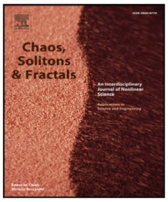
Chaos, Solitons and Fractals 177 (2023) 114200

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

journal homepage: www.elsevier.com/locate/chaos



Detecting communities in higher-order networks by using their derivative graphs

Gonzalo Contreras-Aso [a,c,*], Regino Criado [a,b,c], Guillermo Vera de Salas [a,c], Jinling Yang [a,d]

[a] *Departamento de Matemática Aplicada, Ciencia e Ingeniería de los Materiales y Tecnología Electrónica, Universidad Rey Juan Carlos, C/ Tulipán s/n, Móstoles, 28933 Madrid, Spain*
[b] *Data, Complex Networks and Cybersecurity Sciences Technological Institute, Universidad Rey Juan Carlos, Plaza Manuel Becerra 14, 28028 Madrid, Spain*
[c] *Laboratory of Mathematical Computation on Complex Networks and their Applications, Universidad Rey Juan Carlos, C/ Tulipán s/n, Móstoles, 28933 Madrid, Spain*
[d] *Unmanned Systems Research Institute, Northwestern Polytechnical University, Xi'an 710072, Shaanxi, PR China*

ARTICLE INFO

Dataset link: https://github.com/LaComarca-Lab/HyperGraph-Communities

Keywords:
Hypergraph
Derivative of a hypergraph
Higher-order network
Community
Communities in a hypergraph
UPGMA
Hierarchical clustering

ABSTRACT

Similar to what happens in the pairwise network domain, the communities of nodes of a hypergraph (also called higher-order network) are formed by groups of nodes that share many hyperedges, so that the number of hyperedges they share with the rest of the nodes is significantly smaller, and therefore these communities can be considered as independent compartments (or super-clusters) of the hypergraph. In this work we present a method, based on the so-called derivative graph of a hypergraph, which allows the detection of communities of a higher-order network without high computational cost and several simulations are presented that show the significant computational advantages of the proposed method over other existing methods.

1. Introduction

In the last thirty years, Network Science has grown and developed in such a way that it has become one of the hottest and most successful research fields, with interdisciplinary applications in areas as different as genetics and neuroscience, systems biology, artificial intelligence, meteorology or cybersecurity [1–8]. Complex network models have become indispensable elements for the representation and simulation of the different types of interactions and relationships between the different parts of a system, being applied in many fields such as engineering, linguistics, social networks or economics [6,9–16]. However, there are many contexts and situations in which it is not possible to represent the relationships between the different components of a system in terms of pairwise interactions, so that to obtain an adequate model of the system it is necessary to consider interactions of an order higher than two [17–24]. The emergence of new structures and models with multiple applications have made it possible to represent different types of interactions between the constituent elements of a complex system in a very efficient way. Thus, by extending the concept of interaction between two nodes of a network to an interaction of more than two nodes, the concept of hypergraph or higher-order network appears naturally [17,18,25,26]. It is noteworthy that, as in the case of networks with pairwise connections, in real higher-order networks the connections between nodes are very heterogeneous, with some nodes having multiple connections, while others have a much lower degree of interaction, resulting in groups of nodes with a high concentration of interactions between them and a limited number of interactions with nodes in other groups. Thus, a more or less clear division of the nodes into different groups appears, revealing a community structure that allows the higher-order network to be analyzed at the mesoscale level. At this level, it is possible to study the higher-order network from a new graph in which the vertices are the communities and the edges represent an appropriate and specific size of connections or interactions between them. The underlying idea is that each community clusters nodes that share an important number of properties and that probably play a similar role in the functioning of the network.

It is worth noting that the problem of identifying clusters and modules within a network from its topology has a long tradition. This problem has been studied in relation to phenomena and in the field of very different disciplines (in the context of combinatorial graph theory

* Correspondence to: Universidad Rey Juan Carlos, Calle Tulipán s/n, Móstoles, 28933 Madrid, Spain.
E-mail addresses: gonzalo.contreras@urjc.es (G. Contreras-Aso), regino.criado@urjc.es (R. Criado), guillermo.vera@urjc.es (G. Vera de Salas), jinling.yang@urjc.es (J. Yang).

https://doi.org/10.1016/j.chaos.2023.114200
Received 20 July 2023; Accepted 24 October 2023
Available online 2 November 2023

this problem is known as "graph partitioning"). Thus some approaches to this problem are based on optimization, statistical inference, random walkers, building a hierarchical dendrogram or even obtaining the communities from a local seed by adding nodes until a local optimal result with respect to some quality function is obtained [27–29]. In any case the community structure in networks with pairwise interactions has been extensively studied in the literature [27–37]. In fact, algorithmic methods for community detection have been developed hand in hand with the many disciplines in which this tool has applications [2,4,8,11,27,29,33,34]. On the other hand, within the study and identification of groups of nodes that interact closely and probably play a similar role in the considered structure, the detection of communities in the context of higher-order networks has also received some attention from the network science community [38–41]. In this paper we present a new method for community detection in higher-order networks based on the concept of derivative graph associated to a hypergraph [10,42], which, in addition to being naturally adapted to higher-order networks, presents certain computational advantages over other algorithms used in this context.

The structure of this paper is as follows. After this introduction, Section 2 is devoted to develop some ideas about the concept of derivative of a hypergraph and to establish the fundamentals for a new method to detect communities in a hypergraph through its derivative graph. In Section 3 we apply the mathematical concepts and the structures defined in the previous sections to obtain a new algorithm to detect communities existing in a higher order network. Section 4 is devoted to applying the instruments and tools developed to obtain the corresponding characteristic communities to several practical examples, both synthetic and from the real world. Finally in Section 5 we present some conclusions of this work.

2. Basic concepts and some preliminary results

A graph (or network) is a pair of sets $G = (X, E)$ in which $X = \{1, \dots, N\}$ is a finite set of nodes and $E = \{e_1, \dots, e_m\}$ is a set of edges (or links between certain pairs of nodes). In the following, we will denote by $e_{ij} \in E$ the edge between nodes i and j, although sometimes we will also denote the edge e_{ij} by $\{i, j\}$. Finally, a weighted graph is a graph in which each edge e_{ij} has associated with it a numerical value $w(e_{ij}) = w_{ij}$ called its weight.

