0	v_1g^1	v_1g^2	v_1g^3	v_1g^4	v_1g^5	v_1g^6	v_1g^7	v_1g^8	…	v_1g^∞
State v_1	$\mathscr{C}_1$	$\mathscr{C}_2$	$\mathscr{C}_2$	$\mathscr{C}_1$	$\mathscr{C}_1$	$\mathscr{C}_2$	$\mathscr{C}_2$	$\mathscr{C}_1$	…	…	$\mathscr{C}_1 \mid \mathscr{C}_2$
$\mathbb{P}_t(v_1 \in \mathscr{C}_1 \mid \mathscr{S}_{0,t})$	1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{3}{5}$	$\frac{1}{2}$	$\frac{3}{7}$	$\frac{1}{2}$	…	…	$\frac{1}{2}$
$\mathbb{P}_t(v_1 \in \mathscr{C}_2 \mid \mathscr{S}_{0,t})$	0	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{2}{5}$	$\frac{1}{2}$	$\frac{4}{7}$	$\frac{1}{2}$	…	…	$\frac{1}{2}$
$H(\mathscr{P}, v_1, g^t)$	0	1	0.918	1	0.970	1	0.985	1	…	…	1
Action	v_2g^0	v_2g^1	v_2g^2	v_2g^3	v_2g^4	v_2g^5	v_2g^6	v_2g^7	v_2g^8	…	v_2g^∞
State v_2	$\mathscr{C}_2$	$\mathscr{C}_2$	$\mathscr{C}_1$	$\mathscr{C}_1$	$\mathscr{C}_2$	$\mathscr{C}_2$	$\mathscr{C}_1$	$\mathscr{C}_1$	…	…	$\mathscr{C}_1 \mid \mathscr{C}_2$
$\mathbb{P}_t(v_2 \in \mathscr{C}_1 \mid \mathscr{S}_{0,t})$	0	0	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{2}{5}$	$\frac{1}{3}$	$\frac{3}{7}$	$\frac{1}{2}$	…	…	$\frac{1}{2}$
$\mathbb{P}_t(v_2 \in \mathscr{C}_2 \mid \mathscr{S}_{0,t})$	1	1	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{3}{5}$	$\frac{2}{3}$	$\frac{4}{7}$	$\frac{1}{2}$	…	…	$\frac{1}{2}$
$H(\mathscr{P}, v_2, g^t)$	0	0	0.918	1	0.970	0.918	0.985	1	…	…	1
Action	v_3g^0	v_3g^1	v_3g^2	v_3g^3	v_3g^4	v_3g^5	v_3g^6	v_3g^7	v_3g^8	…	v_3g^∞
State v_3	$\mathscr{C}_2$	$\mathscr{C}_1$	$\mathscr{C}_1$	$\mathscr{C}_2$	$\mathscr{C}_2$	$\mathscr{C}_1$	$\mathscr{C}_1$	$\mathscr{C}_2$	…	…	$\mathscr{C}_1 \mid \mathscr{C}_2$
$\mathbb{P}_t(v_3 \in \mathscr{C}_1 \mid \mathscr{S}_{0,t})$	0	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{2}{5}$	$\frac{1}{2}$	$\frac{4}{7}$	$\frac{1}{2}$	…	…	$\frac{1}{2}$
$\mathbb{P}_t(v_3 \in \mathscr{C}_2 \mid \mathscr{S}_{0,t})$	1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{3}{5}$	$\frac{1}{2}$	$\frac{3}{7}$	$\frac{1}{2}$	…	…	$\frac{1}{2}$
$H(\mathscr{P}, v_3, g^t)$	0	1	0.918	1	0.970	1	0.985	1	…	…	1
Action	v_4g^0	v_4g^1	v_4g^2	v_4g^3	v_4g^4	v_4g^5	v_4g^6	v_4g^7	v_4g^8	…	v_4g^∞
State v_4	$\mathscr{C}_1$	$\mathscr{C}_1$	$\mathscr{C}_2$	$\mathscr{C}_2$	$\mathscr{C}_1$	$\mathscr{C}_1$	$\mathscr{C}_2$	$\mathscr{C}_2$	…	…	$\mathscr{C}_1 \mid \mathscr{C}_2$
$\mathbb{P}_t(v_4 \in \mathscr{C}_1 \mid \mathscr{S}_{0,t})$	1	1	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{3}{5}$	$\frac{2}{3}$	$\frac{4}{7}$	$\frac{1}{2}$	…	…	$\frac{1}{2}$
$\mathbb{P}_t(v_4 \in \mathscr{C}_2 \mid \mathscr{S}_{0,t})$	0	0	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{2}{5}$	$\frac{1}{3}$	$\frac{3}{7}$	$\frac{1}{2}$	…	…	$\frac{1}{2}$
$H(\mathscr{P}, v_4, g^t)$	0	0	0.918	1	0.970	0.918	0.985	1	…	…	1

