Omega 138 (2026) 103387

Contents lists available at ScienceDirect

Omega

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

Modularity for hypergraph clustering: Methodologies and applications☆

Stefano Benati [a], Justo Puerto [b], Francisco Temprano [b,*]

[a] Dipartimento di Sociologia e Ricerca Sociale, Università di Trento, Via Verdi 26, 38122 Trento, Italy
[b] IMUS. Universidad de Sevilla, Avda. Reina Mercedes, s/n, 41012 Sevilla, Spain

ARTICLE INFO

Keywords:
Complex networks
Hypergraph clustering
Modularity
Mathematical programming

ABSTRACT

We consider the problem of detecting communities of the vertices of data structures defined as hypergraphs. There are many methods that were devised to find communities on simple graphs, one of the most important being modularity maximization. Therefore we extend the definition of modularity to deal with hypergraphs, too. The new definition considers whether a hyperedge as internal or not to a community, and in the affirmative case, the hyperedge is assigned with a weight defining the level of cohesiveness of their vertices. Next, we formulate a mixed-integer linear programming model to hypergraph modularity maximization, whose optimal solutions consist in the best vertex partitions of the hypergraph. Previous attempts of partitioning hypergraphs did suggest the projection of hyperedges onto simple graphs, replacing every hyperedge with a complete subgraph. So, we compare our proposal with the previous methodologies, including other hypergraph modularity definitions, and we find that we improve the quality of the partition. Finally, we apply the methodology to the analysis of survey data, and we show how hypergraph modularity clustering highlights some peculiar data structures that otherwise would remain hidden to researchers.

1. Introduction

Finding communities is one of the basic challenges in complex networks analysis. The problem consists in finding a partition of vertices, in such a way that the vertices within a group can be considered as a community. There are many ways to determine communities and a broad taxonomy may distinguish between dynamic models, such as simulating random walk to identify persistent states, inference based models, in which communities are found assuming a probabilistic generative model, optimization models in which the best partition is the one that maximizes an objective function, see surveys [1–3] for a comprehensive treatment about the issue. The most important optimization model is to maximize the modularity, see [4]. The modularity is an index that compares the number of the arcs within a community with its expected number, assuming the same degree distribution between the empiric and theoretical graph (the so-called configuration model). Then, the best vertex partition is the one that maximizes the modularity. Therefore, finding communities is reduced to solving a combinatorial problem that can be formulated as an Integer Linear Programming (ILP) model. In [5], it has been formulated as Clique Partitioning (CP), a known NP-hard combinatorial problem, see [6]. Taking advantage of the graph structure, the CP model can be formulated with a lower number of constraints, see [7]. The CP model can also be formulated as Set Partitioning and solved by column generation, [8]. Finally, in [9], a tailored branch&cut method based on the CP partition model has been developed, whose computational performance seems very promising. However, the aforementioned models must solve a NP-hard problem, so that computational times can be prohibitive. Applications with some hundreds of nodes can only be solved by heuristic methods, whose output is a partition with a modularity value (hopefully) close to the unknown optimum. Most heuristic methods are devised around a greedy scheme, that is, communities are merged every time they result in a new partition with higher modularity. Early models differed on the way the merging process is implemented, see [10,11], or on the use of marginal information to lead the process, see [12,13]. Moreover, the greedy solution can be improved by some post-optimization adjustments, such as communities recombinations or Kerninghan–Lin bisection, see [14], or communities refinement, see [15]. Finally, broad meta-heuristic principles can be applied to modularity maximization, such as Greedy Randomized Adaptive Search (GRASP), see [16], Variable Neighbourhood Search (VNS), see [17,18], and Neural Networks (NN), see [19]. Most of the mentioned methods are compared in [20,21]. There, it has been found a remarkable difference between the exact and the heuristic methods: one may think that approximate solutions of the optimal modularity, nevertheless provides partitions that are very close to the optimal clustering. Unfortunately,

☆ Area: Decision Analysis and Preference Modeling. This manuscript was processed by Associate Editor A. Ishizaka and Area Editor L. Dias.

* Corresponding author.
E-mail addresses: stefano.benati@unitn.it (S. Benati), puerto@us.es (J. Puerto), ftgarcia@us.es (F. Temprano).

https://doi.org/10.1016/j.omega.2025.103387
Received 17 March 2025; Accepted 17 June 2025
Available online 20 August 2025

this is not the case, and one should be aware that heuristic partitions can be very poor approximations of optimal partitions.

In previous research, the modularity equation has been modified in many directions with the purpose to increase both its accuracy and flexibility. For example, the modularity may contain a resolution parameter, with the aim of avoiding the so-called resolution limit, see [22]. Another possibility is to consider the ratio between the modularity and its subset size, that is called the modularity density, see [23,24]. Finally, the persistence probability considers the ratio between the internal arcs and the all the arcs incident to the subset, that can be interpreted as the probability of a random walker to remain in that subset, see [25,26]. In some applications, it is required to obtain overlapping communities, so that the definition of modularity is modified to take into account multiple membership, see [27–29]. Recently, in [30], some techniques are developed to detect the most influential nodes in a complex network. This is a crucial information, since those nodes have a great impact on the structure of the networks that are studied. Finally, community detection is part of the large families of clustering and classification problems, that have a great impact on the management science and the analysis of decisions, see [31,32].

A hypergraph is a graph (or a network) in which arcs can be incident to more than two nodes, and previous contributions recognized the importance of community detection on hypergraphs too, see [33]. In those applications, what has been proposed so far has been to project the hypergraph onto a simple graph, that is, replacing every hyperedge by a complete subgraph of simple arcs. Then, to apply any method for modularity maximization to it, see for example [34–38]. However, the procedure is not flawless. After the projection, a hyperedge is replaced by as many arcs as each node pairs belonging to that hyperedge. Therefore, after that the partition is calculated, *one* hyperedge may result to define *two or more* different clusters, if two or more of its node pairs are in different subsets. So, that hyperedge is *internal* to two (or more) communities, but it is somewhat a contradiction, the hyperedge should be rather considered as *external* to all the communities. Our methodology will try to improve this state-of-the-art.

Our definition of hypergraph modularity retains all the basic elements that were used to define modularity on simple graphs. They are: the definition of the internal arcs to communities, their weighted sum, and their comparison with a null hypothesis under the configuration model. Next, we show how hypergraph modularity maximization can be formulated as a Mixed-Integer Linear Programming model and solved using standard software. We compare the partition quality that is obtained by our new modularity definition with the previous approaches, that considered the hyperedge projection, and we find that our suggestion actually improves the quality of the partitions.

In this contribution, we provide an exact method to solve hypergraph modularity maximization, and we are aware that this method (conversely to heuristics) can only solve small size instances. However, there is a strategic advantage on the use of exact methods instead of heuristic to do community detection, that is, we are safe that our quality improvements are not biased or hindered by the heuristic techniques that are used to calculate the partitions, as is discussed and proved by Aref and Mostajabdaveh [21]. Nevertheless, we found an application of hypergraph clustering that is small sized and to which exact methods can be readily applied. This application is clustering opinions from survey data. In [39], it has been noted that in some opinion surveys, there are questions to which respondents must elicit a maximum number of answers selected from a closed list. Answers can be interpreted as the nodes of a graph and every respondent is represented by an arc (or in our case, a hyperedge) connecting its elicited pair (or elicited subset). Recognizing this representation, community detection can be applied to cluster the answers of the list into general categories, and then use these categories to refine the statistical analysis.

In conclusion, the contributions of this paper can be summarized as follows:

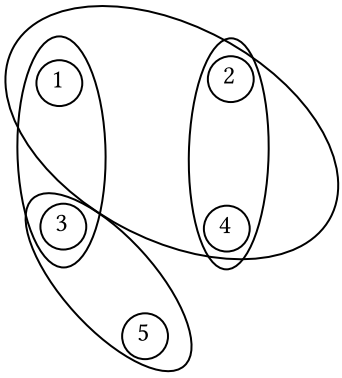


Fig. 1. Example of a simple hypergraph.

1. To introduce a new definition of modularity that can be applied to hypergraphs;
2. To propose an optimization model based on mathematical programming to find the optimal hypergraph partition;
3. To compare its result with the outcome of other models suggested so far, and finding that they are an improvement;
4. To show that hypergraph clustering can be applied to survey data, to carry on statistical analysis otherwise not available.

The paper is organized in 6 sections: in Sections 1 and 2 the problem is presented and motivated, and the methodologies that are already proposed in literature are analysed. In Section 3, we describe how to modify the modularity index to hypergraph data. In Section 4, hypergraph modularity maximization is applied to random artificial hypergraphs, generated with a procedure inspired by Lancichinetti et al. [40]. In Section 5, a heuristic algorithm is proposed and compared with other well-known algorithm from literature. Finally, in Section 6, we present a case study where we apply our best method, based on performance on synthetic instances, to some real data, interpreting and drawing conclusions on their results.

2. Background

Let $V = \{1, \dots, n\}$ be a set of nodes and E be a set of non-empty subsets of V, such that repeated subsets and isolated subsets of cardinal equal to one are not allowed, $H = (V, E)$ is defined as a simple hypergraph with vertex set V and hyperedge set E (see Fig. 1). In order to introduce some notation, we consider that $m = |E|$ is the number of hyperedges, $\delta_i = |\{e \in E : i \in e\}|$ is the number of hyperedges to which a node i belongs to and $\delta = \sum_{i=1}^{n} \delta_i = \sum_{e \in E} |e|$ is the sum of the hyperedge sizes. δ_i can be also understood as the adjacency degree of node i.

The concept of hypergraph can be extended to the direct and weighted cases. For the weighted case, it is assigned a non-negative weight $w_e \geq 0$ to each hyperedge $e \in E$. The adjacency degree of a node in a weighted hypergraph is defined as $\delta_i = \sum_{\substack{e \in E \\ i \in e}} w_e$ and the total sum of weights as $W = \sum_{e \in E} w_e$. If repeated hyperedges are allowed, w_e can represent the number of repetitions. For the directed case, we must consider a new set E^d of pairs of V non-empty subsets, so each directed hyperedge or hyperarc $e \in E^d$ is equivalent to a pair origin-destiny, $e = (O, D)$, such that $\emptyset \neq O, D \subseteq V$.

Since most of the methods already proposed in literature are based on transforming the hypergraph into a graph to which a large number of well-known clustering algorithms can be applied, see [34,38,41,42], we remind some classic metrics for the graph case in this preliminary section. In particular, in this document we have extended the modularity proposed by Newman [4] to the hypergraph case but without transforming it into a graph.

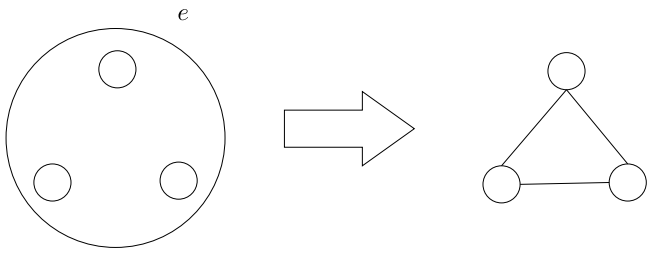


Fig. 2. Hypergraph projection.

Given a simple graph $G = (V, E')$ such that $m' = |E'|$, A is its adjacency matrix and k_i is the adjacency degree of a node i, the modularity value of a node set partition $P = \{C_1, \dots, C_q\}$ is defined as:

$$\frac{1}{2m'} \sum_{s=1}^{q} \sum_{i,j \in C_s} \left(A_{ij} - \frac{k_i k_j}{2m'} \right). \tag{1}$$

The modularity function (1) compares the number of internal edges of the partition P with the expected number of internal edges of a random graph generated by the configuration model proposed by Newman [43]. Actually, the exact expected number of edges between two nodes i and j is $\frac{k_i k_j}{2m'-1}$ but it is commonly approximated by $\frac{k_i k_j}{2m'}$ which allows to extend this expression into the weighted case. The extension to direct and weighted cases can be found in [4,44]. In addition, [29] explain in detail all these exact probabilistic calculations.

Another well-known quality function is the normalized cut which measures the external edge density of a specific k-partition of V. Given a k-partition $\{C_1, \dots, C_k\}$ of V, its k-way normalized cut is defined as:

$$\sum_{s=1}^{k} \frac{\sum_{\substack{i \in C_s \\ j \notin C_s}} A_{ij}}{\sum_{i \in C_s} k_i}. \tag{2}$$

The extension of (2) to weighted graphs can be also found in [45]. Although most of the methods proposed in literature for the Minimum Normalized Cut problem are heuristic algorithms, as in [45–47], recently, [48] present a wide variety of formulations and methods based on mathematical programming that solve exactly the problem.

Once the classic quality measures on which we have based the methods of this paper have been presented, it would be interesting to review the different methodologies that have been already proposed in the literature of hypergraph clustering. As we comment above, it is really common to transform a hypergraph into a graph with approximately the same information. This process can be seen in many papers as [34–38], and in most of them it consists in generating a clique or a complete subgraph for each hyperedge.

Formally described, given a hypergraph $H = (V, E)$, we will create a new graph $G = (V, E')$ such that for each hyperedge $e \in E$ and for each pair of different nodes $i, j \in e$ we generate an edge $(i, j)_e \in E'$. Thus, the number of edges between two different nodes i and j in G is equal to the number of common hyperedges in H.

The main disadvantage of this transformation is that for each hyperedge e we are generating $\frac{|e|(|e|-1)}{2}$ edges, which means that the size of each edge has a great impact and possibly the larger hyperedges have more influence than the smaller ones. Since objective functions as modularity and normalized cut apply a normalizing operation to the internal and external edge number of the graph, this inconvenient is partially solved by them. However, trying to avoid this, many of these transformations into simple graphs normalize the weights of the generated cliques. For example, for each edge from a clique generated by a hyperedge e, [34] assign a weight equal to $\frac{1}{|e|}$ and [42] the weight $\frac{1}{|e|-1}$.

The extension of the above process to weighted hypergraph is trivial. Each edge is associated with the weight of the hyperedge from which it comes. In the directed case, instead of generating cliques, we are going to create complete bipartite subgraphs considering all the possible pairs of origin–destination of the hyperedges.