In the same way, following [20], a hypergraph is a pair of sets $\mathcal{H} = (X, \varepsilon)$ in which $X = \{1, \dots, N\}$ is a finite set of nodes and $\varepsilon = \{h_1, h_2, \dots, h_n\}$ is a collection of subsets of X such that $h_i \neq \emptyset \quad (i = 1, 2, \dots, n)$ and $X = \bigcup_{i=1}^{n} h_i$. The elements of ε are called hyperedges. Thus, hypergraphs appeared as the natural extensions of graphs to describe group interactions between sets of nodes.

Rather than working directly with these sets, it is common to resort to some matricial representation of the graph or hypergraph. In hypergraph contexts, the incidence matrix $I(\mathcal{H}) \equiv (I_{ih}) \in \mathbb{R}^{N \times |e|}$ is usually defined as

$$(I_{ih}) = \begin{cases} 1 & \text{if} \quad i \in h, \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

It is not difficult to check that

$$I(\mathcal{H})^t \cdot I(\mathcal{H}) = \widehat{A(\mathcal{H})} = (\widehat{a_{ij}}) \in \mathbb{R}^{|\varepsilon| \times |\varepsilon|}$$

and

$$I(\mathcal{H}) \cdot I(\mathcal{H})^t = A(\mathcal{H}) = (a_{ij}) \in \mathbb{R}^{N \times N},$$

where

$$\widehat{a_{ij}} = \begin{cases} |h_i| & \text{if} \quad i = j, \\ |h_i \cap h_j| & \text{if} \quad i \neq j, \end{cases}$$

and

$$a_{ij} = \begin{cases} |\{h \in \varepsilon \mid i \in h\}| & \text{if} \quad i = j, \\ |\{h \in \varepsilon \mid i, j \in h\}| & \text{if} \quad i \neq j. \end{cases} \tag{2.2}$$

The matrix $A(\mathcal{H}) = (a_{ij}) \in \mathbb{R}^{N \times N}$, hereinafter denoted by $A = (a_{ij})$, is called frequency matrix of $\mathcal{H}$. The concept of derivative graph makes it possible to quantify the degree of dissimilarity between the nodes of a hypergraph. In fact, it can be said that, since the introduction of Jaccard's index [43], through different adaptations and generalizations of this concept, it can be said that quantifying the similarity between models and structures is one of the most important aspects that has contributed to the development of theories and models in science and technology [43–46].

In the following we will consider the methodology introduced in [10,42] in order to analyze and quantify the similarity between two nodes i, j of a hypergraph. The concept of derivative of a hypergraph with respect to two nodes allows us to quantify the heterogeneity and similarity between two nodes of the considered hypergraph. Thus, if $\mathcal{H} = (X, \varepsilon)$ is a hypergraph whose associated frequency matrix is $A = (a_{ij})$, the derivative of $\mathcal{H}$ with respect to the pair of nodes $i, j \in X$ is the numerical value $\frac{\partial \mathcal{H}}{\partial \{i,j\}}$ obtained by applying the following formula:

$$\frac{\partial \mathcal{H}}{\partial \{i, j\}} = \frac{a_i - a_{ij} + a_j - a_{ij}}{a_{ij}} = \frac{a_i - 2a_{ij} + a_j}{a_{ij}}. \tag{2.3}$$

The numerical value of $\frac{\partial \mathcal{H}}{\partial \{i,j\}}$ is called degree of independence of i and j with respect to $\mathcal{H}$. Obviously, if there is not at least one hyperedge $h \in \varepsilon$ such that $i, j \in h$, it happens that $\frac{\partial \mathcal{H}}{\partial \{i,j\}} = \infty$, and if $\forall h \in \varepsilon \quad (i \in h \Leftrightarrow j \in h)$ then we will have $\frac{\partial \mathcal{H}}{\partial \{i,j\}} = 0$.

Note that $\forall i, j \in X$ we have that $\frac{\partial \mathcal{H}}{\partial \{i,j\}} \geq 0$.

In general, it is possible (and natural) to consider each hyperedge $h \in \varepsilon$ as a property that a node may or may not have, or even as an event or a process in which a particular node may or may not participate. Thus, the value of $\frac{\partial \mathcal{H}}{\partial \{i,j\}}$ characterizes the (relative) heterogeneity of the properties simultaneously satisfied by the nodes i and j represented by the hyperedges of ε to which such nodes belong. On the other hand, the smaller the value of the derivative with respect to the pair of nodes i, j, the greater the identification and similarity between these corresponding nodes i, j with respect to the considered set of properties (in fact, if $\frac{\partial \mathcal{H}}{\partial \{i,j\}} = 0$, these nodes turn out to be, from the point of view of the structure of $\mathcal{H}$, indistinguishable). In other words, the higher the value of the derivative, the greater the number of hyperedges (or properties) that these nodes do not share. Therefore, it makes sense to consider the following definition [10]:

Definition 2.4. If $\mathcal{H} = (X, \varepsilon)$ is a hypergraph, the weighted graph obtained by considering the derivative of $\mathcal{H}$ with respect all the pairs of nodes $i, j \in X$, and by setting $\forall i, j \in X$ the corresponding numerical value of $\frac{\partial \mathcal{H}}{\partial \{i,j\}}$ on the edge $\{i, j\}$ is called the derivative graph $\partial \mathcal{H}$ of $\mathcal{H}$, in such a way that if $\frac{\partial \mathcal{H}}{\partial \{i,j\}} = 0$, then the nodes i and j collapse into a single node (ij), and having in mind that if $\frac{\partial \mathcal{H}}{\partial \{i,j\}} = \infty$, then the edge $\{i, j\}$ does not exist in the derivative graph.

Comprehensively, it can be said that the derivative graph $\partial \mathcal{H}$ gives us a representation of the level of heterogeneity of participation of the nodes over the different hyperedges of $\mathcal{H}$.