Table 1 A group action $g = (v_1\ v_2\ v_3\ v_4)$ of $Aut(C_4)$ as a state transition function of the dynamical system shown in Fig. 3. A column corresponds to the t-th transition of the evolving system. Each column shows for all of the four node labels v_i of the partitioned graph from Fig. 3 in which state it is after the t-th transition. Furthermore, the probability distributions for the two states ($\mathbb{P}_t(\cdot \mid \mathscr{S}_{0,t})$ where $\mathscr{S}_{0,t}$ is the observed symbol stream) and the derived entropy ($H(\cdot)$) are denoted. Of course, only some of the first steps are shown as the system evolves infinitely often. When the system tends to infinity, we can see in the last column that state probabilities are equally distributed which leads to the maximum possible entropy!

probabilities are computed conditionally on the observed symbol stream $\mathscr{S}_{0,t}$. For the rest of this paper we omit this notation. Table 1 shows the first few steps of this construction. The **entropy of the partition** $\mathscr{P} = \{\mathscr{C}_1, \dots, \mathscr{C}_k\}$ **at** v_j **for** $g \in Aut(G)$ is

$$H(\mathscr{P}, v_j, g) = -\sum_{i=1}^{k} \mathbb{P}(v_j \in \mathscr{C}_i) \log_2 \mathbb{P}(v_j \in \mathscr{C}_i). \tag{7}$$

For example, what is the entropy of the partition $\mathscr{P}$ at v_4 for g for $t = 6$? From Table 1, $H(\mathscr{P}, v_4, g^6) = 0.985$. And for $t = \infty$? The answer is provided by Birkhoff's pointwise ergodic theorem (Birkhoff, 1931):

Theorem 1. *Let $(V, \mathscr{X}, \mu)$ be a probability space, $g : V \to V$ a measure preserving transformation and $f \in L^1(V, \mathscr{X}, \mu)$. Then $\lim_{t\to\infty} \frac{1}{t} \sum_{i=0}^{t-1} f(v^{g^i}) = \overline{f}(v)$ exists almost everywhere.*

We state Birkhoff's theorem without proof. For proofs we refer either to Birkhoff's original paper (Birkhoff, 1931) or Petersen (1983, pp. 27-33). We illustrate the working of the dynamic system for the cyclic graph C_4 with the partition $\mathscr{P} = \{\mathscr{C}_1, \mathscr{C}_2\}$ and the action g of $Aut(C_4)$ in Table 1. The group action g is measure preserving. In Table 1 we also show three examples for the $\overline{f}(v)$, namely the relative frequencies that a node v_i is in subset $\mathscr{C}_1$ or $\mathscr{C}_2$ at time t ($\mathbb{P}_t(v_i \in \mathscr{C}_1)$ and $\mathbb{P}_t(v_i \in \mathscr{C}_2)$) and the entropy of this distribution $H(\mathscr{P}, v_i, g^t)$. By the pointwise ergodic theorem, all three functions converge.

The entropy of nodes v_i $(i = \{1,2,3,4\})$ for partition $\mathscr{P}$ is 1 and so is the entropy of partition $H(\mathscr{P}, g)$ (that is $H(\mathscr{P}, v, g^t)$ for $t \to \infty$, $\forall v \in V$). Informally, this is the average entropy of the 4 symbol streams generated by g at the 4 nodes. The **entropy of the partition $\mathscr{P}$ for the group action** g is

$$H(\mathscr{P}, g) = \frac{1}{n} \sum_{j=1}^{n} H(\mathscr{P}, v_j, g). \tag{8}$$

We define a second state transition function: At time t the finite state machine $\overline{fs}$ prints out a symbol $\mathscr{C}_i$ for each node $v \in V$ if node v is in partition $\mathscr{C}_i$ after t applications of g (i.e. four symbols for the C_4, one for each node):

$$\overline{fs}(V^{g^t}) = \left(fs(v_1^{g^t}), \ldots, fs(v_n^{g^t})\right)^T. \tag{9}$$

Following Petersen (1983, p. 233) we introduce the usual notation in ergodic theory: The partitions $\alpha = \{A_1, \ldots, A_n\}$ and $\beta = \{B_1, \ldots, B_m\}$. We define $\alpha \vee \beta = \{A_i \cap B_j \mid i = 1, \ldots, n; j = 1, \ldots, m; A_i \cap B_j \neq \emptyset\}$. β is a refinement of α denoted as $\beta \geq \alpha$ if each B_j is up to a set of measure 0 a subset of some A_i.