3. Modularity for hypergraph clustering

As we have explained in the above section, we have focused on modularity, as clustering quality measure, and we have tried to apply its concepts and methodologies to a hypergraph structure. Given a specific partition of V, most of these classic measures are based on distinguishing each edge into two classes: Internal and external edges. While modularity tries to maximize the difference between the number of internal edges and the expected number of internal edges on a random graph (generated by the configuration model [43]), k-way normalized cut looks for the k-partition that minimizes its external edge density.

If we want to deal with the hypergraph community detection problem without modifying or transforming our original structure into a graph, the first thing we must do is to define and clarify the definition of internal and external hyperedge. In [49–51], given a subset $S \subseteq V$, a hyperedge $e \in E$ is an internal hyperedge of S if $e \subseteq S$. This is the strictest definition in literature, since all nodes of an internal hyperedge must belong to a community. The drawback of the definition is that many hyperedges are useless to define communities, as just one node with different membership is enough to define the hyperedge as external. Conversely, in [52], a hyperedge is internal if more than a half of its nodes belong to the same community. However, we argue that the minimum internal proportion of nodes for an internal hyperedge could be a flexible parameter controlled by researchers to improve the ability of the methodology to detect hidden clusters. Our new definition of internal and external hyperedge depends on a parameter $\gamma \in [0, 1]$. A hyperedge e is internal to a community S if the percentage of nodes of e that belong to S is greater or equal to γ, i.e., $|S \cap e| \geq \gamma |e|$. Clearly, γ represents how strictly an edge is defined as internal. If $\gamma = 1$ we have the interpretation of Chekuri and Xu [49], Kamiński et al. [50] and Fox et al. [51], if $\gamma = 0.5$ we have the definition of Kamiński et al. [52], but intermediate values are possible. Thus, the main novelty of our approach is that the γ selection is a design decision, while the remaining definitions fix a value of γ that must be greater or equal than 0.5. This is not a weakness but a design decision.

For hypergraphs, the same concept of modularity is more complex than the case of a graph, and actually it can be modified: see [50], using a generalization of the Chung-Lu model, and see [53]. In our contribution, we will consider that the importance of an hyperedge to define a community depends on the fraction of nodes that are internal. Therefore, in the following subsections, we first revise all the modularity measures that were previously defined for hypergraphs, and then we will highlight the originality of our proposal.

3.1. Hypergraph modularity by projection

Previous attempts of applying modularity clustering to hypergraphs are based on the hypergraph projection into a simple graph. Firstly, each hyperedge is replaced by a clique between its vertices, to obtain a standard graph with multiple edges, with as many edges between vertex i and j as the number of hyperedges to which the pair i, j belongs. Secondly: modularity optimization is applied to the resulting simple graph. This is the approach followed by Agarwal et al. [37], Li and Milenkovic [38], Rodriguez-Velazquez [36], Zhou et al. [34] and Zien et al. [35]. The difference between the methods can be ascribed to the interpretation of the terms k_i, that is, the degree of vertex i, in formula (1), that was originally developed for unweighted graph. When replacing a hyperedge by the edges of a clique, authors tried to avoid to overrepresent the information of the hyperedge, giving weights to new edges that are decreasing with respect to the size of the hyperedge. To

formalize the approach, an edge that appears to replace a vertex pair belonging to a hyperedge can be expressed as:

$$A^e_{ij} = \begin{cases} 1, & \text{if } i,j \in e,\ i \neq j; \\ 0, & \text{otherwise.} \end{cases}$$

As multiple edges are generated for each hyperedge, authors used a normalized function $\Delta(e)$ to weight A^e_{ij}, and therefore to use a weighted modularity function to assess partitions. In the weighted version of modularity, in Eq. (1) the term A_{ij} is replaced by $A_{ij}/\Delta(e)$, and the terms k_i must be modified accordingly. The *weighted* degree of vertex i is:

$$k_i = \sum_{j=1}^{n} \sum_{\substack{e \in E \\ i \in e}} \frac{A^e_{ij}}{\Delta(e)} = \sum_{\substack{e \in E \\ i \in e}} \frac{|e|-1}{\Delta(e)}.$$

So, the sum of the weighted degrees is $2m' = \sum_{i,j \in V} \sum_{e \in E} \frac{A^e_{ij}}{\Delta(e)} = \sum_{e \in E} \frac{|e|(|e|-1)}{\Delta(e)}$. As mentioned, authors proposed different values for $\Delta(e)$: for example, $\Delta(e) = |e|$ in [34], $\Delta(e) = |e| - 1$ in [42]. A consistent option, followed by this paper, is $\Delta(e) = \frac{|e|(|e|-1)}{2}$, since $\frac{|e|(|e|-1)}{2}$ edges are generated for each hyperedge e. Finally, the modularity function (1) developed for simple graphs can be extended to hypergraphs with the same formula applied to the induced graph. Consider a node partition $\{C_1, \ldots, C_q\}$, then its modularity, later on defined as *projected modularity*, is:

$$Q_1 = \frac{1}{\sum_{e \in E} \frac{|e|(|e|-1)}{\Delta(e)}} \sum_{s=1}^{q} \sum_{i,j \in C_s} \left(\sum_{e \in E} \frac{A^e_{ij}}{\Delta(e)} - \frac{\left(\sum_{\substack{e \in E \\ i \in e}} \frac{|e|-1}{\Delta(e)} \right) \left(\sum_{\substack{e \in E \\ j \in e}} \frac{|e|-1}{\Delta(e)} \right)}{\sum_{e \in E} \frac{|e|(|e|-1)}{\Delta(e)}} \right). \tag{3}$$

Depending on $\Delta(e)$ value, expression (3) can be simplified. For example, if $\Delta(e) = |e| - 1$, then $k_i = \delta_i$ and the *projected modularity* is equal to:

$$\frac{1}{\delta} \sum_{s=1}^{q} \sum_{i,j \in C_s} \left(\sum_{e \in E} \frac{A^e_{ij}}{|e|-1} - \frac{\delta_i \delta_j}{\delta} \right).$$

3.2. Hypergraph modularity by inclusion

The previous approach has the merit of its simplicity. However, the method is not flawless. The main ambiguity is the lack of the definition of *internal* and *external* arcs, a concept that is at the root of Eq. (1). In a simple graph, an arc is within a community if its nodes belong to the same cluster, otherwise it is outside of the communities. In a simple graph, this is a true or false condition, given that a pair belongs or does not belong to the same community. However, when applied to hypergraphs, this notion is vanishing, and actually, it does not appear in Eq. (3). Indeed, in that formula a hyperedge can be internal to *more* than one community and being external too. For example, if $e = \{v_1, v_2, v_3, v_4\}$ and supposing that in modularity optimization v_1 and v_2 are in one cluster, v_3 and v_4 are in a second cluster, then the hyperedge e, through the arcs (v_1, v_2) and (v_3, v_4) is internal to *both* clusters, but in a sense, it is also external because all simple arcs between nodes of different clusters, such as (v_1, v_3), are external. Clearly, if one wants to take advantage of algorithms developed for graphs to hypergraphs as well, then this ambiguity can be forgotten. However, one may also wonder whether a new definition of modularity can be defined, based on a clear dichotomy between internal and external hyperedges.

In the following, we are introducing a new measure of hypergraph modularity. We retain the main logical steps that were used to define Equation (1). They are: First, to define whether a hyperedge is internal or not to some cluster; second, to compare its presence (or absence) to its expected value, given that the hypergraph emerged as the random occurrence of the configuration model. Moreover, we extend that model by considering that a hyperedge can belong to a cluster with different level of membership, as it can be the case that only a fraction of its vertices are contained in a cluster. Therefore, if it is established that the hyperedge is internal or not to a cluster, then it must be weighted by its membership level.

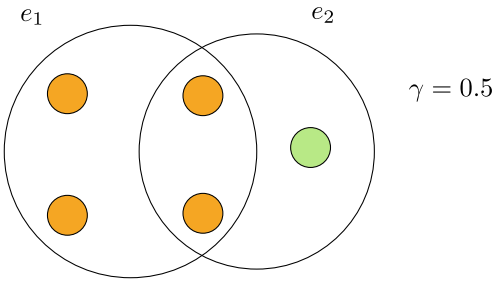


Fig. 3. e_1 and e_2 are internal hyperedges of orange community for $\gamma = 0.5$.

Step 1: Defining internal hyperedges: We define a hyperedge as internal to subset $S \subseteq V$, if it is incident to at least a given fraction of S nodes. More formally, let $\gamma \in [0,1]$ be the contribution parameter, hyperedge e is internal to S, if and only if $|S \cap e| \geq \gamma |e|$. Analogously, for each hyperedge e, we can define a lower bound $\gamma_e = \lceil \gamma |e| \rceil$ such that e is internal to S, if and only if $|S \cap e| \geq \gamma_e$. Relevant values used in our contribution are $\gamma > 0.5$, but values below that threshold are still admissible. Let A^e_S be defined as:

$$A^e_S = \begin{cases} 1, & \text{if hyperedge } e \text{ is internal of } S, \text{ i.e., } |S \cap e| \geq \gamma_e, \\ 0, & \text{otherwise.} \end{cases}$$

The number A_S of the internal hyperedges of S is:

$$A_S = |\{e \in E : \ |S \cap e| \geq \gamma_e\}|,$$

so that $A_S = \sum_{e \in E} A^e_S$. The reader may note that this new internal hyperedge approach allows hyperedges to be internal to multiple communities when the γ threshold is met. Thus, the same hyperedge could contribute to the modularity of more than one community if the γ condition is satisfied in each. Formally, given an integer number p such that $\frac{1}{p+1} < \gamma \leq \frac{1}{p}$, then every hyperedge can be internal to at most p communities.

Step 2: The weight of a hyperedge: Hyperedges should have different weights depending on the fraction of their nodes that are contained in a community. For example, consider Fig. 3, there, a cluster $C_s \subseteq V$ of four nodes is coloured in orange, and a fifth node not belonging to C_s is coloured in green. Two hyperedges are drawn. The first e_1 is incident to all the orange nodes, the second e_2 in incident to two orange and to the green nodes. For $\gamma = 0.5$, both e_1 and e_2 are internal to the community C_s, however, we should take into account that e_1 is more *inside* than e_2, and therefore it should be more important than e_2 to define C_s. Therefore, weights $w(e, C_s)$ should be defined to measure the number of nodes of e that are contained in C_s. If e is internal to C_s, we propose to use the value $w(e, C_s) = (|C_s \cap e| - \gamma_e + 1)$, being $\gamma_e := \lceil \gamma |e| \rceil$, which corresponds to the number of internal nodes that exceed the threshold γ_e. If e is not internal to C_s, then $w(e, C_s) = 0$. So, these weights can be defined as $w(e, C_s) = (|C_s \cap e| - \gamma_e + 1) A^e_{C_s}$. Note that if the modularity is defined on graphs, there is no need to consider weights: in that case $w(e, C_s) = 1$ or 0, depending on whether $e = (i, j) \subseteq C_s$ or not.

Step 3: The configuration model for hypergraphs: Given a vertex clustering on a simple graph G, modularity is an index that compares the number of the arcs internal to clusters with their expected value conditional to the so-called graph configuration model of G, that is, a random graph with no community structure, but with the same degree distribution of G. For a hypergraph H, it is complex to devise a

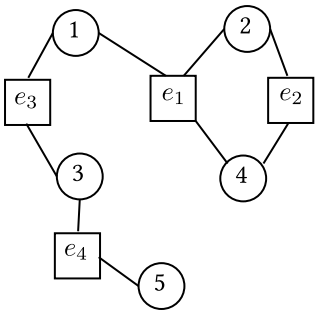


Fig. 4. Alternative representation of a hypergraph from Fig. 2.

configuration model, that is a random hypergraph, still keeping some important structural properties of H. In the literature, configuration models for hypergraphs have been proposed in [50,52,54]. However, our hypergraph configuration model results in three properties that were not all satisfied by previous approaches. They are:

1. a configuration hypergraph with the same node degrees of H;
2. the same number of hyperedges as H;
3. all random hyperedges have the same size as the hyperedges of H.

A simple way to understand the random generation process of the configuration model is to represent each hyperedge as a facility to which the nodes that belong to that hyperedge are connected, see Fig. 4. So, the hypergraph $H = (V, E)$ is represented by a bipartite graph $B = (V, E, Q)$, in which one vertex class V represents the nodes, the other vertex class E represents the edges, and there is an arc $(v, e) \in Q$ if $v \in e$.

As in the original configuration model, every arc of B is cut into two stubs, one stub incident to a facility (hyperedge) node, the other stub incident to a node. Therefore, we obtain δ node stubs and δ facility stubs. All the random graphs of the configuration model are all the matchings between the two families of stubs. In Fig. 5, we have drawn the process of obtaining a random configuration from a given occurrence of H.

Since every random hypergraph is defined by a given matching between δ node stubs and δ facility stubs, then there are $\delta!$ random hypergraphs that can be obtained by the configuration model, all with the same probability.