Let us showcase this idea through the following example:

Example 2.5. Consider the hypergraph $\mathcal{H} = (X, \varepsilon)$ from Fig. 1, where $X = \{1, 2, 3, 4, 5, 6\}$, $\varepsilon = \{h_1, h_2, h_3, h_4\}$, and $h_1 = \{1, 2, 6\}$, $h_2 = \{2, 4\}$, $h_3 = \{3, 4, 5\}$, $h_4 = \{3, 5\}$. We have that

$$I(\mathcal{H})^t = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix},$$

$$I(\mathcal{H}) \cdot I(\mathcal{H})^t = A(\mathcal{H}) = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 2 & 0 & 1 & 0 & 1 \\ 0 & 0 & 2 & 1 & 2 & 0 \\ 0 & 1 & 1 & 2 & 1 & 0 \\ 0 & 0 & 2 & 1 & 2 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

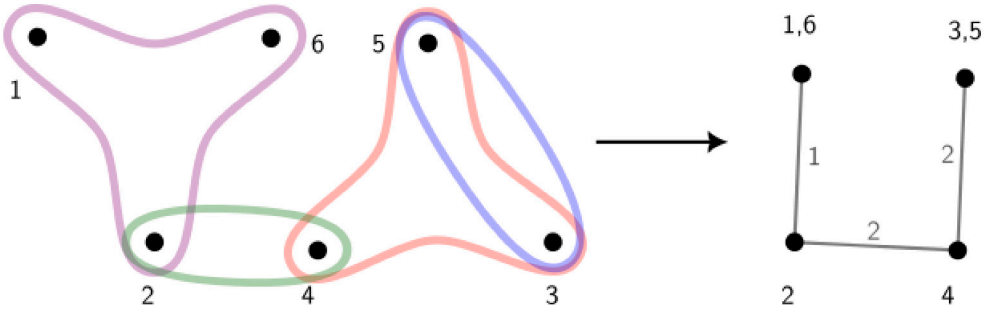


Fig. 1. Hypergraph from Example 2.5 and its corresponding derivative graph, where "identical" nodes (as seen by the derivative operation) have been collapsed together.

The values of the derivatives of $\mathcal{H}$ with respect to all the pair of nodes of G are, respectively:

$$\frac{\partial\mathcal{H}}{\partial\{1,2\}}=1,\ \frac{\partial\mathcal{H}}{\partial\{1,3\}}=+\infty,\ \frac{\partial\mathcal{H}}{\partial\{1,4\}}=+\infty,\ \frac{\partial\mathcal{H}}{\partial\{1,5\}}=+\infty,\ \frac{\partial\mathcal{H}}{\partial\{1,6\}}=0,$$
$$\frac{\partial\mathcal{H}}{\partial\{2,3\}}=+\infty,\ \frac{\partial\mathcal{H}}{\partial\{2,4\}}=2,\ \frac{\partial\mathcal{H}}{\partial\{2,5\}}=+\infty,\ \frac{\partial\mathcal{H}}{\partial\{2,6\}}=1,\ \frac{\partial\mathcal{H}}{\partial\{3,4\}}=2,$$
$$\frac{\partial\mathcal{H}}{\partial\{3,5\}}=0,\ \frac{\partial\mathcal{H}}{\partial\{3,6\}}=+\infty,\ \frac{\partial\mathcal{H}}{\partial\{4,5\}}=2,\ \frac{\partial\mathcal{H}}{\partial\{4,6\}}=+\infty,\ \frac{\partial\mathcal{H}}{\partial\{5,6\}}=+\infty.$$

Note that nodes 1, 6 have collapsed into a single node in the derivative network, and likewise nodes 3, 5.

As we stated before, the derivative graph implements a measure of similarity between nodes. It is therefore natural to ask if we can leverage the information provided by this structure to partition the nodes of the hypergraph into distinct classes, separated by this similarity: this will constitute the basis of the two community detection methods that we are going to discuss next.

3. Different approaches to detect communities in hypergraphs

Now that we have established some useful concepts and notation, we will start describing the two algorithms we are putting forward in order to obtain hypergraph communities. Both of them have a hierarchical clustering background, although a different approach to the partitioning itself: the first is essentially a data-driven algorithm, while the second one does take into account the hypergraph nature of the problem. In the following, we will also briefly describe an alternative method of community detection in hypergraphs presented in [47], which we will use to compare our results.

3.1. Community detection through unsupervised clustering

Within hierarchical clustering, there are two procedures: agglomerative and divisive. Agglomerative clustering, which is the focus of this text, merges the pair of closest clusters in each step until there is one final node left, which comprises the entire dataset.

The agglomerative hierarchical clustering method is particularly useful for partitioning datasets for which merely two pairwise distance functions are defined. One distance function to measure distances between nodes and a second to measure distances between clusters depending on the distance between points, traditionally called linkage function. There are several functions that can be found in the literature (single link, average link or UPGMA, Ward, ...) [48–50]. In our study we will focus on the average link function, for reasons we will discuss in the next paragraphs.

Usually, the hierarchical clustering method works on a set of data that can be seen as $\mathbb{R}^n$ points. Because of this, a classical choice for modeling the distance between points in the dataset is to use the Euclidean distance. The successive steps in which the nodes are clustered can be represented using a diagram, called a dendrogram. Specifically, on one of the axes the points of the dataset will be represented and on a perpendicular axis the height, i.e. the distance between clusters.

We propose to use the derivative between nodes as a semi-distance between points of the dataset. More precisely, let $\mathcal{H}=(X,\varepsilon)$ be a hypergraph and let ∂X be the set of vertices of the derivative graph. Let us consider the function $d:\partial X\times\partial X\to\mathbb{R}$ defined as

$$d(i,j):=\frac{\partial\mathcal{H}}{\partial\{i,j\}}. \tag{3.1}$$

Then d is a semidistance, i.e., for every $i,j\in\partial X$,

- it is symmetric, $d(i,j)=d(j,i)$,
- it is positive, $d(i,j)\geq 0$ and $d(i,j)=0$ if and only if $i=j$,

The derivatives do not define a distance on ∂X since the triangle inequality does not hold as can be seen in the following example.

Example 3.2. Let us consider the following hypergraph with 4 nodes and 5 hyperedges

$$H=(X,\varepsilon),\quad X=\{a,b,c,d\},\quad \varepsilon=\{\{a,b\},\{a,b,d\},\{b,c\},\{a,c\},\{a,c,d\}\}.$$

In this example we have

$$\frac{\partial\mathcal{H}}{\partial\{a,b\}}=\frac{3}{2},\quad \frac{\partial\mathcal{H}}{\partial\{a,c\}}=4,\quad \frac{\partial\mathcal{H}}{\partial\{b,c\}}=\frac{3}{2},$$

and as we have anticipated, we find

$$4=\frac{\partial\mathcal{H}}{\partial\{a,c\}}\nleq\frac{\partial\mathcal{H}}{\partial\{a,b\}}+\frac{\partial\mathcal{H}}{\partial\{b,c\}}=3$$

Since d is just defined on a discrete set, it does not make sense to consider linkage functions that are using auxiliary elements, such as centroids, to compute the distance between clusters. Thus, the appropriate selection for the linkage function in the case at hand will be the average (UPGMA).