The symbol stream generated by $\overline{fs}(V^{g^j})$ for $j = 0, \ldots, t-1$ corresponds to the set $\bigcap_{j=0}^{t-1} A_{i_j}^{g^j}$ which is an element of the partition $\alpha_0^{t-1} = \bigvee_{j=0}^{t-1} \alpha^{g^j}$ (the least common refinement of the partitions α, α^g, ..., $\alpha^{g^{t-1}}$).

We use in the following the standard notation for partitions used in ergodic theory with the tacit understanding that they get replaced by the corresponding symbol stream on which we compute the entropy.

According to Petersen (1983, p. 233), the block entropy in (4) is $H_k = \frac{1}{k} H(\alpha \vee \alpha^g \vee \ldots \vee \alpha^{g^{k-1}})$ and the entropy of the source of (5) is $h(\alpha, g) = \lim_{k\to\infty} \frac{1}{k} H(\alpha \vee \alpha^g \vee \ldots \vee \alpha^{g^{k-1}})$.

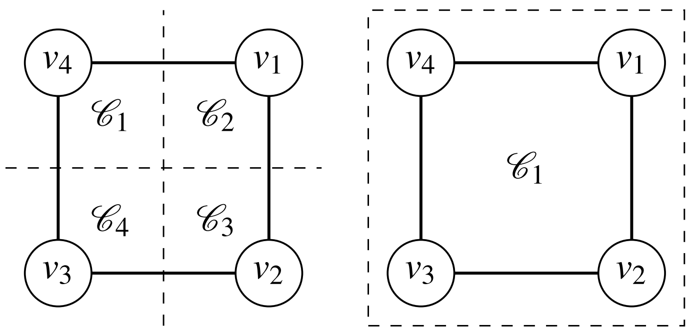
$h(\alpha, g)$ is a measure of the average uncertainty per unit of time we have about which element of the partition α a node will enter next under the action of the automorphism g. And, obviously, we are interested in the maximal uncertainty over all finite state processes (partitions) associated with g (the **Kolmogorov-Sinai entropy of** g):

$$h(g) = \sup_{\alpha} h(\alpha, g). \tag{10}$$

The size of $h(g)$ reflects the degree to which g disorganizes the space. $h(g)$ is an isomorphism invariant of g and it is a weak one.

4.3.4 Computation of the Kolmogorov-Sinai Entropy

In Rokhlin and Sinai (1961), Rokhlin proposed a denumerably infinite sequence of increasingly refined partitions tending to a partition of points for studying isomorphisms of stationary stochastic processes.



(a) Singleton partition $\mathscr{P}_{(1^n)}$ (b) Trivial Partition $\mathscr{P}_{(n^1)}$

Fig. 4 The two partitions are both special: The singleton partition results in the maximum possible entropy and the trivial partition has the minimum possible entropy of zero (not only for this but for *every* graph)

For the case of a finite permutation group, we get a finite chain of increasingly refined partitions, with two special partitions, namely the singleton (point) partition $\mathscr{P}_{(1^n)}$ and the trivial partition $\mathscr{P}_{(n^1)}$. For the trivial partition for any group action g, the finite state machine of the dynamical system produces a constant symbol stream. Therefore, $h(\mathscr{P}_{(n^1)}, g) = 0$. $\mathscr{P}_{(n^1)}$ is always the coarsest partition possible. We show examples of both special partitions in Fig. 4.