Consider a subset $S \subseteq V$, the adjacency degree of S is $\delta_S = \sum_{i \in S} \delta_i$. Next, consider S and a hyperedge e. In some random hypergraphs, e contains exactly t nodes of S. That is, exactly t stubs incident to e are paired with t elements of S. The number of hypergraphs with this property is:

$$\binom{\delta_S}{t}\binom{\delta-\delta_S}{|e|-t}|e|!(\delta-|e|)!.$$

Indeed, there are $\binom{\delta_S}{t}$ possible subsets of t stubs that can be selected out of S and $\binom{\delta-\delta_S}{|e|-t}$ possible subsets of $|e|-t$ stubs can be selected from $V \backslash S$. Those $|e|$ stubs can be matched in $|e|!$ possible ways with the stubs incident to e. Moreover, these combinations must be combined with the remaining $(\delta - |e|)!$ possible matchings between the rest of the stubs. Finally, if we divide that value by the total number of possible cases, the probability of the random occurrence of a hypergraph in which $|S \cap e| = t$ is:

$$\frac{\binom{\delta_S}{t}\binom{\delta-\delta_S}{|e|-t}|e|!(\delta-|e|)!}{\delta!} = \frac{\binom{\delta_S}{t}\binom{\delta-\delta_S}{|e|-t}}{\binom{\delta}{|e|}}.$$

One can observe that the probability corresponds to the hypergeometric distribution $H(\delta, \delta_{C_s}, |e|)$ in which we are taking $|e|$ elements from a population of δ elements with δ_{C_S} of one of the classes. There are t of these $|e|$ elements that belong to a group of cardinality δ_S corresponding to the stubs that are adjacent to S. The remaining $|e|-t$ are from a group of cardinality $\delta - \delta_S$, corresponding to the stubs not adjacent to S. Note that the above probability only makes sense if $\max\{0, |e| + \delta_S - \delta\} \le t \le \min\{\delta_S, |e|\}$. So, considering we need at least γ_e connections between S and e, the exact expected value of A^e_S is:

$$E[A^e_S] = \sum_{t=\max\{\gamma_e, |e|+\delta_S-\delta\}}^{\min\{|e|,\delta_S\}} \frac{\binom{\delta_S}{t}\binom{\delta-\delta_S}{|e|-t}}{\binom{\delta}{|e|}}.$$

Let us notice that the above expression is equal to the probability of e being internal to S.

Step 4: Modularity for hypergraphs. Modularity compares the weighted sum of the internal hyperedges to community to the expected value of the same weighted sum under the configuration model. The weighted sum of the internal hyperedges to community can be expressed as:

$$\sum_{s=1}^{q}\sum_{e\in E}(|C_s \cap e| - \gamma_e + 1)A^e_{C_s}.$$

From *Step 3*, we can calculate the exact expected value of the above weighted sum:

$$\begin{aligned}
E\left[\sum_{s=1}^{q}\sum_{e\in E}(|C_s \cap e| - \gamma_e + 1)A^e_{C_s}\right] &= \sum_{s=1}^{q}\sum_{e\in E} E\left[(|C_s \cap e| - \gamma_e + 1)A^e_{C_s}\right] \\
&= \sum_{s=1}^{q}\sum_{e\in E}\sum_{t=\max\{\gamma_e, |e|+\delta_{C_s}-\delta\}}^{\min\{|e|,\delta_{C_s}\}} (t - \gamma_e + 1) \\
&\quad \times P[|C_s \cap e| = t] \\
&= \sum_{s=1}^{q}\sum_{e\in E}\sum_{t=\max\{\gamma_e, |e|+\delta_{C_s}-\delta\}}^{\min\{|e|,\delta_{C_s}\}} (t - \gamma_e + 1) \\
&\quad \times \frac{\binom{\delta_{C_s}}{t}\binom{\delta-\delta_{C_s}}{|e|-t}}{\binom{\delta}{|e|}}.
\end{aligned}$$

Therefore, summarizing our discussion above, we get to the following *hypergraph modularity* equation:

$$Q^{\gamma}_2 = \frac{1}{m}\sum_{s=1}^{q}\sum_{e\in E}\left((|C_s \cap e| - \gamma_e + 1)A^e_{C_s} - \sum_{t=\max\{\gamma_e, |e|+\delta_{C_s}-\delta\}}^{\min\{|e|,\delta_{C_s}\}} (t - \gamma_e + 1)\frac{\binom{\delta_{C_s}}{t}\binom{\delta-\delta_{C_s}}{|e|-t}}{\binom{\delta}{|e|}}\right). \tag{4}$$

In particular, the expression $(|C_s \cap e| - \gamma_e + 1)A^e_{C_s}$ takes value 0 if e is not internal to community C_s, otherwise it takes the value of the weight $(|C_s \cap e| - \gamma_e + 1)$. Then the same expression is compared to its exact expected value. Our proposal for a new modularity function (4) is similar to the modularity expression proposed by Kamiński et al. [52], as both consider partial memberships and weights, depending on the inclusion degree of the hyperedges. However, our model is more flexible, since the membership threshold γ is controlled. Moreover, our configuration model is more accurate: the actual hypergraph H and its random counterpart share three important structural properties: adjacency degree, number of hyperedges and hyperedge size, while using binomial distributions in the Chung-Lu model can generate hypergraphs where the adjacent degree of each node is different between the two hypergraphs.

The two modularity measures (3) and (4) can be used to determine the best partition of V. In the following, we will introduce two Mixed Integer Linear Programming (MILP) formulations for that purpose. It is worth to note that with some straightforward modification, our models can solve the modularity proposed by Kamiński et al. [52] as well.

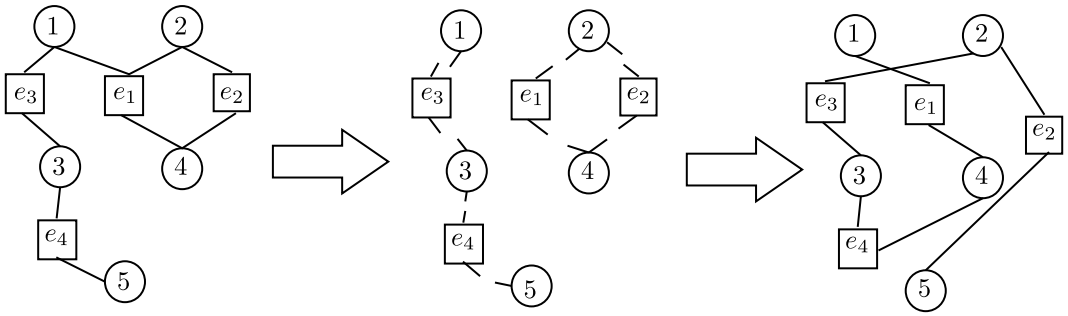


Fig. 5. New configuration model process example with hypergraph from Figs. 2 and 4.

3.2.1. MILP formulations for maximum modularity

The projected modularity (3) is maximized through the clique partition model. The following variables define the partition:

$$x_{ij} = \begin{cases} 1, & \text{if nodes } i \text{ and } j \text{ are in the same community,} \\ 0, & \text{otherwise,} \end{cases}$$

for any $i, j = 1, \dots, n$, such that $i < j$. The maximum modularity/clique partition model is the following integer linear program:

$$(F_x^1): \quad \max \quad \frac{1}{\sum_{e\in E} \frac{|e|(|e|-1)}{\Delta(e)}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(\sum_{e\in E} \frac{A_{ij}^e}{\Delta(e)} - \frac{\left(\sum_{e\in E, i\in e} \frac{|e|-1}{\Delta(e)}\right)\left(\sum_{e\in E, j\in e} \frac{|e|-1}{\Delta(e)}\right)}{\sum_{e\in E} \frac{|e|(|e|-1)}{\Delta(e)}} \right) x_{ij} \tag{5a}$$

$$\text{s.t.} \quad x_{ij} + x_{jk} - x_{ik} \le 1, \qquad \forall i, j, k = 1, \dots, n,\ i < j < k, \tag{5b}$$

$$x_{ij} - x_{jk} + x_{ik} \le 1, \qquad \forall i, j, k = 1, \dots, n,\ i < j < k, \tag{5c}$$

$$-x_{ij} + x_{jk} + x_{ik} \le 1, \qquad \forall i, j, k = 1, \dots, n,\ i < j < k, \tag{5d}$$

$$x_{ij} \in \{0, 1\}, \qquad i, j = 1, \dots, n,\ i < j. \tag{5e}$$

In this formulation, it is only necessary to impose that the binary variables x_{ij} guarantee the triangle inequalities defined by (5b)–(5d). That is, if $i, j \in C_s$ and $j, k \in C_s$ for some s, then we must have $i, k \in C_s$. There is no need to impose additional constraints, such as the connectivity of C_s, as it has been proven that optimal partitions are composed of connected subsets in [7].

Optimizing the hypergraph modularity (4) is more complex. The potential maximum number of communities to partition the vertex set V, with $|V| = n$, is n. Therefore, we assume that V is always partitioned into n communities $\{C_1, \dots, C_n\}$, some of them can be empty. The partition is defined by the variables:

$$z_{is} = \begin{cases} 1, & \text{if node } i \text{ belongs to community } s, \\ 0, & \text{otherwise.} \end{cases}$$

If C_s is a non-empty community, then s is defined by the highest index of the nodes that belongs to it, i.e., if $|C_s| > 0$ then $\max_{i\in C_s} i = s$. So, we can define the z_{is} variables for any $i, s = 1, \dots, n$, such that $i \le s$. In this way, symmetric solutions of a MILP problem are removed, as can be seen in [28,48,55,56].

Since the hypergraph modularity (4) depends on the number of internal hyperedges and the adjacent degree of each community, the following variables are needed:

$$d_{ls} = \begin{cases} 1, & \text{if the adjacency degree of community } s \text{ is equal to } l, \\ 0, & \text{otherwise.} \end{cases}$$

$$h_{es} = \begin{cases} 1, & \text{if } e \text{ is an internal hyperedge of community } s, \\ 0, & \text{otherwise.} \end{cases}$$

Variables d_{ls} are defined for any $s = 1, \dots, n$, and $l = \delta_s, \dots, \delta - \sum_{i=s+1}^{n} \delta_i$. Variables h_{es} are defined for any $s = 1, \dots, n$ and $e \in E$. Finally, continuous non-negative variables h_{es}^+ are needed to model the excess of an internal hyperedge with respect to γ_e, and they are defined for any $s = 1, \dots, n$ and $e \in E$.

The maximum of Eq. (4) can be obtained by the following MILP problem:

$$(F_{z,\gamma}^2): \quad \max \quad \frac{1}{m} \sum_{s=1}^{n} \sum_{e\in E} \left(h_{es} + h_{es}^+ - \sum_{l=\delta_s}^{\delta - \sum_{i=s+1}^{n} \delta_i} \left(\sum_{t=\max\{\gamma_e, |e|+l-\delta\}}^{\min\{|e|, l\}} (t - \gamma_e + 1) \frac{\binom{l}{t}\binom{\delta-l}{|e|-t}}{\binom{\delta}{|e|}} \right) d_{ls} \right) \tag{6a}$$

$$\text{s.t.} \quad \sum_{s=i}^{n} z_{is} = 1, \qquad \forall i = 1, \dots, n, \tag{6b}$$

$$z_{is} \le z_{ss}, \qquad \forall s = 1, \dots, n,\ i = 1, \dots, s, \tag{6c}$$

$$\sum_{l=\delta_s}^{\delta - \sum_{i=s+1}^{n} \delta_i} d_{ls} = z_{ss}, \qquad \forall s = 1, \dots, n, \tag{6d}$$

$$\sum_{l=\delta_s}^{\delta - \sum_{i=s+1}^{n} \delta_i} l d_{ls} = \sum_{i=1}^{s} \delta_i z_{is}, \qquad \forall s = 1, \dots, n, \tag{6e}$$

$$\gamma_e h_{es} + h_{es}^+ \le \sum_{\substack{i=1 \\ i\in e}}^{s} z_{is}, \qquad \forall s = 1, \dots, n,\ e \in E, \tag{6f}$$

$$h_{es}^+ \le h_{es}(|e| - \gamma_e), \qquad \forall s = 1, \dots, n,\ e \in E, \tag{6g}$$

$$z_{is} \in \{0, 1\}, \qquad \forall s = 1, \dots, n,\ i = 1, \dots, s, \tag{6h}$$

$$d_{ls} \in \{0, 1\}, \qquad \forall s = 1, \dots, n,\ l = \delta_s, \dots, \delta - \sum_{i=s+1}^{n} \delta_i, \tag{6i}$$

$$h_{es} \in \{0, 1\}, \qquad \forall s = 1, \dots, n,\ e \in E, \tag{6j}$$

$$h_{es}^+ \ge 0, \qquad \forall s = 1, \dots, n,\ e \in E. \tag{6k}$$

Objective function (6a) is equivalent to hypergraph modularity (4), since $h_{es} + h_{es}^+$ is equal to the excess of an internal hyperedge with respect to γ_e, and d_{ls} is multiplied for the exact expected weighted sum of the internal hyperedges, if the total adjacency degree of the community is equal to l. Constraints (6b)–(6c) guarantee that each node belongs to one and only one community, that a community s is non-empty if and only if the node s belongs to it, and that s is also the maximum index of the community C_s. Then, equalities (6d)–(6e) impose that each non-empty community is assigned one and only one adjacency degree and this has to be equal to the sum of adjacency degrees of its nodes. Finally, the family of inequalities (6f) ensures that a hyperedge e is internal to a community s if and only if its number of nodes belonging to the community is greater or equal to γ_e. All variables

are defined as binary by (6h)–(6j). To reduce the problem size, the h-variables can be preprocessed before running the optimization: given a hyperedge e and a community s, if $|\{i \in e : i \le s\}| < \gamma_e$, then $h_{es} = 0$ and $h^+_{es} = 0$, because e cannot be an internal hyperedge of community s.

For the special case that $\gamma = 1$, that is, hyperedges are internal only if all their nodes are inside a community, the problem $(F^2_{z,\gamma})$ can be simplified. All the h variables above are replaced by the following:

$$h_e = \begin{cases} 1, & \text{if } e \text{ is an internal hyperedge,} \\ 0, & \text{otherwise,} \end{cases}$$

to obtain the formulation:

$$(F^2_{z,1}): \quad \max \quad \frac{1}{m}\left(\sum_{e\in E} h_e - \sum_{s=1}^{n}\sum_{l=1}^{\delta-\sum_{i=s+1}^{n}\delta_i}\left(\sum_{e\in E}\sum_{t=|e|}^{\min\{|e|,l\}} \frac{\binom{l}{t}\binom{\delta-l}{|e|-t}}{\binom{\delta}{|e|}}\right) d_{ls}\right) \tag{7a}$$

s.t. (6b)–(6e), (6h)–(6j).,

$$h_e \le 1 - z_{is}, \qquad \forall e \in E,\ s = 1,\dots,n,\ \max_{i'\in e} i' > s,\ i \in e,\ i \le s, \tag{7b}$$

$$h_e \le 1 + z_{js} - z_{is}, \qquad \forall e \in E,\ s = 1,\dots,n,\ \max_{i'\in e} i' \le s,\ i,j \in e,\ i \ne j. \tag{7c}$$

In this special case, if there are two nodes from a hyperedge e that belong to different communities then e cannot be internal, due to constraints (7b)–(7c).