To initiate the agglomerative hierarchical clustering algorithm, each initial cluster will be a singleton containing a node. In each step it will be computed the distance between clusters using the average linkage, i.e.,

$$D(C_i,C_j)=\frac{1}{|C_i\|C_j|}\sum_{x\in C_i,y\in C_j}d(x,y) \tag{3.2}$$

Hence, we will establish the communities through the dendrogram.

It is worth noting that the selected approach gives us the flexibility to select different relationships as to represent the distances between nodes. For example, it is possible to consider the distance defined as $d(i,j):=1-\mathcal{J}(\varepsilon(i),\varepsilon(j))$ where $\mathcal{J}$ is the Jaccard index and $\varepsilon(i)=\{h\in\varepsilon\ |i\in h\}$. However, in this specific case, after extensive evaluation and computation, the results obtained for this distance were worse for the datasets considered in this work.

3.2. A general criterion to cut the dendrogram: highest gap cut

Before pointing out a criterion to cut the dendrogram, and as pointed out in [51] in agglomerative hierarchical clustering, there is no uniqueness for pair-group methods when two or more distances between different clusters coincide during the clustering process. The usual approach to solve this drawback is to take any arbitrary criterion to break the ties between distances, which results in different hierarchical rankings depending on the criterion followed. Although it would be possible to consider other criteria, such as grouping more than two clusters at the same time when ties occur (as proposed in [51]) it is computationally more efficient to use, in Algorithm 1 considered, the sorting criterion given by an internal variable in the case of coincidence of two or more distances between different clusters.

In any case, in a broad context, we can apply a criterion to select a specific partition of the dendrogram. Although this approach does not use additional information from the hypergraph, its results may be sufficiently accurate to be borne in mind. Despite its simplicity, the criterion is intuitive and worth explaining.

The agglomerative hierarchical clustering, in each step, will merge two different clusters minimizing the distance between clusters, which will hereinafter be denoted by D. For instance, in the first step it looks for the minimum value of

$$D(C_{0,i}, C_{0,j}) := D^0_{ij} \tag{3.3}$$

for $i, j \in \{1, \ldots, |\partial X|\}$ and merge those two cluster related to the minimum value. In general, after k steps, we will end up with $|\partial X| - k$ clusters. Recall that in the dendrogram, the height is coinciding with the distance values. Therefore when two clusters are merging it means that the distance between those two are the height.

Notice that in the dendrogram, the height between branches arises from the $(k-1)$-th and kth steps can be expressed as

$$\tau_k = \lambda_k - \lambda_{k-1} \geqslant 0 \tag{3.4}$$

where $\lambda_k = \min_{i,j \in \{1,\ldots,|\partial X|-k\}} D(C_{k,i}, C_{k,j}) = \min_{i,j} D^k_{ij}$, and $C_{k,i}$, $C_{k,j}$ are two clusters from the kth step. Bigger values of τ means that the distances between clusters are bigger, i.e., it can be interpreted as to make the next clustering is more expensive. As a unified criterion we suggest to stop the algorithm for the step n where

$$\tau_{\max} = \max_{k=1,\ldots,|\partial X|} \tau_k \tag{3.5}$$

Therefore the dendrogram should be cut at height

$$h_{\text{cut}} = \lambda_{k-1} + \frac{\tau_{\max}}{2}, \quad \text{with} \quad k \quad \text{such that} \quad \tau_k = \tau_{\max} \tag{3.6}$$

This choice seems reasonable enough since it coincides when we have the longest height between steps. Nevertheless, if all these gaps are similar enough, i.e., for $\varepsilon > 0$ small $|\tau_i - \tau_j| < \varepsilon$, this criterion may not be useful.

Algorithm 1: Community detection at largest dendrogram gap

```
Data: Hypergraph $\mathcal{H}$, linkage method
Result: node partition
/* Compute the derivative graph */
derivatives ← $\mathcal{H}$;
/* Compute the linkage function */
$Z$ ← linkage(derivatives, linkage method);
/* Find the largest gap, and its middle height */
$h_{\text{cut}}$ ← largest_gap_cut($Z$);
/* Find partition associated to said height */
node partition ← cut_tree($Z$, $h_{\text{cut}}$);
```

3.3. A criterion based on modularity to cut the dendrogram

Modularity is a concept used in traditional network analysis to quantify the presence of community structure or modular organization within a network. A modular network is characterized by the division of nodes into distinct groups or communities, where nodes within a community are more densely connected to each other than to nodes in other communities.

The modularity of a pairwise network measures the strength of this community structure. It is a scalar value that ranges from −1 to 1, with higher values indicating a stronger modular structure. A modularity value close to 1 suggests a clear separation of communities, while values close to 0 or negative values indicate a more random or poorly defined community structure. It is defined as

$$Q = \frac{1}{2m} \sum_{ij} \left[a_{ij} - \frac{k_i k_j}{2m} \right] \delta(C_i, C_j), \tag{3.7}$$

where $A = (a_{ij})$ is the adjacency matrix of the network, m is the number of edges, k_i and k_j are the degrees of nodes i and j, δ is the Kronecker delta (whose value is equal to 1 if i and j are in the same community and 0 otherwise) and C_k represent the community where the node k belongs.

However, as of yet there is no consensus on the definition of hypergraph modularity, as there have been different approaches to define it through the use of (3.7) with certain pairwise adjacency matrix constructed from the hypergraph. In particular, Kumar et al. [47] define the hypergraph modularity using the *clique reduction* of a hypergraph, applying a few extra changes since the degree of each node in the graph arising from the clique reduction does not coincide with the degree in the hypergraph. More precisely, they define the graph with adjacency matrix

$$A_{reduc} = I(\mathcal{H}) W (D_e - \mathbb{I})^{-1} I(\mathcal{H})^t \tag{3.8}$$

where $I(\mathcal{H})$ is the incidence matrix of the hypergraph $\mathcal{H}$, W is the hyperedge weight matrix, D_e is the hyperedge degree matrix and $\mathbb{I}$ is the identity matrix of size $|\varepsilon| \times |\varepsilon|$. Thus, hypergraphs containing a large number of hyperedges will directly impact the computational complexity of algorithms utilizing the modularity score.