The entropy of $\mathscr{P}_{(1^n)}$ with $k = n$ singletons and a group action g which moves all nodes is $H(\mathscr{P}_{(1^n)}, g) = \frac{1}{n}\sum_{j=1}^{n}\left(-\sum_{i=1}^{k}\mathbb{P}(v_j \in \mathscr{C}_i)\log_2 \mathbb{P}(v_j \in \mathscr{C}_i)\right)$ and for a system with k states with equal probability, this reduces to $H(\mathscr{P}_{(1^n)}, g) = \log_2 k$. The symbol stream produced by the action v^{g^t} on $\mathscr{P}_{(1^n)}$ observed at some vertex in a cycle of a permutation is $\ldots\mathscr{C}_1\mathscr{C}_2\mathscr{C}_3\mathscr{C}_4\mathscr{C}_1\mathscr{C}_2\mathscr{C}_3\mathscr{C}_4\ldots$ and thus $H(\mathscr{P}_{(1^4)}, g) = \log_2 4 = 2$. The chain $\mathscr{P}_{(4^1)} \leq \mathscr{P}_{(2^2)} \leq \mathscr{P}_{(2\,1^2)} \leq \mathscr{P}_{(1^4)}$ for our example has the following sequence of entropies: $H(\mathscr{P}_{(4^1)}, g) = 0 < H(\mathscr{P}_{(2^2)}, g) = 1 < H(\mathscr{P}_{(2\,1^2)}, g) = 1.5 < H(\mathscr{P}_{(1^4)}, g) = 2$. Therefore, the Kolmogorov-Sinai entropy for the example shown in Fig. 3 is $h_{C_4}(g) = 2$. The (test) partition associated with the Kolmogorov-Sinai entropy is the coarsest partition at which the Kolmogorov Sinai entropy of the group action is maximal.

Until now, we have computed the Kolmogorov-Sinai entropy for a single action of the automorphism group $Aut(G)$. To extend the computation of the Kolmogorov-Sinai entropy to characterize the whole automorphism group $Aut(G)$ with maximal uncertainty we redefine the second finite state machine so that at time t an action of $Aut(G)$ is randomly selected (we use $rand(Aut(G))$ to denote this) and the finite state machine $\overline{fs}$ prints out a symbol $\mathscr{C}_i$ for each node $v \in V$ if node v is in partition $\mathscr{C}_i$ after t applications of $rand(Aut(G))$:

$$\overline{fs}(V^{rand(Aut(G))^t}) = \left(fs(v_1^{g^t}), \ldots, fs(v_n^{g^t})\right)^T, \quad g = rand(Aut(G)). \tag{11}$$

The definition of the Kolmogorov-Sinai entropy for a source which generates a symbol stream by the finite state machine (11) is then:

$$h(\alpha, Aut(G)) = \lim_{k\to\infty}\frac{1}{k}H(\alpha \vee \alpha^{rand(Aut(G))} \vee \ldots \vee \alpha^{rand(Aut(G))^{k-1}}) \tag{12}$$

and

$$h(Aut(G)) = \sup_{\alpha} h(\alpha, Aut(G)). \tag{13}$$

The Kolmogorov-Sinai entropy of the automorphism group of a graph is the average entropy of the symbol stream defined above. α_{KS} is any partition with $h(\alpha_{KS}, Aut(G)) = h(Aut(G))$. In practice, $h(Aut(G))$ is computed by taking the average of $h(g)$ for all $g \in Aut(G)$:

$$h(Aut(G)) = \frac{1}{|Aut(G)|} \sum_{g \in Aut(G)} h(g). \tag{14}$$

The computation of the Kolmogorov-Sinai entropy of $Aut(C_4)$ is shown in Table 2. Note, that $\alpha_{KS} = \mathscr{P}_{(1^4)}$ for the C_4 and $h(Aut(C_4)) = 1$.

$g_i \in Aut(C_4)$	$h(g)$
$g_1 =$ id	0
$g_2 =$ $(v_1\ v_2\ v_3\ v_4)$	2
$g_3 =$ $(v_1\ v_4\ v_3\ v_2)$	2
$g_4 =$ $(v_1\ v_3)$	0.5
$g_5 =$ $(v_2\ v_4)$	0.5
$g_6 =$ $(v_1\ v_4)(v_2\ v_3)$	1
$g_7 =$ $(v_3\ v_4)(v_1\ v_2)$	1
$g_8 =$ $(v_1\ v_3)(v_2\ v_4)$	1
$h(Aut(C_4))$	$\frac{8}{8} = 1$

Table 2 Computation of $h(Aut(C_4))$ for the partition α_{KS} which is the singleton partition shown in Fig. 4a. The left column denotes the permutations that form the automorphism group, the right column shows the entropy for each permutation. In the last row we show the entropy of the whole group which is calculated with Eq. 14

Of course, we are also interested in the entropy of a single node. By averaging over all group actions of $Aut(G)$ (or of one of its subgroups) instead of the nodes (Eq. 8) we get

$$H(\mathscr{P}, v, Aut(G)) = \frac{1}{|Aut(G)|} \sum_{g \in Aut(G)} H(\mathscr{P}, v, g). \tag{15}$$

If $H(\alpha_{KS}, v, Aut(G)) > 0$ then the node v is unstable.