It is worth noting that we can improve formulation $(F^2_{z,\gamma})$ applied to some special case. For example, in the simple case of 3-uniform hypergraph, where every hyperedge contains exactly three vertices, formulation $(F^2_{z,\gamma})$ reduces to:

$$(F^3_{z,\gamma}): \quad \max \quad \frac{1}{m}\sum_{s=1}^{n}\sum_{\substack{e\in E\\|e|=3}}\left(h_{es} + h^+_{es} - \sum_{l=\delta_s}^{\delta-\sum_{i=s+1}^{n}\delta_i}\left(\sum_{t=\max\{\gamma_e,3+l-\delta\}}^{\min\{3,l\}} (t-\gamma_e+1)\frac{\binom{l}{t}\binom{\delta-l}{3-t}}{\binom{\delta}{3}}\right) d_{ls}\right) \tag{8a}$$

s.t. (6b)–(6e),(6h)–(6i).,

$$\gamma_{(i,j,k)}h_{(i,j,k)s} + h^+_{(i,j,k)s} \le z_{is} + z_{js} + z_{ks}, \quad \forall i,j,k,s = 1,\dots,n,\ i<j<k\le s, \tag{8b}$$

$$\gamma_{(i,j,k)}h_{(i,j,k)s} + h^+_{(i,j,k)s} \le z_{is} + z_{js}, \qquad \forall i,j,k,s = 1,\dots,n,\ i<j\le s<k, \tag{8c}$$

$$\gamma_{(i,j,k)}h_{(i,j,k)s} + h^+_{(i,j,k)s} \le z_{is}, \qquad \forall i,j,k,s = 1,\dots,n,\ i\le s<j<k, \tag{8d}$$

$$\gamma_{(i,j,k)}h_{(i,j,k)s} + h^+_{(i,j,k)s} = 0, \qquad \forall i,j,k,s = 1,\dots,n,\ s<i<j<k, \tag{8e}$$

$$h^+_{(i,j,k)s} \le h_{(i,j,k)s}(3-\gamma_{(i,j,k)}), \qquad \forall i,j,k,s = 1,\dots,n,\ i<j<k, \tag{8f}$$

$$h_{(i,j,k)s} \in \{0,1\}, \qquad \forall i,j,k,s = 1,\dots,n,\ i<j<k, \tag{8g}$$

$$h^+_{(i,j,k)s} \ge 0, \qquad \forall i,j,k,s = 1,\dots,n,\ i<j<k. \tag{8h}$$

This new formulation leads us to conclude that it is possible to fix variable values of formulation $(F^2_{z,\gamma})$ with a very simple preprocessing. As mentioned before, we can fix variables $h_{es} = 0$ and $h^+_{es} = 0$, whenever $U_{es} = |\{i \in e : i \le s\}| < \gamma_e$, since in this case e is not internal to the community s. In addition, if $U_{es} \ge \gamma_e$, inequalities (6g) and (8f) can be strengthened by:

$$h^+_{es} \le h_{es}(U_{es} - \gamma_e), \tag{9}$$

since U_{es} is the maximum number of nodes that can belong to hyperedge e and community s.

This preprocessing may reduce computational times, however, the scalability limitations are unavoidable due to the NP-hardness nature of the problem. Indeed, most papers about hypergraph clustering focuses their numerical contributions to heuristic algorithms: they have the advantage of providing feasible solutions in a reasonable time, at the cost of being less accurate than exact methods. A Louvain-like clustering algorithm is proposed by Kamiński et al. [52]. Here, we tailor a simple heuristic based on local search that can be found in [28,29,57] and that can be easily extended to optimize hypergraph modularity. We will assess the quality of the local search heuristic through applications of hypergraph modularity to both random networks and real data.

4. Computational tests

In this section, we analyse how the modularity Eqs. (3) and (4) and their optimization models can recover the community structure of hypergraphs. Moreover, the performance of our modularity (4) and the one proposed by Kamiński et al. [52] is compared. The following experimental design is common in network analysis, see [29,40,58].

Following the same logic as [40] and [59], we generated hypergraphs with a hidden community structure characterized by the following parameters:

- n: the number of nodes.
- m: the number of hyperedges.
- n_c: the maximum number of communities.
- L: the maximum hyperedge size.

The community structure of a hypergraph can be easy or hard to recover, depending on the following parameters:

- $1-\mu$: fraction of hyperedges that are internal.
- $1-\xi$: fraction of nodes from an internal hyperedge that belong to the same community.

The algorithm to generate random hypergraphs is an adaption of the h–ABCD model proposed by Kamiński et al. [59], whose code can be found in https://github.com/bkamins/ABCDHypergraphGenerator.jl. The noise parameter $1-\xi$ is related to the parameter γ that is used in model $(F^2_{z,\gamma})$ to determine an internal arc, therefore that model has been tested with $\gamma = 1-\xi$. This approach provides some advantage to our algorithm against the remaining methods since this information is normally unknown. However, the aim of this section is to show that if the γ parameter is properly chosen, then formulation $(F^2_{z,\gamma})$ is able to outperforms any other model. Indeed, even if we select $\gamma = 1-\xi$, there may be a better γ value to detect the hidden community structure. For this reason, a methodology to select a good γ value is provided in the following section.

On the other hand, model (F^1_x) has been solved for all values: $\Delta(e) = \{1, |e|, |e|-1, \frac{|e|(|e|-1)}{2}\}$. These values include our suggestion $\Delta(e) = \frac{|e|(|e|-1)}{2}$ and all the other parameters suggested by the literature to reduce hypergraphs clustering to a graph problem. In the results below, we report only the best solution from graph reduction.

The quality of the community structure obtained by each method is measured comparing the outcome with the true community structure, that is known by construction. The statistics that we consider are:

- Normalized Mutual Information (NMI), presented in [60].
- Omega index (OI), presented in [61].

The reason for testing two different metrics is due to the some known misbehaviour of the NMI statistic as a metric for evaluating clustering and community detection algorithms. This biased behaviour, due to factors such as the number of communities, has been formally proved by Mahmoudi and Jemielniak [62].

Table 1
Computational results with $n = 10, 20$.

Formulation					F^1_x		$F^2_{z,\gamma}$		Formulation					F^1_x		$F^2_{z,\gamma}$	
n	m	μ	ξ	L	NMI	OI	NMI	OI	n	m	μ	ξ	L	NMI	OI	NMI	OI
10	50	0	0	3	**1**	**1**	**1**	**1**	20	50	0	0	3	**0.97**	**0.96**	**0.97**	0.95
				5	**1**	**1**	**1**	**1**					5	**1**	**1**	**1**	**1**
			0.2	3	**1**	**1**	**1**	**1**				0.2	3	**0.95**	**0.92**	**0.95**	0.91
				5	0.98	0.96	**1**	**1**					5	**1**	**1**	**1**	**1**
			0.4	3	0.95	0.91	**0.96**	**0.94**				0.4	3	0.86	0.80	**0.92**	**0.87**
				5	0.96	0.92	**0.98**	**0.94**					5	0.98	0.98	**0.99**	**0.99**
		0.2	0	3	**1**	**1**	0.98	0.96			0.2	0	3	0.88	0.82	**0.90**	**0.85**
				5	0.97	0.96	**1**	**1**					5	**1**	**1**	0.97	0.95
			0.2	3	0.98	0.97	**1**	**1**				0.2	3	0.92	0.87	**0.95**	**0.93**
				5	0.92	0.85	**0.98**	**0.96**					5	**0.96**	**0.93**	0.94	0.88
			0.4	3	0.81	0.67	**0.90**	**0.81**				0.4	3	0.67	0.51	**0.79**	**0.65**
				5	0.95	0.88	**0.96**	**0.92**					5	0.91	0.87	**0.94**	**0.91**
		0.4	0	3	**0.94**	**0.89**	**0.94**	0.87			0.4	0	3	0.75	0.67	**0.77**	**0.69**
				5	0.87	0.80	**1**	**1**					5	0.96	0.95	**0.97**	**0.96**
			0.2	3	0.93	0.87	**0.96**	**0.92**				0.2	3	0.73	0.63	**0.80**	**0.72**
				5	0.83	0.73	**0.92**	**0.87**					5	**0.86**	**0.80**	0.85	0.78
			0.4	3	0.81	0.72	**0.90**	**0.82**				0.4	3	0.43	0.25	**0.46**	**0.27**
				5	0.64	0.48	**0.84**	**0.70**					5	0.75	0.66	**0.80**	**0.70**
	100	0	0	3	**1**	**1**	**1**	**1**		100	0	0	3	**1**	**1**	**1**	**1**
				5	**1**	**1**	**1**	**1**					5	**1**	**1**	**1**	**1**
			0.2	3	**1**	**1**	**1**	**1**				0.2	3	**1**	**1**	**1**	**1**
				5	**1**	**1**	**1**	**1**					5	**1**	**1**	**1**	**1**
			0.4	3	0.98	0.96	**1**	**1**				0.4	3	0.96	0.94	**0.98**	**0.97**
				5	0.98	0.97	**1**	**1**					5	0.98	0.97	**0.99**	**0.98**
		0.2	0	3	**1**	**1**	**1**	**1**			0.2	0	3	**1**	**1**	**1**	**1**
				5	0.96	0.93	**1**	**1**					5	**1**	**1**	**1**	**1**
			0.2	3	**1**	**1**	**1**	**1**				0.2	3	**0.99**	**0.99**	**0.99**	**0.99**
				5	**1**	**1**	**1**	**1**					5	**1**	**1**	**1**	**1**
			0.4	3	0.96	0.93	**1**	**1**				0.4	3	0.89	0.84	**0.92**	**0.89**
				5	0.92	0.85	**1**	**1**					5	0.97	0.95	**1**	**1**
		0.4	0	3	**1**	**1**	**1**	**1**			0.4	0	3	0.93	0.90	**0.94**	**0.91**
				5	0.94	0.90	**1**	**1**					5	**1**	**1**	**1**	**1**
			0.2	3	0.96	0.93	**0.99**	**0.96**				0.2	3	0.91	0.88	**0.92**	**0.89**
				5	0.92	0.87	**1**	**1**					5	**1**	**1**	**1**	**1**
			0.4	3	0.96	0.91	**0.98**	**0.94**				0.4	3	0.61	**0.48**	**0.65**	**0.48**
				5	0.85	0.75	**1**	**1**					5	**0.88**	0.82	**0.88**	**0.85**

As (F^1_x) and $(F^2_{z,\gamma})$ are MILP models formulated to solve NP-hard problems, they could imply extensive computational times to calculate the optimum. Therefore, we cannot test the models using large size hypergraphs. However, and conversely to common practices, we did not rely on heuristic procedures, but we have calculated the *exact* best solutions. As a consequence, our results are a reliable comparison between measures (3) and (4), because the outcomes are not biased by the quality of the heuristics. So, tests are run on instances with sizes $n = \{10, 20, 30\}$, $m = \{50, 100\}$, $L = \{3, 5\}$ and the maximum number of communities $n_c = 5$. Tested parameter values are $\mu = \{0, 0.2, 0.4\}$ and $\xi = \{0, 0.2, 0.4\}$. Algorithms and models have been implemented in Python, and MILP formulations have been solved with Gurobi.

For every combination of parameters, we have solved optimally 50 random instances of the hypergraph modularity problem with both models. Results, reported as averages, can be see in Tables 1 and 2. In most cases, according to both statistics NMI and OI, the formulation $(F^2_{z,\gamma})$ performs better than the best partition calculated by (F^1_x), with varying values of $\Delta(e)$. Indeed, when $n = 10$ and considered 36 parameters combination, $(F^2_{z,\gamma})$ outperformed (F^1_x) in 23 or 22 cases (depending by the index). In only one case, the reverse result has been obtained. For $n = 20$, $(F^2_{z,\gamma})$ obtained the best results in 19 out of the 36 cases. For $n = 30$ the result has been 17 against 8. Moreover, both formulations obtained statistics close to 1, showing that hypergraph modularity is an effective concept to determine the hidden communities.

Grouping the instances further, therefore reporting the same results, but considering the average marginal metrics after pairing n with m, or L, or μ or ξ, results reported in Table 3 are reported. First, it is interesting to observe that the quality of the partition improves as the number of hyperedges increases. We may guess that it happens because the ideal communities are better connected, or well separated, when the number of hyperedges is greater. Something similar happens with the parameter L, because the greater L, the easier is to cover and to connect the nodes that within a community. Conversely, and as could be expected, to increase the noise parameters μ and ξ, affects negatively the quality of the partitions. We outline that parameter ξ has a more negative impact on F^1_x, because in that model a *noising* hyperedge contributes to the definition of a community. Conversely, in the second formulation $F^2_{z,\gamma}$ that *noising* factor is taken into account by γ, and therefore it does not negatively affects the index. As a final summary, for each size n the overall average values are reported in Table 4. As can be seen, we can conclude that the hypergraph modularity Eq. (4) outperforms the projected modularity (3).

Formulation $(F^2_{z,\gamma})$ can be extended to maximize a version of the modularity measure proposed by Kamiński et al. [52]. We refer to the linear hypergraph modularity case with parameter $\tau = 1$ in [52]. For different values of τ, formulation $(F^2_{z,\gamma})$ can be easily modified. For the linear case $\tau = 1$, the hypergraph modularity proposed by Kamiński et al. [52] is equivalent to model $(F^2_{z,\gamma})$ with $\gamma = 0.5$, but with the

Table 2
Computational results with $n = 30$.