Thus, in this method we will choose the partition (given by the dendrogram) that maximizes the modularity with (3.8). Note that, although the derivative graph (Definition 2.4) provides us with an adjacency matrix, we cannot use it to calculate the modularity value since, taking into account the definition of the derivative graph, some nodes of the hypergraph could collapse into new nodes in the derivative graph, in addition to the fact that smaller values of a derivative mean more similarity between nodes, as opposed to smaller values of a derivative in 3.8 meaning a weaker community structure. Furthermore, considering that we want to compare our method with that of [47], we need to use the same adjacency matrix to make such a comparison.

Algorithm 2: Community detection at the maximum modularity

```
Data: Hypergraph $\mathcal{H}$, linkage method
Result: Node partition
/* Compute the derivative graph */
derivatives ← $\mathcal{H}$;
/* Reduced graph */
$G_{\text{reduc}}$ ← $\mathcal{H}$
/* Compute the linkage function */
$Z$ ← linkage_function(derivatives, linkage method);
/* Iterate over the number of clusters in the
 dendrogram */
modularity ← empty_list
for $n$ in $1, \ldots, N$ do
    node partition ← cut_tree($Z$, n)
    $Q$ ← compute_modularity($G_{\text{reduc}}$, node partition)
    modularity ← append($Q$)
end
/* Find $n$ associated to the maximum modularity */
num clusters ← index(modularity == max(modularity))
/* Find partition associated to said number of
 clusters */
node partition ← cut_tree($Z$, num clusters)
```

Iterated partitioning. Note that, depending on the level of granularity one desires for the communities, Algorithm 2 can be applied iteratively to each of the obtained communities (considering the sub-hypergraph associated to them). This generally does increase the obtained modularity.

3.4. Other methods: Iteratively reweighted modularity maximization [47]

In the literature there is already a hypergraph community detection method maximizing the modularity based on the clique reduction (3.8), the Iteratively Reweighted Modularity Maximization (IRMM) algorithm [47]. We will briefly summarize it as we will later use it for comparison with our methods.

The idea of the IRMM algorithm is to apply on the clique reduced graph of $\mathcal{H}$ the Louvain algorithm [52] to get a partition that maximizes the modularity. It then recalculates the weights of the hyperedges using a reweighting function that depends on the current partitioning. The reweighting aims to emphasize the importance of hyperedges that are well captured by the current clusters and reduce the influence of less informative hyperedges. By iteratively reweighting the hyperedges maximizing the modularity, the IRMM algorithm aims to discover a partitioning that maximizes the community structure in the hypergraph. The iterative process helps refine the clustering by gradually adapting the hyperedge weights and node assignments.

4. Applications and real world examples

Now that we have established the theoretical basis of both types of community detection algorithms 1 and 2, we now turn to their application to both synthetic and real networks, together with a comparison with the previously proposed community detection method [47].

We will begin by applying them to a “handcrafted” hypergraph, where we will find that it achieves the partitioning one would expect by simple visual inspection, with both algorithms. We afterwards use a real, labeled dataset of Primary School students [53,54], where we see that it predicts with high accuracy the node labels (individual classes). We finally end this section with the application to a more heterogeneous dataset, a scientific collaboration network.

All numerical simulations were performed on a dedicated server (4.0 GHz Intel Xeon Gold 5220R), with data and codes available at https://github.com/LaComarca-Lab/HyperGraph-Communities for the sake of reproducibility.

4.1. A simple toy model

As a playground for the ideas introduced and discussed in the previous section, we have considered a “toy model”, i.e. a hypergraph designed with the communities we expect in mind. This hypergraph has 14 nodes (labeled as letters in alphabetical order) and 21 hyperedges, and it is shown in 2, we also show the dendrogram one obtains from its associated derivative graph.

With the general method (cutting the dendrogram at the highest gap) we find the partition

$$C_1 = \{a,b,c,d\}, \quad C_2 = \{e,f,g\}, \quad C_3 = \{h,i,j,k\}, \quad C_4 = \{l,m,n\}. \tag{4.1}$$

If we instead use the maximum modularity the partition we find is the same (meaning that cutting the tree at the highest gap provides the partition with maximum modularity, as evidenced by Fig. 2d), with a modularity value of $Q = 0.34056$.

Using the Kumar algorithm, we find a different set of communities:

$$C'_1 = \{a,b,c,d,m\}, \quad C'_2 = \{e,f,g,l,n\}, \quad C'_3 = \{h,i,j,k\} \tag{4.2}$$

with a modularity value of $Q = 0.32944$.

It is easy to see that the only difference between the two partitions is the fact that in the first one we have an extra community C_4, which is split between communities C'_1 and C'_2 in the second one. Given that there is always some subjectivity in the question community detection (and the fact that there is no ground truth in this concocted example), a discrepancy like this is within a reasonable margin.

Having shown that in a simple example everything works as expected, we now turn to more realistic use cases.

Table 1
Comparison between the two methods in terms of the number of communities found, their modularity value, the time taken for each of them (averaged over 100 realizations).

Method	Number of communities	Modularity	Average time
Height-based cut 1	8	0.428	0.008 s
Max. modularity 2	6	0.435	8.256 s

4.2. Validation of the partitioning with real data

One of the main issues one faces when applying community detection to a real network is whether the partition obtained “makes sense”. This sense is usually derived from information outside the graph, often related to the characteristics of the network discussed and/or the information or labels contained in the data which were not used to construct the graph or hypergraph.

It is thus interesting, for the sake of validating the proposed methods, to apply them to a real (not handcrafted, like the previous Toy Model) dataset where there could be some “universal agreement” on the obtained communities, based on the information on the dataset.

For this purpose, we will analyze data on face-to-face interactions between 232 children and 10 teachers over two-day period at a Primary School in Lyon, France [53,54]. The dataset includes a total of 10 different classrooms, with two classrooms for each grade level from first to fifth grade. The interactions were measured using proximity sensors, and each sensor is associated with a group: either a specific classroom for students or the label `'teachers'` for the teachers. Thus, the hypergraph consists on these 242 nodes, representing students as well as teachers, and 12699 hyperedges representing face-to-face interactions captured by the proximity sensors in a 20-second timeframe.

Applying Algorithms 1 and 2, we obtained meaningful results. With the height-based cut Algorithm, we identified 8 communities. Six of them correspond to individual classes 1 A, 1B, 2 A, 2B, 4 A, 4B. The remaining two correspond to the agglomeration of 3 A with 3B, and 5 A with 5B. It should be noted that some communities contain few “outliers” from other classes, but the vast majority falls within their peers. If we instead used the modularity maximization Algorithm, we would find that 1 A, 1B, 2 A and 2B would be fused together.