Computing the Kolmogorov-Sinai entropy for a group action, a subgroup or a group as described until now is a formidable task because of the necessity to search the partition for which $h(Aut(G))$ (or $h(g)$) is the supremum over all partitions (see Eqs. 13 and 10).

Fortunately, $Aut(G)$ is a **finite** permutation group represented by a set of permutation functions in cycle form as in Table 2 and $Aut(G)$ is **known**. In this case we can compute the limits directly from the finite cycle structure of the group action: The cycles define how nodes are permuted. After a finite number of applications of a permutation function the original state is reached and all nodes in the cycle have been moved once. We call the set of nodes moved by a group action the orbit of the group action.

All nodes on the orbit of the group action are moved, all other nodes stay fixed (have an entropy of 0). All nodes in a cycle have the same entropy and the length of the cycle l determines the entropy: The probability that a node is in a certain position on the cycle is $\frac{1}{l}$ in the limit.

	Permutation of label Measures instability of the graph						Permutation of color Measures instability of the test partition			
$H(\mathscr{P},v,g_4)$	$lab(v^{g_4^3})$	$lab(v^{g_4^2})$	$lab(v^{g_4^1})$	$lab(v^{g_4^0})$	**Cell**		$col(v^{g_4^0})$	$col(v^{g_4^1})$	$col(v^{g_4^2})$	$col(v^{g_4^3})$
1.0	$(v_3$	$v_1)$	$(v_3$	$v_1)$	v_1	$\mathscr{C}_1$	$\mathscr{C}_1$	$\mathscr{C}_2$	$\mathscr{C}_1$	$\mathscr{C}_2$
0.0	v_2	v_2	v_2	v_2	v_2	$\mathscr{C}_2$	$\mathscr{C}_2$	$\mathscr{C}_2$	$\mathscr{C}_2$	$\mathscr{C}_2$
1.0	$(v_1$	$v_3)$	$(v_1$	$v_3)$	v_3	$\mathscr{C}_2$	$\mathscr{C}_2$	$\mathscr{C}_1$	$\mathscr{C}_2$	$\mathscr{C}_1$
0.0	v_4	v_4	v_4	v_4	v_4	$\mathscr{C}_1$	$\mathscr{C}_1$	$\mathscr{C}_1$	$\mathscr{C}_1$	$\mathscr{C}_1$
$h(g_4)=0.5$										
$H(\mathscr{P},v,g_6)$	$lab(v^{g_6^3})$	$lab(v^{g_6^2})$	$lab(v^{g_6^1})$	$lab(v^{g_6^0})$	**Cell**		$col(v^{g_6^0})$	$col(v^{g_6^1})$	$col(v^{g_6^2})$	$col(v^{g_6^3})$
1.0	$(v_4$	$v_1)$	$(v_4$	$v_1)$	v_1	$\mathscr{C}_1$	$\mathscr{C}_1$	$\mathscr{C}_1$	$\mathscr{C}_1$	$\mathscr{C}_1$
1.0	$(v_3$	$v_2)$	$(v_3$	$v_2)$	v_2	$\mathscr{C}_2$	$\mathscr{C}_2$	$\mathscr{C}_2$	$\mathscr{C}_2$	$\mathscr{C}_2$
1.0	$(v_2$	$v_3)$	$(v_2$	$v_3)$	v_3	$\mathscr{C}_2$	$\mathscr{C}_2$	$\mathscr{C}_2$	$\mathscr{C}_2$	$\mathscr{C}_2$
1.0	$(v_1$	$v_4)$	$(v_1$	$v_4)$	v_4	$\mathscr{C}_1$	$\mathscr{C}_1$	$\mathscr{C}_1$	$\mathscr{C}_1$	$\mathscr{C}_1$
$h(g_6)=1.0$										

Table 3 Applying group actions g_4 and g_6 to the labels (nodes) and to the colors (clusters) of the C_4 in state 0 in Fig. 3. Start reading from the center **Cell** column. To the right, the symbol stream of clusters/colors is shown. To the left, the symbol stream of nodes/labels is shown.