Formulation					F^1_x		$F^2_{z,\gamma}$	
n	m	μ	ξ	L	NMI	OI	NMI	OI
30	50	0	0	3	**0.91**	0.86	0.90	**0.87**
				5	**0.99**	**0.98**	**0.99**	**0.98**
			0.2	3	**0.95**	**0.92**	0.94	**0.92**
				5	**0.98**	**0.96**	0.96	0.93
			0.4	3	0.71	0.58	**0.76**	**0.65**
				5	0.89	0.85	**0.96**	**0.93**
		0.2	0	3	**0.83**	**0.76**	0.80	0.72
				5	**0.98**	**0.97**	**0.98**	**0.97**
			0.2	3	0.80	0.71	**0.84**	**0.79**
				5	0.89	0.85	**0.92**	**0.87**
			0.4	3	0.51	0.33	**0.58**	**0.43**
				5	**0.77**	**0.70**	**0.77**	**0.70**
		0.4	0	3	0.51	0.36	**0.54**	**0.42**
				5	**0.73**	**0.67**	0.72	0.65
			0.2	3	0.59	0.45	**0.60**	**0.46**
				5	0.75	0.66	**0.80**	**0.75**
			0.4	3	**0.38**	0.20	**0.38**	**0.23**
				5	0.48	0.38	**0.52**	**0.40**
	100	0	0	3	**0.99**	**0.99**	**0.99**	**0.99**
				5	**1**	**1**	**1**	**1**
			0.2	3	**0.99**	**0.99**	**0.99**	**0.99**
				5	**1**	**1**	**1**	**1**
			0.4	3	0.90	0.88	**0.94**	**0.92**
				5	**0.99**	**0.99**	0.98	0.98
		0.2	0	3	0.97	0.96	**0.98**	**0.97**
				5	**1**	**1**	**1**	**1**
			0.2	3	**0.98**	**0.98**	**0.98**	0.97
				5	**0.99**	**0.99**	**0.99**	**0.99**
			0.4	3	0.69	0.61	**0.74**	**0.65**
				5	0.96	0.95	**0.98**	**0.97**
		0.4	0	3	**0.72**	**0.64**	**0.72**	0.62
				5	0.97	0.97	**0.99**	**0.99**
			0.2	3	0.72	0.62	**0.74**	**0.68**
				5	0.95	**0.95**	**0.96**	**0.95**
			0.4	3	**0.47**	**0.35**	0.45	0.28
				5	0.70	0.65	**0.72**	**0.67**

objective function (6a) replaced by:

$$\frac{1}{m}\sum_{s=1}^{n}\sum_{e\in E}\left(\frac{h_{es}\gamma_e + h^+_{es}}{|e|} - \sum_{l=\delta_s}^{\delta-\sum_{i=s+1}^{n}\delta_i}\left(\sum_{t=\max\{\gamma_e,|e|+l-\delta\}}^{\min\{|e|,l\}}\frac{t}{d}\binom{|e|}{t}\left(\frac{l}{\delta}\right)^t \times\left(1-\frac{l}{\delta}\right)^{|e|-t}\right)d_{ls}\right). \tag{10}$$

In Eq. (10), weights $w(e, C_s)$ are replaced by $\frac{|C_s\cap e|}{|e|}$ and the hypergeometric distribution $H(\delta, \delta_{C_s}, |e|)$ replaced by the binomial distribution $B(|e|, \frac{\delta_{C_s}}{\delta})$. We will refer to this MILP formulation as $(F^{kam}_{z,0.5})$. This alternative model has been tested on the same instances of Tables 1, 2, 3 and 4, above. Results are reported in Tables 5, 6, 7 and 8.

As can be seen, formulation $(F^2_{z,\gamma})$ outperforms $(F^{kam}_{z,0.5})$ in terms of NMI and OI. Only in the case $\xi = 0.4$, the results are similar due to the internal threshold considered by both models. Moreover, in those cases where μ and ξ increase, the actual communities are not clearly separated by $(F^{kam}_{z,0.5})$. Hence, we can conclude that our new modularity function (4) improves the performance of the one proposed in [52] and solves some of its misbehaviour.

5. Local search heuristic algorithm for large scale hypergraphs

In this section, we focus on heuristic algorithms to approximate the hypergraph modularity problem. These methods partially amend the scalability issues posed by the NP-hardness of the problem and they obtain good approximate solutions for large scale data. Specifically, we rely on GRASP [63] and multi-start meta-heuristics [64] to improve the subroutine available in [28,29,57,65] for similar community detection problems.

For a given a random partition of nodes, the best local improvement is selected among changing the membership of a node to another community or to let the node form a new community by itself as a singleton. Local improvements are then repeated until a local optimum has been detected. In this case, a new solution is found by starting the local search from another random partition. We will refer to this heuristic as *Local search for hypergraph modularity* (LSHM).

To test the LSHM algorithm, we followed the same steps of the previous section to generate some random hypergraphs. Then, we compare the LSHM with the Louvain hypergraph clustering heuristic run on model $(F^{kam}_{z,0.5})$, referred to as h-Louvain, and proposed in [52].[1] We considered random hypergraphs with size $n = 50$ paired with $m = \{50, 100\}$, and with size $n = 100$ paired with $m = \{100, 200\}$.

For sake of comparison and checking the robustness of our findings, we report previous metrics NMI and OI, and also the Silhouette score (SIL), see [66]. SIL takes values between -1 and 1, and the higher the score, the better the communities are recognized. To calculate SIL, it is necessary to define the distance between two nodes. We define the distance between two nodes as the length of the shortest between them, considering two nodes as adjacent if they belong to the same hyperedge at least once. So, the distance between two adjacent nodes is equal to 1. Results are summarized in averages in Tables 9 and 10.

The reported metrics support the same conclusions of the comparison using exact methods. That is, model $(F^2_{z,\gamma})$ is better than model $(F^{kam}_{z,0.5})$. In Table 10 the global averages are reported for $n = 50{,}100$. There it can be seen that for the three statistics, NMI, OI and SIL, model $(F^2_{z,\gamma})$ gets the highest values to recognize the hidden communities. In Table 9, the same statistics are reported for problem class, and the superiority of model $(F^2_{z,\gamma})$ is still evident: only in two cases and for the SIL index the model did not get the highest value. However, the h-Louvain is always faster than LSHM. This is due to the structure the algorithms. The h-Louvain starts from a partition where each community is formed by a single node, and then it proceeds by merging first singletons, and then communities in a greedy fashion until no improvement is obtained. The diversification of the method is limited. Conversely, LSHM tries to improve different random partitions, and therefore it allows more computational steps. However, the computational costs are not prohibitive as they never exceed few seconds. In addition, the LSHM computational time highly depends on the number of algorithm multi-starts, which is a parameter that can be controlled. In this case, we run the algorithm 30 times for $n = 50$ and 60 times for $n = 100$.

6. Applications to real data hypergraphs

In the following section we will apply hypergraph modularity to three applications and using real data. These applications are:

- Voting patterns at the United Nations General Assembly (UNGA), [67];
- Morphological classification of zoo animals, [34];
- Multi-items questions in opinion surveys, [39].

The applications have been selected with the purpose of showing the role of the parameter γ in the optimization process and then showing the relevance of exact methods and the LSHM algorithm when the number of nodes is not excessive.

[1] The h-Louvain algorithm Python code can be found in the GitHub repository https://github.com/pawelwm/h-louvain.

Table 3
Marginal values for each one of the 4-factors in the experiment.

Formulation		F_x^1		$F_{z,\gamma}^2$		Formulation		F_x^1		$F_{z,\gamma}^2$	
n	m	NMI	OI	NMI	OI	n	L	NMI	OI	NMI	OI
10	50	0.92	0.87	**0.96**	**0.93**	10	3	0.96	0.93	**0.98**	**0.96**
	100	0.97	0.94	**1**	**0.99**		5	0.93	0.88	**0.98**	**0.97**
20	50	0.87	0.81	**0.89**	**0.83**	20	3	0.86	0.80	**0.88**	**0.83**
	100	0.95	0.93	**0.96**	**0.94**		5	**0.96**	**0.94**	**0.96**	**0.94**
30	50	0.76	0.68	**0.77**	**0.70**	30	3	0.76	0.68	**0.77**	**0.70**
	100	0.89	0.86	**0.90**	**0.87**		5	0.89	0.86	**0.90**	**0.87**
Formulation		F_x^1		$F_{z,\gamma}^2$		Formulation		F_x^1		$F_{z,\gamma}^2$	
n	μ	NMI	OI	NMI	OI	n	ξ	NMI	OI	NMI	OI
10	0	**0.99**	0.98	**0.99**	**0.99**	10	0	0.97	0.96	**0.99**	**0.99**
	0.2	0.96	0.92	**0.98**	**0.97**		0.2	0.96	0.93	**0.99**	**0.98**
	0.4	0.89	0.82	**0.96**	**0.92**		0.4	0.90	0.83	**0.96**	**0.92**
20	0	**0.98**	0.96	**0.98**	**0.97**	20	0	**0.96**	**0.94**	**0.96**	**0.94**
	0.2	0.93	0.90	**0.95**	**0.92**		0.2	0.94	0.92	**0.95**	**0.93**
	0.4	0.82	0.75	**0.84**	**0.77**		0.4	0.82	0.76	**0.86**	**0.80**
30	0	0.94	0.92	**0.95**	**0.93**	30	0	**0.88**	**0.85**	**0.88**	**0.85**
	0.2	0.86	0.82	**0.88**	**0.84**		0.2	0.88	0.84	**0.89**	**0.86**
	0.4	0.66	0.57	**0.68**	**0.59**		0.4	0.71	0.62	**0.73**	**0.65**

Table 4
Summary of the two metrics NMI and OI for the different instance sizes.

Formulation	F_x^1		$F_{z,\gamma}^2$	
n	NMI	OI	NMI	OI
10	0.94	0.91	**0.98**	**0.96**
20	0.91	0.87	**0.92**	**0.89**
30	0.82	0.77	**0.84**	**0.79**

6.1. Voting at the united nations general assembly

At the UNGA, world nations vote in favour (yes), against (no), or abstain on a given issue. Most issues are approved by unanimity, but there are some that are disputed, they are approved (or rejected) only by the simple majority of the countries. When this is the case, the group of nations that voted in the same way reveals a hidden pattern of common preferences, and the possibility of coordinating their behaviour in similar questions. In [67], a simple graph has been obtained with the following procedure. Every nation is a graph node, and then, for a given issue, there is an arc between a nations pairs if they voted in the same way, that is, yes-yes, no-no, or abstain-abstain. Then the procedure is repeated for all the disputed issues to obtain a non-oriented graph with multiple arcs between nodes. Then modularity maximization is applied to reveal the graph community structure. The procedure by which we applied hypergraph modularity to voting is as follows. We considered the disputed votes for the years 2021–2022, as recorded in [68]. They include 193 countries that voted on 165 disputed issues. For every issue, we considered three hyperedges: one is composed of the nations that voted yes, another of nations that voted no, and then of the nations that abstained. We have used the LSHM heuristic to solve hypergraph modularity with $\gamma = \{0.8, 0.6\}$. These results are illustrated in Fig. 6. In the figure, the projected simple graph of the original hypergraph is represented.

We can observe how the structures change depending on the value of γ. For large values of γ, we are more restrictive considering whether a hyperedge is internal or not, so many hyperedges partially contained in communities are not taken into account. For smaller values of γ, we are less strict and most of the hyperedges that are partially contained in some community are considered internal. Although in many cases the parameter γ is chosen depending on how strict we want to be, there are situations where we do not know how to fix the parameter and we have to choose which is the best community structure. Here we must be more precise about the findings of the three γ.

Determine what is the best value of γ depends on the purpose of the analysis and cannot be determined by purely mathematical arguments (see also the next discussion on the zoo animals hypergraph). For example, the research purpose could be a broad, classification with just two clusters, or a fine-tuned classification, with many small clusters. However, here we give some suggestion. Following [67], in which they tried to fix the resolution parameter in modularity optimization, one can choose the γ among the most robust values: that is, a value that when perturbed by some fraction, still it provides the same community structure. A second possibility is to resort to another auxiliary metric to establish what is the better partition among the maximizers of hypergraph modularity. In the following examples, we will use the projected modularity (3) to compare those community structures. The second methodology consists of using a preprocessing phase that allows us to obtain a primary community structure. Once we obtain this first structure, we can try to estimate the parameter γ based on the proportion of hyperedges that are found inside a community.

For the UNGA voting data, the highest projected modularity is obtained for $\gamma = 0.8$. The two communities are depicted in Fig. 7. One can distinguish two important cores. In the first community, it is possible to detect the group formed by emerging countries called BRICS: Brazil, Russia, India, China, South Africa, Egypt, Ethiopia, Indonesia, Iran and the United Arab Emirates. In the second community, we can find the G7 group: Canada, France, Germany, Italy, Japan, the United Kingdom, the United States and the European Union.

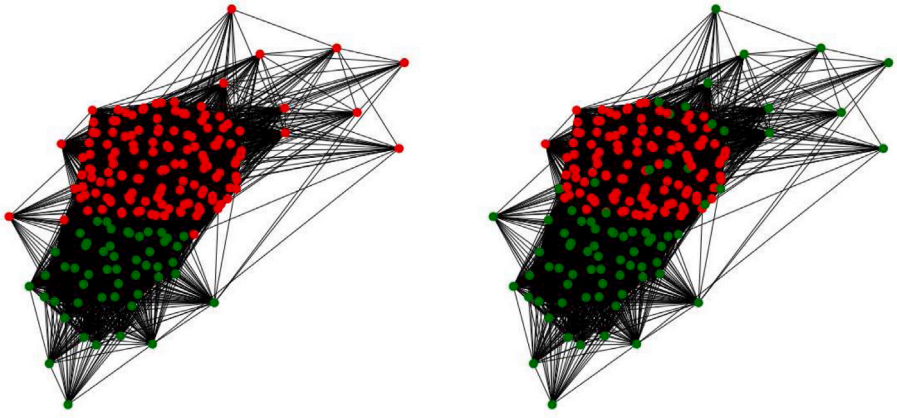
6.2. The classification of zoo animals

The second database that we consider has been previously analysed in [34]. It consists of 101 species of animals from which the value of 16 attributes has been collected: hair, feathers, eggs, milk, airborne, aquatic, predator, toothed, backbone, breathes, venomous, fins, legs, tail, domestic and catsize. All the variables are binary, excepted the number of legs that is an integer. Animals are the nodes of the hypergraph, while there is an hyperedge on the nodes for which an attribute has the same modality, that is, for a binary variable an hyperedge corresponds to the 1-class and another hyperedge for the 0-class. The interest of this application relies on the fact that it has a ground-truth classification: it is known whether animals are Mammal, Bird, Reptile, Fish, Amphibian, Bug or Invertebrate.