The conclusions that we can draw from this analysis are twofold: From a community detection perspective, we can see that these Algorithms lead to sensible classification, as they (specially the highest-based cut, in this particular example) match the expectations of community detection applied to a Primary School. From a social network perspective, this hints at the fact that younger children may have communities spanning different levels, whereas older ones are more differentiated by age.

A more quantitative comparison between both methods applied to this dataset can be found in Table 1. It is remarkable to notice that, despite the amount of hyperedges present in the hypergraph, the highest-based cut method performs the community detection task very efficiently. The maximum modularity method suffers from the fact that the construction of the clique-reduction graph (3.8), necessary to compute the modularity, is computationally expensive.

Now that we have verified the sensibility of the obtained partitions in a real dataset, we turn to a more heterogeneous one for the final comparison of the advantage of our methods against the ones previously found in the literature.

4.3. Further results with real data

In order to showcase and compare our algorithm when applied to a real, we have considered Prof. Stefano Boccaletti’s co-authors network as a playground to detect communities. We will first describe the main features of this dataset and the different choices made in order to construct the hypergraph. We will then put our algorithm to the test, and compare it with the other proposals which have been put forward, as we did with the Toy Model previously.

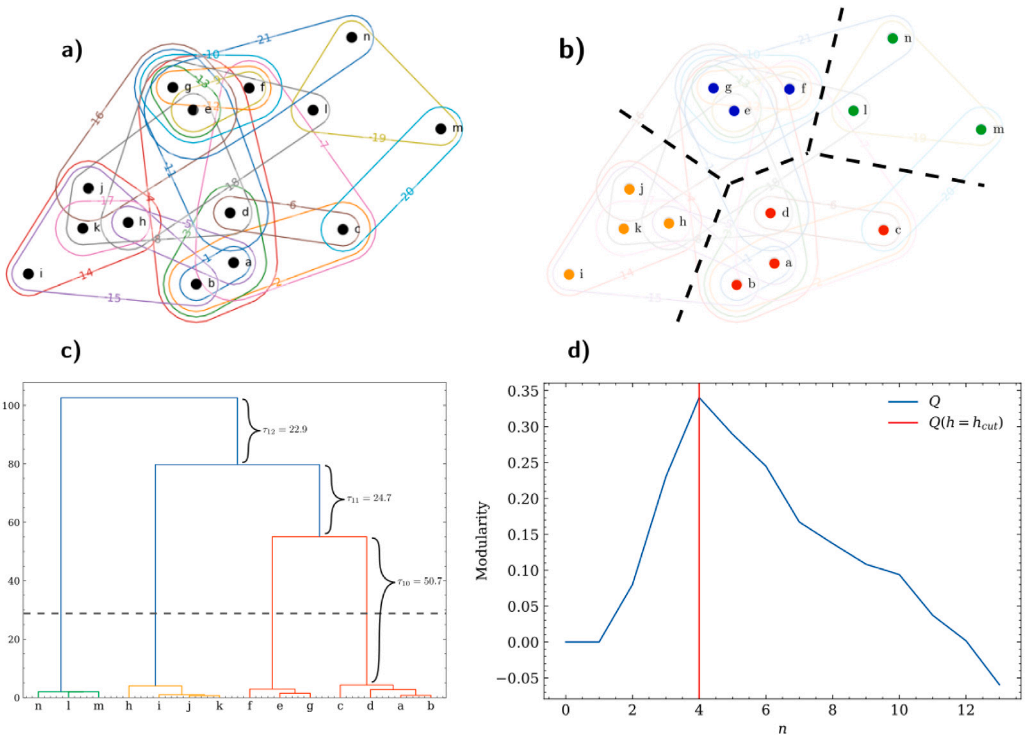


Fig. 2. **(a)** Toy example hypergraph. **(b)** Partition into communities using either of the two methods based on the derivative graph. **(c)** Dendrogram corresponding to the average (UPGMA) clustering via the derivative graph, as discussed in the main text. **(d)** Modularity at each partition of the toy hypergraph given by its dendrogram, where n is the number of communities. The modularity given by the partition at the highest gap is also explicitly shown.

The dataset. We have constructed an initial collaboration hypergraph, where each node is an author who has collaborated with Prof. Boccaletti, with each hyperedge being a scientific publication. The source of the data is Scopus, and it amounts to a total of 338 publications with 413 co-authors.

In order to give additional structure to the hypergraph, we included another set with all publications, this time not including Prof. Boccaletti, of each of the co-authors (which could in principle include other authors which have never collaborated with Prof. Boccaletti, although we do not include them in the hypergraph). The hypergraph is thus enlarged, containing now a total of 15237 hyperedges.

While our algorithm can work with the hypergraph as-is, we have found that the IRMM algorithm proposed in [47] does not converge in reasonable time (more than 24 h in a dedicated server with 4.0 GHz Intel Xeon Gold 5220R) when applied to it. In view of a comparison between both methods, we decided to filter the hypergraph based on the following criteria: we only keep authors with 5 or more publications in common with Prof. Boccaletti (i.e. we are considering *frequent* co-authors). This filtered hypergraph contains 67 authors with 1685 publications among them and/or Prof. Boccaletti.

Community detection. We are going to apply the four methods (derivative graph highest gap cut, maximum modularity, iterated maximum modularity, IRMM), to the Stefano Boccaletti's coauthors hypergraph. Before showing the actual partitions (Fig. 3), let us present some quantitative results and metrics of each of them.

It is clear from Table 2 that, while the best partitioning in terms of modularity is the IRMM one, its computational cost is not worth that slight increase in modularity, being around 320 times slower than our maximum modularity method. It should also be remarked that it could be expected that the IRMM would achieve the greatest modularity, as it is an algorithm specifically designed to optimize this score, unlike the other two. It is hence quite impressive that a simple swipe through the N partitions in the dendrogram, picking the one with maximum modularity, achieves a similar (0.04 off) score so efficiently. Notice that although both IRMM and maximizing the modularity are using the clique reduction, which was expensive in the Primary School example (see previous subsection), the decrease in the number of hyperedges leads to a surprisingly faster computation of the clique reduction. This speed up is, however, hindered by the fact that the reweighting procedure of IRMM is very computationally expensive, as it requires several realizations of the Louvain algorithm [52] over the clique reduction graph.

Table 2
Comparison between the three methods in terms of the number of communities found, their modularity value, the time taken for each of them (averaged over 100 realizations).