Table 3 shows the symbols streams generated by g_4 and g_6 and illustrates how $h(g_4)$ and $h(g_6)$ in Table 2 are computed. The table also illustrates that we have defined two types of measures:

1. $h(g)$ and $h(Aut(G))$ (the Kolmogorov-Sinai entropy of a group action on a graph and of the automorphism group of a graph) are invariant measures of the instability of the graph. Entropies above zero indicate potential instabilities and multiple equivalent solutions of cluster algorithms.
2. $h(\alpha,g)$ and $h(\alpha,Aut(G))$ are the Kolmogorov-Sinai entropy of a graph partition with regard to a group action and with regard to the automorphism group of a graph. Table 3 shows that the partition shown in state 0 in Fig. 3 detects the instability caused by g_4 but not the instability caused by g_6. These measures can be used to diagnose the stability of the best solutions found by cluster algorithms.

4.4 Applications to Graph Clustering

We conclude with two examples of a straightforward application of the ergodic constructions used in the computation of the Kolmogorov-Sinai entropy to graph cluster analysis. Given a graph G, its automorphism group $Aut(G)$, and P_{OPT} the optimal partition found by a graph clustering algorithm:

1. If $h(Aut(G)) = 0$, then no structurally equivalent solutions with the same value of the cluster criterion exist. However, congruent solutions may exist.
2. If $h(P_{OPT}, Aut(G)) = 0$, then P_{OPT} is compatible with $G^{Aut(G)}$. This means that with regard to $Aut(G)$ the optimal solution found by the cluster algorithm is unique: No other structurally equivalent solution with the same value of the cluster criterion exists.
3. If $h(P_{OPT}, Aut(G)) > 0$, then some of the group actions in $Aut(G)$ move nodes between clusters of P_{OPT}, the solution found is unstable and multiple structurally equivalent solutions (which can be enumerated with the help of $Aut(G)$) exist.
4. $h(P_{OPT}, Aut(G))$ is an invariant measure of the instability of G with regard to the partition P_{OPT}.

For Zachary's karate network K (Zachary, 1984, Fig. 5), the modularity-optimal partition (with $Q(K, P_{OPT}) = 0.4198$, Ovelgönne et al, 2010) is

$$\begin{aligned}P_{OPT} = \{&\{10,4,5,6,16\},\{17,21,12,11,1,3,0,7,2,19,13\},\\ &\{29,33,30,8,26,32,9,14,15,22,18,20\},\{25,24,23,27,31,28\}\}\end{aligned}$$

and $Aut(K)$ has three non-trivial subgroups, namely $G_1 = Sym(\Omega_1)$ with $\Omega_1 = \{14,15,18,20,22\}$, $G_2 = Sym(\Omega_2)$ with $\Omega_2 = \{17,21\}$, and the subgroup $G_3 = \{(),(4\ 10)(5\ 6)\}$. All three non-trivial subgroups are contained in the clusters of P_{OPT}, in detail $\Omega_1 \subset C_3$, $\Omega_2 \subset C_2$ and G_3 acts on nodes in C_1. The finite state machine (11) produces a constant symbol stream for P_{OPT} and $h(P_{OPT}, Aut(K)) = 0$. Nonetheless, $h(Aut(K)) \approx 0.29$. The node entropy for any node $u \in \Omega_1$ is $H(\alpha_{KS}, u, Aut(K)) \approx 1.38$ and the node entropy for any node $w \in \{4,5,6,10,17,21\}$ is $H(\alpha_{KS}, w, Aut(K)) = 0.5$. For all other (stable!) nodes v, $H(\alpha_{KS}, v, Aut(K)) = 0$.

Finally we present some useful test partitions for a graph and provide examples for the karate network in Table 4

1. The finest partition β with entropy zero ($\nexists \alpha \geq \beta : h(\beta, Aut(G)) = 0 \wedge h(\alpha, Aut(G)) = 0$): Any coarser partition than β also has entropy zero, there-