We let $\gamma = \{1, 0.8, 0.5, 0.2\}$. Note that in this application we allowed $\gamma = 0.2$, so that an hyperedge is useful to define more than a cluster. For the different γ's, the community structures are solved by LSHM and illustrated in Fig. 8. Note that for the highest values of γ the partition

Table 5
Computational results with $n = 10, 20$.

Formulation					$F^{kam}_{z,0.5}$		$F^{2}_{z,\gamma}$		Formulation					$F^{kam}_{z,0.5}$		$F^{2}_{z,\gamma}$	
n	m	μ	ξ	L	NMI	OI	NMI	OI	n	m	μ	ξ	L	NMI	OI	NMI	OI
10	50	0	0	3	**1**	**1**	**1**	**1**	20	50	0	0	3	0.90	0.81	**0.97**	**0.95**
				5	0.97	0.96	**1**	**1**					5	0.93	0.88	**1**	**1**
			0.2	3	0.96	0.94	**1**	**1**				0.2	3	0.87	0.78	**0.95**	**0.91**
				5	0.91	0.85	**1**	**1**					5	0.93	0.90	**1**	**1**
			0.4	3	**1**	**1**	0.96	0.94				0.4	3	0.81	0.70	**0.92**	**0.87**
				5	0.91	0.89	**0.98**	**0.94**					5	0.98	0.97	**0.99**	**0.99**
		0.2	0	3	0.87	0.80	**0.98**	**0.96**			0.2	0	3	0.84	0.71	**0.90**	**0.85**
				5	0.91	0.86	**1**	**1**					5	0.90	0.82	**0.97**	**0.95**
			0.2	3	0.90	0.83	**1**	**1**				0.2	3	0.80	0.65	**0.95**	**0.93**
				5	0.92	0.86	**0.98**	**0.96**					5	0.80	0.67	**0.94**	**0.88**
			0.4	3	**0.91**	**0.86**	0.90	0.81				0.4	3	0.72	0.56	**0.79**	**0.65**
				5	0.89	0.85	**0.96**	**0.92**					5	0.80	0.71	**0.94**	**0.91**
		0.4	0	3	0.78	0.66	**0.94**	**0.87**			0.4	0	3	0.72	0.59	**0.77**	**0.69**
				5	0.92	0.87	**1**	**1**					5	0.71	0.59	**0.97**	**0.96**
			0.2	3	0.76	0.66	**0.96**	**0.92**				0.2	3	0.46	0.22	**0.80**	**0.72**
				5	0.86	0.78	**0.92**	**0.87**					5	0.70	0.58	**0.85**	**0.78**
			0.4	3	**0.96**	**0.93**	0.90	0.82				0.4	3	**0.56**	**0.29**	0.46	0.27
				5	**0.87**	**0.84**	0.84	0.70					5	0.59	0.37	**0.80**	**0.70**
	100	0	0	3	**1**	**1**	**1**	**1**		100	0	0	3	**1**	**1**	**1**	**1**
				5	0.94	0.93	**1**	**1**					5	0.99	0.98	**1**	**1**
			0.2	3	**1**	**1**	**1**	**1**				0.2	3	**1**	**1**	**1**	**1**
				5	0.98	0.96	**1**	**1**					5	0.96	0.95	**1**	**1**
			0.4	3	0.96	0.94	**1**	**1**				0.4	3	0.95	0.93	**0.98**	**0.97**
				5	**1**	**1**	**1**	**1**					5	**1**	**1**	0.99	0.98
		0.2	0	3	**1**	**1**	**1**	**1**			0.2	0	3	0.93	0.85	**1**	**1**
				5	**1**	**1**	**1**	**1**					5	0.98	0.97	**1**	**1**
			0.2	3	**1**	**1**	**1**	**1**				0.2	3	0.89	0.82	**0.99**	**0.99**
				5	**1**	**1**	**1**	**1**					5	0.89	0.86	**1**	**1**
			0.4	3	**1**	**1**	**1**	**1**				0.4	3	0.91	0.86	**0.92**	**0.89**
				5	**1**	**1**	**1**	**1**					5	0.97	0.96	**1**	**1**
		0.4	0	3	0.92	0.89	**1**	**1**			0.4	0	3	0.70	0.54	**0.94**	**0.91**
				5	0.98	0.96	**1**	**1**					5	0.87	0.80	**1**	**1**
			0.2	3	0.96	0.93	**0.99**	**0.96**				0.2	3	0.83	0.74	**0.92**	**0.89**
				5	0.88	0.81	**1**	**1**					5	0.87	0.78	**1**	**1**
			0.4	3	0.85	0.84	**0.98**	**0.94**				0.4	3	**0.73**	**0.59**	0.65	0.48
				5	0.98	0.96	**1**	**1**					5	**0.88**	0.83	**0.88**	**0.85**



(a) γ=0.8 (b) γ=0.6

Fig. 6. Communities of countries in UNGA. For all meaningful γ the method always finds two similar communities.

is not accurate as we wished, as only two or three clusters are obtained. However, relying on the auxiliary projected modularity, the best community structure is obtained for $\gamma = 0.2$. Specifically, the detected communities are reported in Table 11. We observe 5 communities of which, community 1 consists in non-predator Mammals, community 2 in predator Mammals, community 3 in Fishes, Reptiles and Amphibians, community 4 in Birds and community 5 in Bugs and Invertebrates. Even though this classification is different from the ground-truth, still

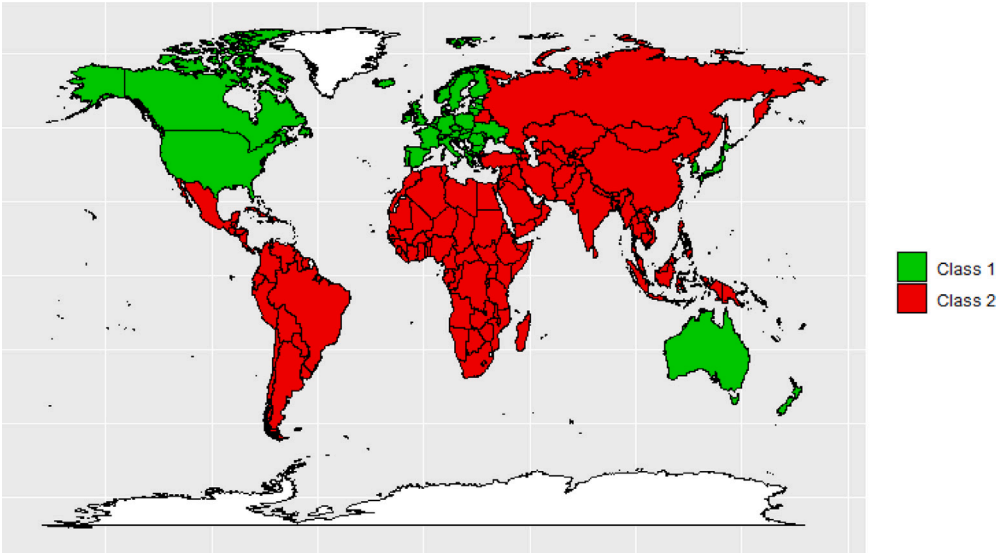


Fig. 7. Clusters world map representation.

Table 6
Computational results with $n = 30$.

Formulation					$F^{kam}_{z,0.5}$		$F^{2}_{z,\gamma}$	
n	m	μ	ξ	L	NMI	OI	NMI	OI
30	50	0	0	3	0.84	0.70	**0.90**	**0.87**
				5	0.93	0.88	**0.99**	**0.98**
			0.2	3	0.81	0.61	**0.94**	**0.92**
				5	0.93	0.89	**0.96**	**0.93**
			0.4	3	**0.78**	0.62	0.76	**0.65**
				5	0.90	0.85	**0.96**	**0.93**
		0.2	0	3	0.70	0.47	**0.80**	**0.72**
				5	0.70	0.55	**0.98**	**0.97**
			0.2	3	0.70	0.52	**0.84**	**0.79**
				5	0.75	0.62	**0.92**	**0.87**
			0.4	3	**0.59**	0.36	0.58	**0.43**
				5	**0.77**	0.67	**0.77**	**0.70**
		0.4	0	3	**0.59**	0.36	0.54	**0.42**
				5	0.61	0.43	**0.72**	**0.65**
			0.2	3	0.57	0.34	**0.60**	**0.46**
				5	0.65	0.47	**0.80**	**0.75**
			0.4	3	**0.47**	0.20	0.38	**0.23**
				5	**0.54**	0.36	0.52	**0.40**
	100	0	0	3	0.94	0.88	**0.99**	**0.99**
				5	0.98	0.97	**1**	**1**
			0.2	3	0.93	0.86	**0.99**	**0.99**
				5	0.99	0.99	**1**	**1**
			0.4	3	0.92	0.87	**0.94**	**0.92**
				5	**0.99**	**0.99**	0.98	0.98
		0.2	0	3	0.79	0.71	**0.98**	**0.97**
				5	0.88	0.84	**1**	**1**
			0.2	3	0.82	0.71	**0.98**	**0.97**
				5	0.89	0.85	**0.99**	**0.99**
			0.4	3	**0.74**	0.57	**0.74**	**0.65**
				5	0.92	0.88	**0.98**	**0.97**
		0.4	0	3	0.70	0.54	**0.72**	**0.62**
				5	0.80	0.75	**0.99**	**0.99**
			0.2	3	0.66	0.49	**0.74**	**0.68**
				5	0.78	0.68	**0.96**	**0.95**
			0.4	3	**0.69**	**0.54**	0.45	0.28
				5	**0.74**	0.64	0.72	**0.67**

common properties shared by the community members can be found. For example, community 3 is formed by animals that are aquatic and community 5 by animals that lack a skeleton.

6.3. Applications to survey data

In the following section, we apply hypergraph modularity to find the hidden clusters of the public opinions emerging from survey data. In some social surveys, there are questions in which the respondents can elicit two or more items from a list. As discussed in [39], the Eurobarometer is one example of these surveys. The Standard Eurobarometer includes questions about what are the two most important problems of the country. To that question, respondents can elicit at most two items from a closed list containing 17 terms, such as Terrorism, Unemployment, and so on. As described in [39], the answers to that question can be modelled as a graph, in which nodes correspond to the items of the list, and there is an arc from item i and j for every respondent that answered eliciting both i and j (note that multiple arcs and loops are allowed). The resulting graph, called the Items Graph, can be analysed using the network statistics and methodologies, such as modularity and community detection, and it has been found that network methods are much more informative than the alternative standard statistics, like k-means and principal component analysis. Indeed, modularity maximization found hidden opinion groups that could not be detected by k-means clustering or principal component analysis.

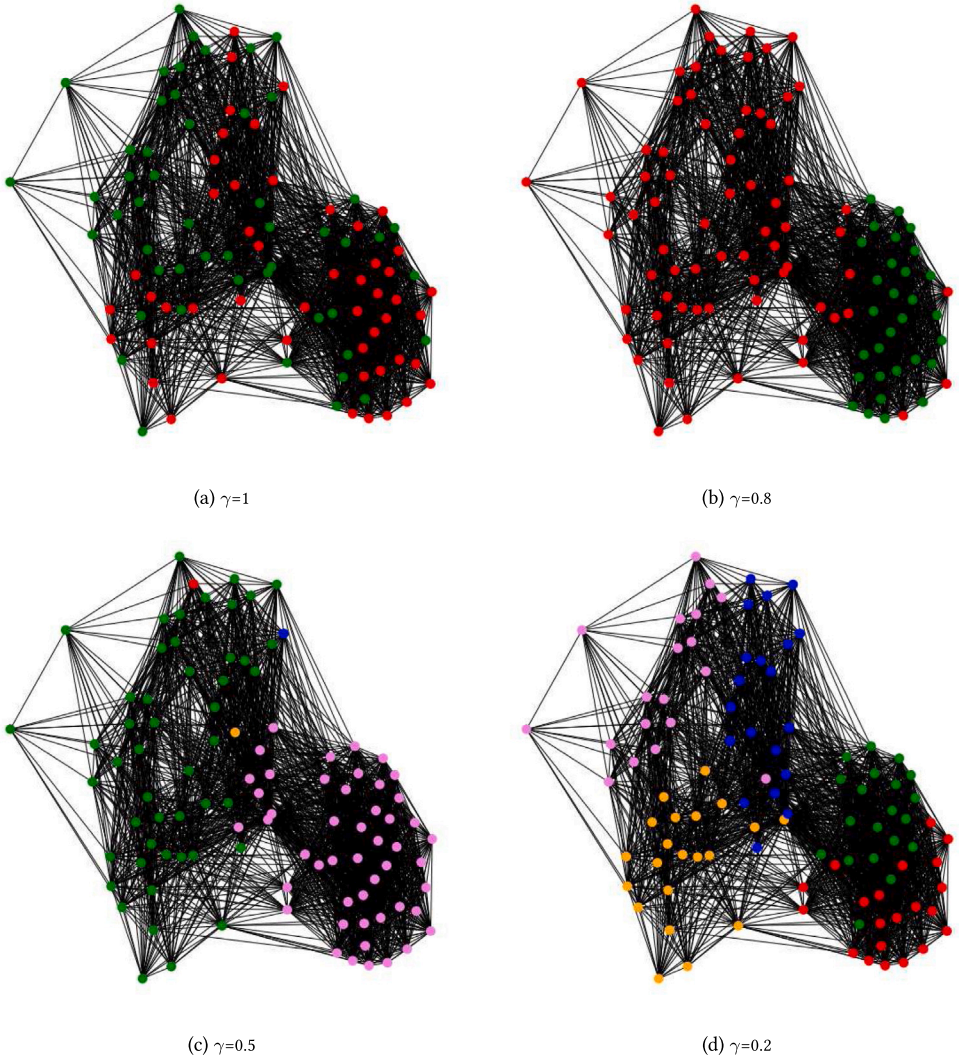
In the following, we are applying modularity clustering to survey questions to which respondents can elicit more than two items. In the first example, coming from the Eurobarometer, respondents may elicit up to four items, chosen from a closed list. In the second example, respondents may answer up to three items, that can be chosen from a list and also freely pointed by respondents (forming an open list of items). We will show how hypergraph modularity can be applied to both settings and uncover hidden data patterns.