Method	Number of communities	Modularity	Average time
Height-based cut 1	16	0.642	0.002 s
Max. modularity 2	9	0.678	0.457 s
Max. modularity, iterated	24	0.564	0.719 s
IRMM [47]	9	0.714	146.604 s

5. Conclusions

In this paper we present two methods for the detection of higher-order network communities (in the context of hypergraphs) that relies on the so-called derivative graph of a hypergraph. As shown through several examples and simulations, the concept of similarity and the semi-distance between nodes induced by the derivative graph of the considered hypergraph are particularly useful for the establishment of a linkage distance between the clusters obtained in the aggregation process.

Through several simulations it is shown that, while the method that gives a slightly better partition in terms of modularity is IRMM, the original methods presented in this work. The first one consists of identifying the largest distance between branches. This algorithm besides being the fastest, is also the most efficient, since it achieves a very high modularity value, close to that of IRMM. Furthermore, IRMM

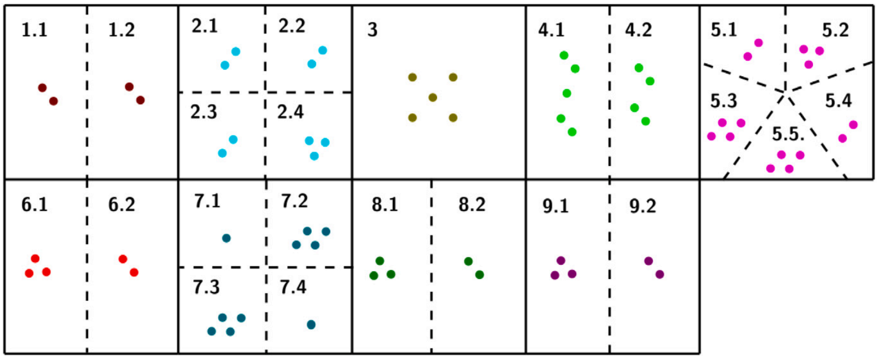


Fig. 3. A visualization of the communities of Prof. Stefano Boccaletti's network of co-authors classified by the modularity maximization method 2. The sub-communities obtained by iterating over each of the partitions are also shown, separated by dashed lines. For reference, the authors belonging to each one are listed below.
1.1: Li D., Havlin S.; **1.2:** Barzel B., Zhang X.;
2.1: Bragard J., Mendoza C.; **2.2:** Kurths J., Zhou C.S.; **2.3:** Mancini H., Maza D.; **2.4:** Meucci R., Allaria E., Arecchi F.T.;
3: Bortolozzo U., Ramazza P.L., Pampaloni E., Residori S., Giaquinta A.;
4.1: Jusup M., Wang Z., Li X., Dai X., Perc M.; **4.2:** Shi L., Guo H., Jia D., Shen C.;
5.1: Sousa P.A.C., Menasalvas E.; **5.2:** Papo D., Buldú J.M., Zanin M.; **5.3:** del-Pozo F., Gutiérrez R., Maestú F., Bajo R.; **5.4:** Jaimes-Reátegui R., Sevilla-Escoboza R.; **5.5:** Navas A., Sendiña-Nadal I., Leyva I., Almendral J.A.;
6.1: Hramov A.E., Koronovskii A.A., Moskalenko O.I.; **6.2:** Maksimenko V.A., Makarov V.V.;
7.1: Raigorodskii A.M.; **7.2:** Frasca M., Moreno Y., Latora V., Gómez-Gardeñes J.; **7.3:** del Genio C.I., Alfaro-Bittner K., Criado R., Romance M.; **7.4:** Musatov D.;
8.1: Guan S., Liu Z., Zou Y.; **8.2:** Qiu T., Bonamassa I.;
9.1: Chavez M., Amann A., Hwang D.-U.; **9.2:** Valladares D.L., Pecora L.M.

is much more computationally expensive and, in addition, in one of the examples shown, it fails to complete the required computation in reasonable time. On the other hand, the second algorithm presented in this work produces an improvement in modularity and, although it incurs in an additional computational cost, this cost is irrelevant and not comparable to IRMM, that produces slightly better modularity values, but has a significantly higher computational cost.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All datasets and methods used in the manuscript are available in the following GitHub repository https://github.com/LaComarca-Lab/HyperGraph-Communities.

Acknowledgments

This work has been partially supported by projects PGC2018-101625-B-I00 (Spanish Ministry, AEI/FEDER, UE, Spain), M2978, M3033 and M1967 (URJC Grants). G. C-A acknowledges funding from the URJC, Spain fellowship PREDOC-21-026-2164.