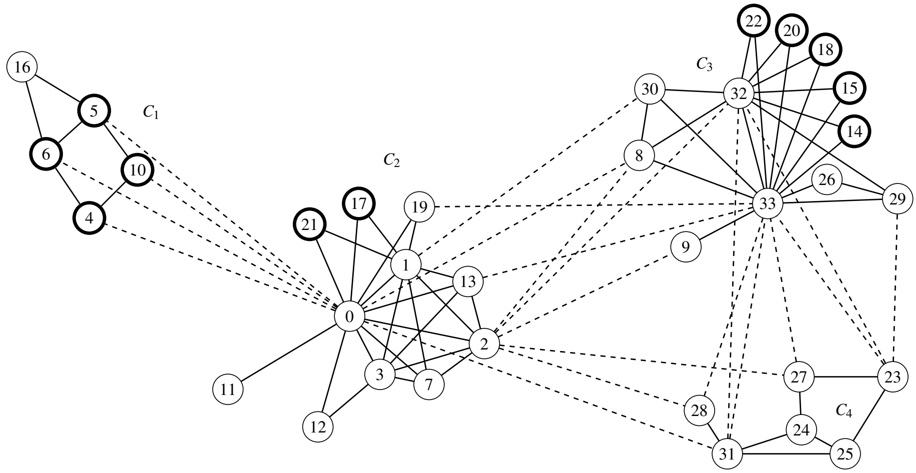


Fig. 5 The karate network which represents relationships between members of a karate club that Zachary (1977) analyzed. The four clusters of the modularity-optimal solution are connected by the dashed edges, nodes moved by automorphisms a drawn with thick border

fore if P_{OPT} is a coarsening of β it is a stable solution. The partition β_K shown in Table 4 is a refinement of the optimal graph clustering partition ($P_{OPT} \leq \beta_K$) which proves that P_{OPT} is stable.

2. The coarsest partition α which has maximum entropy ($h(\alpha, Aut(G)) = h(Aut(G))$): All finer partitions have also maximum entropy. This partition may not be unique. The partition α_K in Table 4 is not invariant with regard to $Aut(K)$.
3. A special coarsest partition $\gamma \geq \alpha$ that *is* invariant and therefore unique. The difference between α_K and γ_K in Table 4 is that in α_K nodes that are moved by different subgroups are in the same cluster.

Partition	Value for the karate network
β_K	$\{\{0\},\{1\},\{2\},\{3\},\{4,10\},\{5,6\},\{7\},\{8\},\{9\},\{11\},\{12\},\{13\},\{14,15,18,20,22\},$ $\{17,21\},\{16\},\{19\},\{23\},\{24\},\{25\},\{26\},\{27\},\{28\},\{29\},\{30\},\{31\},\{32\},\{33\}\}$
α_K	$\{\{16,0,1,2,3,7,11,12,13,19,23,24,25,27,28,31,8,9,26,29,30,32,33,14\},$ $\{6,4,15,17\},\{5,10,18,21\},\{20\},\{22\}\}$
γ_K	$\{\{16,0,1,2,3,7,11,12,13,19,23,24,25,27,28,31,8,9,26,29,30,32,33\},\{6,4\},\{5,10\},$ $\{17\},\{21\},\{14\},\{15\},\{18\},\{20\},\{22\}\}$

Table 4 Three test partitions for the karate network K that have different properties: β_K is the finest possible partition with zero entropy, α_K is an example of a coarsest possible partition with maximum entropy and γ_K also has maximum entropy but is invariant with regard to $Aut(K)$

5 Conclusion and Further Research

In this paper we have introduced three weak invariants of actions of the automorphism group of a graph, namely the modularity measure, the graph partition type and the Kolmogorov-Sinai entropy for sets of actions of $Aut(G)$. For finite permutation groups, we have presented a constructive method of computing the Kolmogorov-Sinai entropy from the group actions of a permutation group.

We have given a formal definition of the stability of graph cluster solutions with regard to the automorphism group of the graph. This definition of stability requires no further assumptions but relies only on the inherent symmetries of the graph.

In addition, we have defined two types of invariant Kolmogorov-Sinai entropy measures: one for the instability of a graph and the second for the instability of a partition. These measures address the problems of MSE solutions for graph clustering in several ways: $h(Aut(G)) > 0$ implies that MSE solutions may exist. However, MSE solutions for P_{OPT} exist, if in addition $h(P_{OPT}, Aut(G)) > 0$. The second type of measures is based on the idea of test partitions underlying the general construction the Kolmogorov-Sinai entropy for ergodic dynamic systems. These measures are used to test if the optimal partition found by a clustering algorithm is unique or if multiple structurally equivalent partitions exist. At the moment, this requires the computation of the optimal partition by a graph clustering algorithm and the computation of the automorphism group of the graph.

An analysis of the node entropies shows where the graph is unstable and can be used to visualize the extent of the instability of the graph.

What remains to be done is to directly integrate stability diagnostics with graph clustering algorithms. We see modularity and the partition type (as byproducts of (hierarchical) graph clustering algorithms) as properties useful in finding candidate partitions from which graph permutation functions can be identified efficiently. We plan to use the Kolmogorov-Sinai entropy for analyzing a sample of dendrograms from efficient, randomized ensemble learning modularity clustering algorithms (see Ovelgönne and Geyer-Schulz, 2013).