A survey question in which respondents can elicit three (or more) number of items can be modelled as a hypergraph, in which nodes correspond to the list items, and there is a hyperedge between items i, j and k for every answer eliciting the three items i, j, k. For example, we considered the Eurobarometer ZA7997, see [69], held in June 2023. In that survey, there is a question in which respondents can elicit at most four items. The question is:

On which of the following areas would you like the recovery plan NextGenerationEU to be spent in priority?

Table 7
Marginal values for each one of the 4-factors in the experiment.

Formulation		$F^{kam}_{z,0.5}$		$F^{2}_{z,\gamma}$		Formulation		$F^{kam}_{z,0.5}$		$F^{2}_{z,\gamma}$	
n	m	NMI	OI	NMI	OI	n	L	NMI	OI	NMI	OI
10	50	0.90	0.86	**0.96**	**0.93**	10	3	0.93	0.90	**0.98**	**0.96**
	100	0.97	0.96	**1**	**0.99**		5	0.94	0.91	**0.98**	**0.97**
20	50	0.78	0.65	**0.89**	**0.83**	20	3	0.81	0.70	**0.88**	**0.83**
	100	0.91	0.86	**0.96**	**0.94**		5	0.88	0.81	**0.96**	**0.94**
30	50	0.71	0.55	**0.77**	**0.70**	30	3	0.74	0.58	**0.77**	**0.70**
	100	0.84	0.76	**0.90**	**0.87**		5	0.82	0.74	**0.90**	**0.87**
Formulation		$F^{kam}_{z,0.5}$		$F^{2}_{z,\gamma}$		Formulation		$F^{kam}_{z,0.5}$		$F^{2}_{z,\gamma}$	
n	μ	NMI	OI	NMI	OI	n	ξ	NMI	OI	NMI	OI
10	0	0.97	0.95	**0.99**	**0.99**	10	0	0.94	0.91	**0.99**	**0.99**
	0.2	0.95	0.92	**0.98**	**0.97**		0.2	0.93	0.89	**0.99**	**0.98**
	0.4	0.89	0.85	**0.96**	**0.92**		0.4	0.94	**0.93**	**0.96**	0.92
20	0	0.94	0.91	**0.98**	**0.97**	20	0	0.87	0.79	**0.96**	**0.94**
	0.2	0.87	0.79	**0.95**	**0.92**		0.2	0.83	0.75	**0.95**	**0.93**
	0.4	0.72	0.58	**0.84**	**0.77**		0.4	0.83	0.73	**0.86**	**0.80**
30	0	0.91	0.84	**0.95**	**0.93**	30	0	0.79	0.67	**0.88**	**0.85**
	0.2	0.77	0.65	**0.88**	**0.84**		0.2	0.79	0.67	**0.89**	**0.86**
	0.4	0.65	0.48	**0.68**	**0.59**		0.4	**0.75**	0.63	0.73	**0.65**



(a) γ=1 (b) γ=0.8

(c) γ=0.5 (d) γ=0.2

Fig. 8. Communities resulting from the 101 animal database. Each community appears with nodes in a different colour. For instance, for $\gamma = 0.2$ five homogeneous communities appear: (1) non-predator Mammals, (2) predator Mammals, (3) Fishes, Reptiles and Amphibians, (4) Birds and (5) Bugs and Invertebrates.

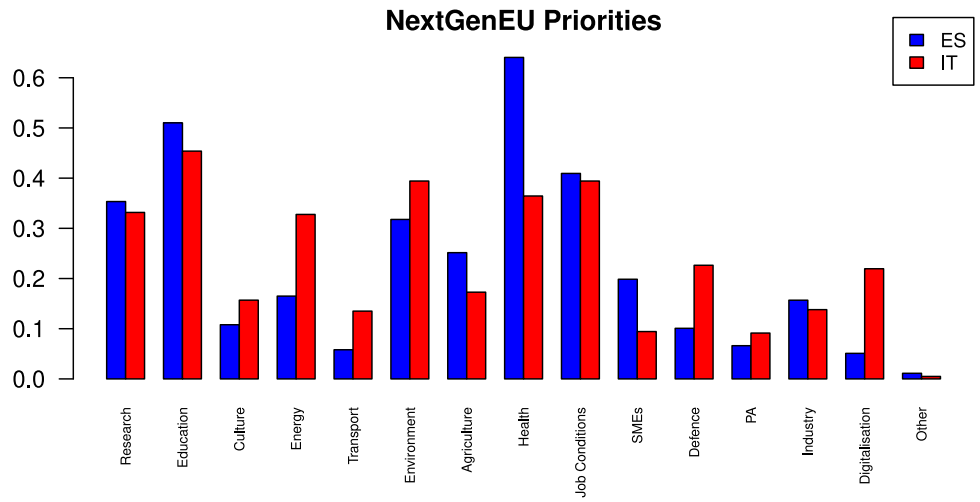
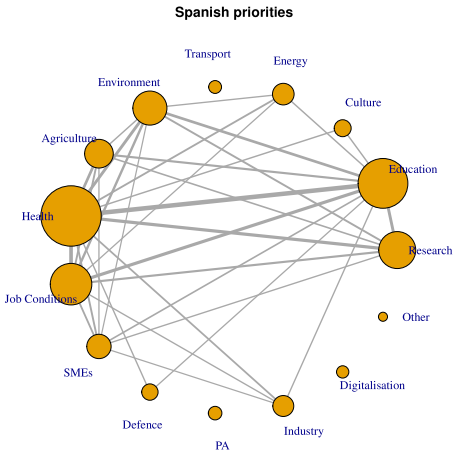
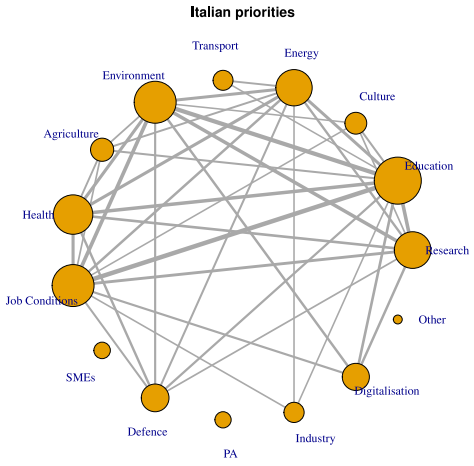


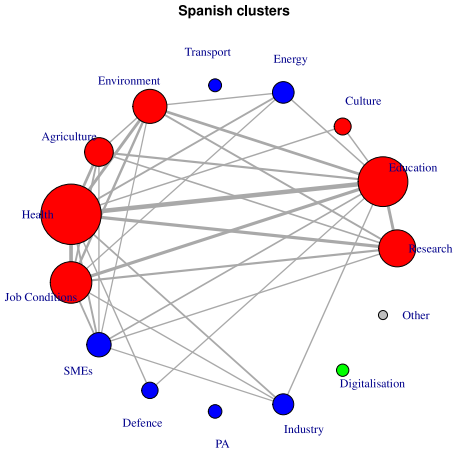
Fig. 9. Spending priorities for Spain and Italy.



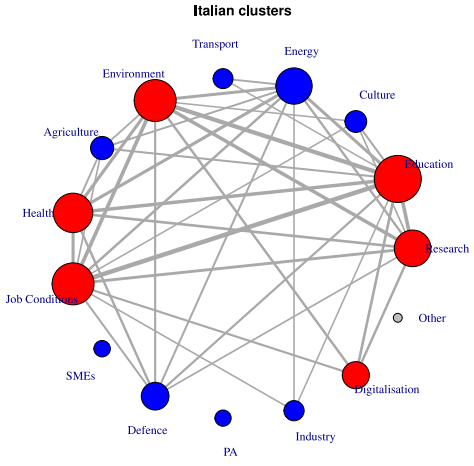
(a) Item graph of the Spanish priorities.



(b) Item graph of the Italian priorities

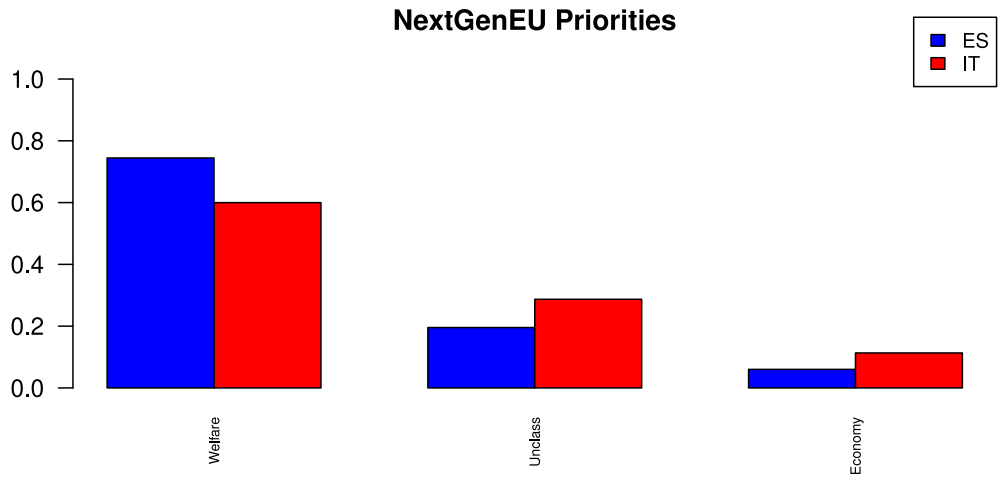


(c) Spanish priorities clusters.

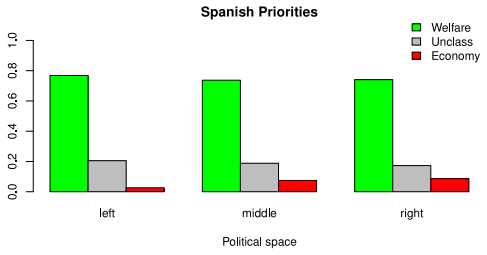


(d) Italian priorities clusters.

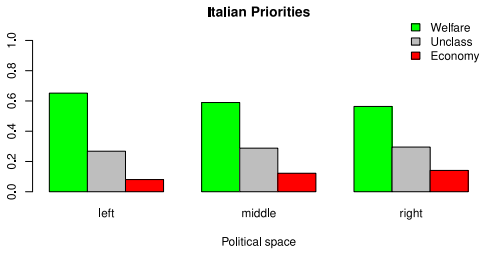
Fig. 10. Hypergraph clustering of the Spanish and Italian priorities.



(a) Spending priorities recorded for Spain and Italy.



(b) Spanish priorities depending on political position.



(c) Italian priorities depending on political position.

Fig. 11. Frecuencies of the clusters organized by country and political position.

Table 8
Summary of the two metrics NMI and OI for the different instance sizes.

Formulation	$F^{kam}_{z,0.5}$		$F^{2}_{z,\gamma}$	
n	NMI	OI	NMI	OI
10	0.94	0.91	**0.98**	**0.96**
20	0.84	0.76	**0.92**	**0.89**
30	0.78	0.66	**0.84**	**0.79**

The list proposed to respondents is:

- Scientific research and development
- Youth, education and training
- Culture and media
- Energy
- Transport
- Climate change and environmental protection
- Agriculture and rural development
- Health
- Improvement of working conditions of EU citizens
- Support to SMEs
- Defence and security
- Public Administration
- Industry
- Digitalization of economy and society
- Other
- None
- Don't know

In a preliminary analysis, priorities could be grouped together into affinity areas. For example, we could expect that one group could consist of the economic priorities, such as Industry, Agriculture and Transport, while another group could be composed of the cultural investments, such as Research, Education and Culture. However, there are exponentially many possibilities by which the priorities can be a-priori combined and the cluster definitions can be biased by the researcher preferences. Therefore, some form of neutral, or non-arbitrary, data pre-processing is necessary: here, we apply the hypergraph modularity and model $F^2_{z,\gamma}$.

We consider two European nations, Spain and Italy, and we compare their expenditure priorities. When we analyse the frequencies by which priorities are mentioned, data are reported in Fig. 9. It can be seen that there are some regularities as well as differences between the two nations. For example, Education is ranked high in both nations, while Health is an important priority to Spain, but not as much in Italy. Conversely, Energy, Defence and Digitalization are mentioned more often in Italy than in Spain.

The Item Graph, see [39], is a useful tool to visualize how answers are jointly selected. In Fig. 10(a), the hypergraph is projected into a simple graph: there is an arc between two items every time they were jointly selected by a respondent. In the original hypergraph, model ($F^2_{z,\gamma}$) has been run with parameter $\gamma = \frac{2}{3}$: as respondents may elicit up to four items, a hyperedge is internal to a community under the following conditions. If the hyperedge contains two nodes, then both nodes must be in the same community. If it contains three nodes, then at least two of those nodes must be contained in the same community. If it contains four nodes, then at least three nodes must be contained in the same community.

In Fig. 10(b), the item graph is drawn after the hypergraph clustering, clusters are composed of nodes with the same colour. As can be seen, the results of the two nations are different, but some regularity can be found. In both nations we can find that Health, Job condition, Environment, Education and Research are clustered together, forming the core of priorities that can be ascribed to redistributive policies, that we could term as Welfare spending priorities. It can be found,

Table 9
Marginal values for each one of the 4-factors in the experiment.