References

[1] Albert R, Barabasi AL. Statistical mechanics of complex networks. Rev Modern Phys 2002;74:47–97.
[2] Boccaletti S, Latora V, Moreno Y, Chavez M, Hwang D-U. Complex networks: Structure and dynamics. Phys Rep 2006;424:75–308.
[3] Criado R, Flores J, García del Amo A, Gómez-Gardeñes J, Romance M. A mathematical model for networks with structures in the mesoscale. Int J Comput Math 2012;89(3):291–309.
[4] Estrada E. Networks Science. New York: Springer; 2010.
[5] Iglesias S, Criado R. Combining multiplex networks, time series attributes and big data: a new way to optimize real estate forecasting in new york from cab rides. Physica A 2023;609:128306.
[6] Iglesias S, Moral S, Criado R. A new approach to combine multiplex networks and time series attributes: Building intrusion detection systems (IDS) in cybersecurity. Chaos Solitons Fractals 2021;150:111143.
[7] Newman M. Networks: An introduction. Oxford University Press; 2010.
[8] Wasserman S, Faust K. Social network analysis. Cambridge: Cambridge University Press; 1994.
[9] Boccaletti S, Bianconi G, Criado R, Del Genio CI, Gómez-Gardeñes J, Romance M, et al. The structure and dynamics of multilayer networks. Phys Rep 2014;544(1):1–122.
[10] Criado-Alonso A, Aleja D, Romance M, Criado R. Derivative of a hypergraph as a tool for linguistic pattern analysis. Chaos Solitons Fractals 2022;163:112604.
[11] Costa Ld F, Oliveira ON, Travieso G, Rodrigues FA, Villas Boas PR, Antiqueira L, et al. Analyzing and modeling real-world phenomena with complex networks: a survey of applications. Adv Phys 2011;60(3):329–412.
[12] de Arruda HF, Nascimento S, Marinho VQ, Amancio DR, Costa LdF. Representation of texts as complex networks: a mesoscopic approach. J Complex Netw 2018;6(1):125–44.
[13] Dogorovtsev SN, Mendes JFF. Language as an evolving word web. Proc R Soc Lond B 2001;268:2603–6.
[14] Ferrer i Cancho RV. The small world of human language. Proc R Soc London B 2001;286:2261–6.
[15] Latora V, Nicosia V, Russo G. Complex Networks: Principles, Methods and Applications. Cambridge University Press; 2017.
[16] Partida A, Gerasis S, Criado R, Romance M, Giráldez E, Taboada J. The chaotic, self-similar and hierarchical patterns in bitcoin and ethereum price series. Chaos Solitons Fractals 2022;165:112806.
[17] Boccaletti S, De Lellis P, Del Genio CI, Alfaro-Bittner K, Criado R, Jalan S, et al. The structure and dynamics of networks with higher order interactions. Phys Rep 2023;1018:1–64.
[18] Battiston F, Cencetti G, Iacopini I, Latora V, Lucas M, Patania A, et al. Networks beyond pairwise interactions: structure and dynamics. Phys Rep 2020;87492.
[19] Benson A. Three hypergraph eigenvector centralities. SIAM J Math Data Sci 2019;1(2):293–312.
[20] Berge C. Hypergraphs. combinatorics of finite sets. North-Holland; 1989.
[21] Bermond JC, Heydemann MC, Sotteau D. Line graphs of hypergraphs. I. Discrete Math 1977;18(3):235–41.
[22] Criado R, Romance M, Vela-Pérez M. Hyperstructures, a new approach to complex systems. IJBC 2010;20(3):877–83.
[23] Lambiotte R, Rosvall M, Scholtes I. From networks to optimal higher-order models of complex systems. Nat Phys 2019;15:313–20.
[24] Naik RJ. Intersection graphs of graphs and hypergraphs: A survey. 2018, http://dx.doi.org/10.48550/arXiv.1809.08472, arXiv:1809.08472.
[25] Bianconi G. Higher-order networks. Cambridge Univ. Press; 2021.
[26] Gambuzza LV, Di Patti F, Gallo L, Lepri S, Romance M, Criado R, et al. Stability of synchronization in simplicial complexes. Nature Commun 2021;12(1):1–13.
[27] Fortunato S. Community detection in graphs. Phys Rep 2010;486:75–174.
[28] Fortunato S, Newman MEJ. 20 Years of network community detection. Nat Phys 2022;18:848–50.
[29] Newman MEJ. Modularity and community structure in networks. Proc Natl Acad Sci USA 2006;103(23):8577–82.
[30] Evans TS, Lambiotte R. Line graphs, link partitions, and overlapping communities. Phys Rev E 2009;80:016105.
[31] Evans TS, Lambiotte R. Line graphs of weighted networks for overlapping communities. Eur Phys J B 2010;77:265–72.

[32] Fortunato S, Barthélemy M. Resolution limit in community detection. Proc Natl Acad Sci USA 2007;104(1):36–41.

[33] Girvan M, Newman MEJ. Community structure in social and biological networks. Proc Natl Acad Sci USA 2002;99(12):7821–6.

[34] Newman MEJ, Girvan M. Finding and evaluating community structure in networks. Phys Rev E 2004;69:026113.

[35] Newman MEJ, Peixoto TP. Generalized communities in networks. Phys Rev Lett 2015;115:088701.

[36] Palla G, Derényi I, Farkas I, Vicsek T. Uncovering the overlapping community structure of complex networks in nature and society. Nature 2005;435:814–8.

[37] Reichardt J, Bornholdt S. Partitioning and modularity of graphs with arbitrary degree distribution. Phys Rev E 2007;76:015102, (R).

[38] Carletti T, Fanelli D, Lambiotte R. Random walks and community detection in hypergraphs. J Phys Complex 2021;2(2021):015011.

[39] Chien I, Lin C, Wang I. Community detection in hypergraphs: Optimal statistical limit and efficient algorithms. Proc Mach Learn Res 2018;84:871–9, Available from https://proceedings.mlr.press/v84/chien18a.html.

[40] Contisciani M, Battiston F, De Bacco C. Inference of hyperedges and overlapping communities in hypergraphs. Nature Commun 2022;13(2022):7229.

[41] Zhang Y, Lucas M, Battiston F. Higher-order interactions shape collective dynamics differently in hypergraphs and simplicial complexes. Nature Commun 2023;14:1605.

[42] Criado-Alonso A, Aleja D, Romance M, Criado R. A new insight into linguistic pattern analysis based on multilayer hypergraphs for the automatic extraction of text summaries. Math Methods Appl Sci 2023;2023:1–18.

[43] Jaccard P. Distribution de la flore alpine dans le bassin des dranses et dans quelques regions voisines. Bull Soc Vaudoise Des Sci Nat 1901;37:241–72.

[44] Brusco M, Cradit JD, Steinley D. A comparison of 71 binary similarity coefficients: The effect of base rates. PLoS One 2021;16(4):e0247751.

[45] Hamers L, Hemeryck Y, Herweyers G, Janssen M, Ketters H, Rousseau H, et al. Similarity measures in scientometric research: The jaccard index versus Salton's cosine formula. Inf Process Manage 1989;25(3):315–8.

[46] Vijaymeena MK, Kavitha K. A survey on similarity measures in text mining. Mach Learn Appl 2016;3(1):1–28.

[47] Kumar T, Vaidyanathan S, Ananthapadmanabhan H, et al. Hypergraph clustering by iteratively reweighted modularity maximization. Appl Netw Sci 2020;5:52.

[48] Cormack RM. A review of classification (with discussion). J R Stat Soc Series A. General 1971;134(3):321–67.

[49] Sneath PHA, Sokal RR. Numerical taxonomy (the principles and practice of numerical classification.). San Francisco, Calif: W. H. Freeman and Co; 1973.

[50] Gordon AD. Classification. 2nd ed.. Chapman & Hall/CRC.; 1999.

[51] Fernández A, Gómez S. Solving non-uniqueness in agglomerative hierarchical clustering using multidendograms. J Classification 2008;25:43–68.

[52] Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008;2008:P10008.

[53] Benson AR, Abebe R, Schaub MT, Jadbabaie A, Kleinberg J. Simplicial closure and higher-order link prediction. Proc Natl Acad Sci 2018;115(48):E11221–30.

[54] Stehlé J, Voirin N, Barrat A, Cattuto C, Isella L, Pinton J-F, et al. High-resolution measurements of face-to-face contact patterns in a primary school. PLoS One 2011;6(8).