Acknowledgements We thank the anonymous reviewers for their helpful comments which led us to a major revision of the section on the Kolmogorov-Sinai entropy.

References

Birkhoff GD (1931) Proof of the ergodic theorem. Proceedings of the National Academy of Sciences of the United States of America 17:656 – 660, DOI 10.1073/pnas.17.2.656

Brandes U, Delling D, Gaertler M, Görke R, Hoefer M, Nikoloski Z, Wagner D (2008) On modularity clustering. IEEE Transactions on Knowledge and Data Engineering 20(2):172–188, DOI 10.1109/TKDE.2007.190689

Diaconis P (1985) Theories of data analysis. In: Hoaglin DC, Mosteller F, Tukey JW (eds) Exploring Data Tables, Trends, and Shapes, Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, New York, chap 1, pp 1 – 36

Doob JL (1994) Measure Theory. Graduate Texts in Mathematics, Springer, New York, DOI 10.1007/978-1-4612-0877-8

Fortunato S (2010) Community detection in graphs. Physics Reports 486(3-5):75 – 174, DOI 10.1016/j.physrep.2009.11.002

Fortunato S, Barthélemy M (2007) Resolution limit in community detection. Proceedings of the National Academy of Sciences of the United States of America 104(1):36 – 41, DOI 10.1073/pnas.0605965104

Hennig C (2015) What are the true clusters? Pattern Recognition Letters 64:53 – 62, DOI 10.1016/j.patrec.2015.04.009

James GD (1978) The Representation Theory of the Symmetric Groups, Lecture Notes in Mathematics, vol 682. Springer, Berlin, DOI 10.1007/BFb0067708

McKay BD (1981) Practical graph isomorphism. Congressus Numerantium 30:45 – 87

McKay BD, Piperno A (2014) Practical graph isomorphism, {II}. Journal of Symbolic Computation 60:94 – 112, DOI 10.1016/j.jsc.2013.09.003

Meilă M (2003) Comparing clusterings by the variation of information. In: Schölkopf B, Warmuth MK (eds) Learning Theory and Kernel Machines, Lecture Notes in Computer Science, vol 2777, Springer Berlin / Heidelberg, pp 173 – 187, DOI 10.1007/978-3-540-45167-9_14

Newman MEJ, Girvan M (2004) Finding and evaluating community structure in networks. Physical Review E 69(2):026,113, DOI 10.1103/PhysRevE.69.026113

Ovelgönne M, Geyer-Schulz A (2013) An ensemble learning strategy for graph clustering. In: Bader DA, Meyerhenke H, Sanders P, Wagner D (eds)

Graph Partitioning and Graph Clustering, American Mathematical Society, Providence, Contemporary Mathematics, vol 588, pp 187 – 205, DOI 10.1090/conm/588/11701

Ovelgönne M, Geyer-Schulz A, Stein M (2010) Randomized greedy modularity optimization for group detection in huge social networks. In: SNA-KDD'10: Proceedings of the 4th Workshop on Social Network Mining and Analysis, ACM, New York, NY, USA, pp 1 – 9

Petersen K (1983) Ergodic Theory, Cambridge Studies in Advanced Mathematics, vol 2. Cambridge University Press, Cambridge

Rokhlin VA, Sinai YG (1961) Construction and properties of invariant measurable partitions. Doklady Akademiia Nauk SSSR 141:1038 – 1041

Rudolph DJ (1990) Fundamentals of Measurable Dynamics. Ergodic Theory on Lebesgue Spaces. Oxford University Press, Oxford

Shannon CE (1948) A mathematical theory of communication. The Bell System Technical Journal 27:379 – 423

Sinai YG (1959) The notion of entropy of a dynamical system. Doklady Akademiia Nauk SSSR 125(5):768 – 771

Sinai YG (2010) On the notion of entropy of a dynamical system. In: Selecta: Ergodic Theory and Dynamical Systems, vol 1, Springer-Verlag, Dordrecht, chap 1, pp 3 – 14

Weisfeiler B (1976) On Construction and Identification of Graphs, Lecture Notes in Mathematics, vol 558. Springer, Berlin, DOI 10.1007/BFb0089374

Wielandt H (1964) Finite Permutation Groups. Academic Press, New York

Zachary WW (1977) An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33(4):452 – 473

Zachary WW (1984) Modeling social network processes using constrained flow representations. Social Networks 6(3):259–292, DOI 10.1016/0378-8733(84)90013-3