Formulation		$h-Louvain$				$LSHM$			
n	m	NMI	OI	SIL	Time (s)	NMI	OI	SIL	Time (s)
50	50	0.59	0.37	0.21	**0.13**	**0.60**	**0.49**	**0.25**	0.41
	100	0.77	0.67	0.27	**0.16**	**0.88**	**0.85**	**0.37**	0.58
100	100	**0.51**	0.30	0.13	**0.15**	0.44	**0.37**	**0.15**	2.45
	200	0.67	0.57	0.19	**0.23**	**0.82**	**0.80**	**0.32**	4.85
Formulation		$h-Louvain$				$LSHM$			
n	L	NMI	OI	SIL	Time (s)	NMI	OI	SIL	Time (s)
50	3	0.64	0.46	0.26	**0.10**	**0.68**	**0.60**	**0.33**	0.45
	5	0.71	0.59	0.22	**0.18**	**0.79**	**0.73**	**0.30**	0.54
100	3	**0.55**	0.34	0.18	**0.13**	**0.55**	**0.49**	**0.23**	3.23
	5	0.63	0.53	0.14	**0.25**	**0.71**	**0.68**	**0.24**	4.08
Formulation		$h-Louvain$				$LSHM$			
n	μ	NMI	OI	SIL	Time (s)	NMI	OI	SIL	Time (s)
50	0	0.84	0.74	0.43	**0.17**	**0.90**	**0.87**	**0.58**	0.66
	0.2	0.69	0.53	0.19	**0.13**	**0.77**	**0.71**	**0.22**	0.42
	0.4	0.50	0.30	0.11	**0.13**	**0.53**	**0.43**	**0.14**	0.41
100	0	0.79	0.65	0.29	**0.18**	**0.83**	**0.79**	**0.54**	3.93
	0.2	0.58	0.42	0.10	**0.20**	**0.66**	**0.61**	**0.11**	3.71
	0.4	0.40	0.22	**0.07**	**0.20**	**0.41**	**0.35**	0.04	3.32
Formulation		$h-Louvain$				$LSHM$			
n	ξ	NMI	OI	SIL	Time (s)	NMI	OI	SIL	Time (s)
50	0	0.72	0.57	0.32	**0.13**	**0.76**	**0.70**	**0.41**	0.44
	0.2	0.72	0.58	0.26	**0.15**	**0.80**	**0.75**	**0.36**	0.52
	0.4	0.59	0.43	0.14	**0.15**	**0.64**	**0.56**	**0.17**	0.53
100	0	0.65	0.50	0.23	**0.18**	**0.67**	**0.62**	**0.37**	3.40
	0.2	0.64	0.49	0.15	**0.19**	**0.72**	**0.68**	**0.26**	3.62
	0.4	0.48	0.32	**0.09**	**0.21**	**0.51**	**0.45**	0.07	3.94

Table 10
Summary of the two metrics NMI and OI for the different instance sizes.

Formulation	$h-Louvain$				$LSHM$			
n	NMI	OI	SIL	Time (s)	NMI	OI	SIL	Time (s)
50	0.68	0.53	0.24	**0.14**	**0.74**	**0.68**	**0.31**	0.50
100	0.59	0.43	0.16	**0.19**	**0.63**	**0.59**	**0.23**	3.65

Table 11
Clusters different animals: (1) non-predator Mammals, (2) predator Mammals, (3) Fishes, Reptiles and Amphibians, (4) Birds and (5) Bugs and Invertebrates..

Class	Animals
1	antelope, buffalo, calf, cavy, deer, elephant, fruitbat, giraffe, girl, goat, gorilla, hamster, hare, oryx, pony, pussycat, reindeer, squirrel, vampire, vole, wallaby
2	aardvark, bear, boar, cheetah, dolphin, leopard, lion, lynx, mink, mole, mongoose, opossum, platypus, polecat, porpoise, puma, raccoon, seal, sealion, wolf
3	bass, carp, catfish, chub, dogfish, frog1, frog2, haddock, herring, newt, pike, piranha, pitviper, seahorse, seasnake, slowworm, sole, stingray, tuatara, tuna
4	chicken, crow, dove, duck, flamingo, gull, hawk, kiwi, lark, ostrich, parakeet, penguin, pheasant, rhea, skimmer, skua, sparrow, swan, tortoise, vulture, wren
5	clam, crab, crayfish, flea, gnat, honeybee, housefly, ladybird, lobster, moth, octopus, scorpion, seawasp, slug, starfish, termite, toad, wasp, worm

too, that Energy, Transport, SMEs, Defence, Public Administration and Industry are clustered together, forming the core of priorities that can be ascribed to Economic spending priorities. There are three items that are ambiguous: they are Agriculture, Culture, and Digitalization. The first two swapped their memberships between the two clusters, while Digitalization is a singleton cluster for Spain, but a Welfare issue for Italy.

As shown in [39], clusters can be used to reduce data dimension. Hyperedges correspond to respondents, so that we can analyse the feature of all the respondents within a cluster. In this example, respondents are classified as prioritizing the Welfare or the Economic spending, or termed as Unclassified if their hyperedges are not internal. After this reduction, in Fig. 11(a), it can be seen that in both countries the large majority of the citizens are in favour of the Welfare spending rather than the Economic spending, with the Spanish audience more oriented to it than the Italian. It is interesting to analyse further this finding. In political analysis, it is important to assess if political orientation plays a role on preferences. As can be seen in Figs. 11(b) and 11(c), it can be seen that in both countries the percentage of voters that are in favour of the Economic spending increases as the political preferences goes to the right of the political spectrum. In Spain, only 2% of left voters are in favour of the Economic spending, but they increase to 9% of the right voters. In Italy, they go from 8% of the left voters to 14% of the right voters. Conversely, the percentage of voters that in favour of the Welfare spending decreases. Therefore, hypermodularity clustering allowed to detect the regularity of the opinions along the political spectrum.

The second survey application that we consider is the Spanish Barometer, conducted by the *Centro de Investigacion Sociologica (CIS)*. On a monthly basis, Spanish voters are surveyed about various political and social issues. Among the questions, one is referring to the main Spanish problems, to which respondents can elicit up to three issues. Considering the Barometer of February 2024 [70], the list of the issues is:

- The economic crisis, the economic problems.
- The unemployment.
- Political problems in general.
- What political parties do.
- The misbehaving of the politicians.
- The corruption and fraud.
- The health system.

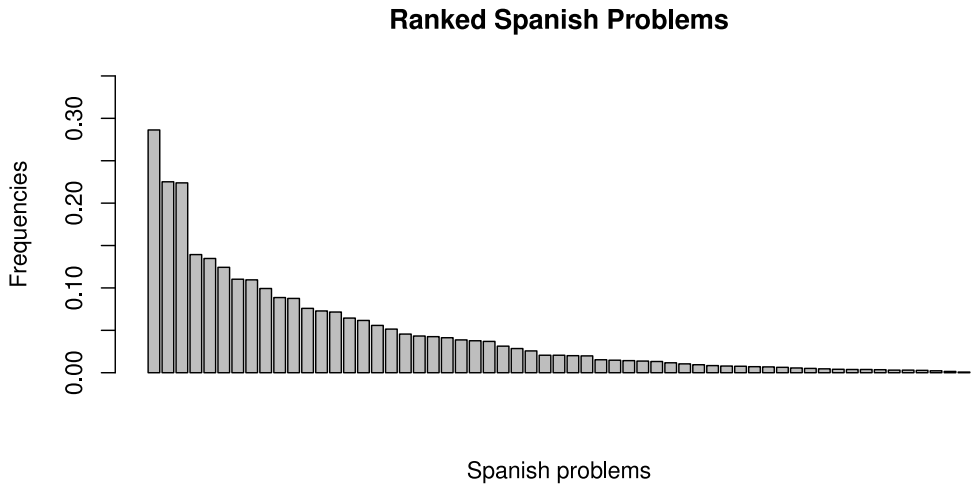


Fig. 12. Frequencies of the most important Spanish problem.

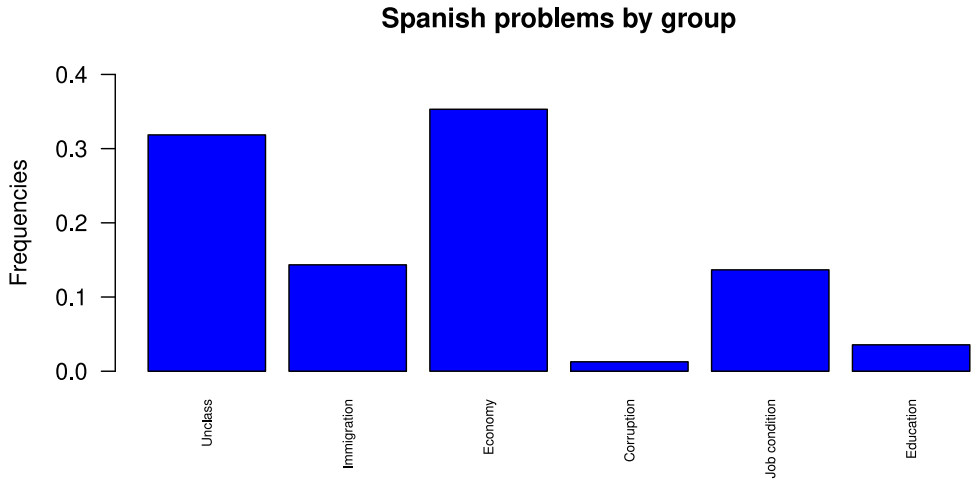


Fig. 13. Frequencies of the Spanish problem groups.

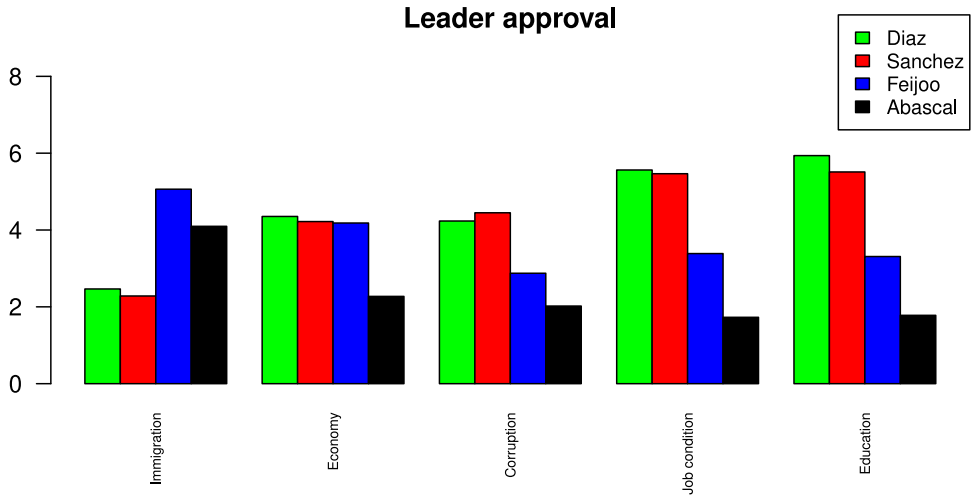


Fig. 14. Leader approval by opinion groups.

Table 12
Clusters most relevant problems.

Class	Problem	Frequencies
1	The Government, the parties, or other political leaders	0.13
1	The immigration	0.11
2	The economic crisis, the economic problems	0.29
2	The unemployment	0.23
3	What political parties do	0.06
3	The corruption and fraud	0.04
4	The problems related to the job conditions.	0.14
4	The housing	0.09
5	The education system	0.11
5	The health system	0.10

- The social problems.
- The immigration.
- The problems related to the job conditions.
- The pensions.
- The lack of agreement, unity, ability to collaborate
- The functioning of public services.
- The education system.
- The growth of tariffs, the energy crisis.
- The Spanish production model, the missing inversion on industry and I + D.
- The housing.
- The war between Ukraine and Russia.
- The climate change.
- Problems related to water scarcity, the drought.
- Other: What?

There are 19 items, but the last choice, *Other: What?* represents an open question. Here, respondents can specify what issue not present in the list they consider as a main Spanish problem. In that specific survey, respondents elicited up to 59 issues, some of them completely original, as the Cybersecurity, while some overlap with at least one of the closed list. For example, the freely chosen problem *The environment* can be considered quite similar to *The climate change*, available in the list.

It is problematic to analyse correctly these data, without some form of grouping or classification of all the mentioned issues in few general categories. In Fig. 12, the frequencies by which issues are mentioned are reported. As can be seen, data are quite dispersed. There are only three items that are elicited by more than 20% of the respondents. They are: *The economic problems*, *the unemployment*, *the political problems in general*. The rest of the issues are elicited with much lower frequencies. Given the dispersion of the answers, it is rather impervious to analyse data by single issues, but it could be convenient to aggregate problems into some classes. In this way, we may obtain classes with a sufficient size to calculate reliable statistics. We will apply hypergraph clustering to these data. The nodes of the graph are the problems selected by the respondents, and for each of the respondents there is a hyperedge connecting its selected problems. Model ($F^2_{z,\gamma}$) is run $\gamma = \frac{2}{3}$, since each respondent can elicit at most three issues, so that internal arcs must have at least two nodes within a group.

We found that the Spanish problems can be clustered into five groups, whose composition seems sufficiently homogeneous. In Table 12, we have reported the two most important problems for each group, and the frequencies by which they were elicited by respondents. The pairs are actually composed of similar issues. For example, *Unemployment* is paired with the *Economic problems*, the *Education* is paired with the *Health system*, and so on. The full list of the cluster's composition is reported in the Appendix Additional information from survey data. According to the main problems of the groups, we termed them as: *Immigration*, *Economy*, *Corruption*, *Job Condition*, *Education*. Next, we can redefine the respondents' elicited problems as one of the five classes above, or being unclassified if answers are spread among different groups. After this coding, we obtain that the clusters are selected with the frequencies reported in Fig. 13. It can be seen that the majority of the respondents selected the *Economy* group (35.3%) or are *Unclassified* (31.8%), but there are two not negligible groups of respondents that centred the Spanish problems on *Immigration* (14.3%) and *Job conditions* (13.6%). Then, there are two minoritarian groups, composed by people that focused the Spanish problems on *Corruption* (1.3%) and on *Education* (3.6%).

Hypergraph modularity succeeded in reducing the dimension of the Spanish problems from a list of 59 issues to 5 groups. This is a synthetic description of the Spanish public opinion that can be used in many ways. To make an example, we can analyse the association between main problems and the approval of the main Spanish political leaders. The leaders that are surveyed by the questionnaire are: Pedro Sanchez, leader of the Socialist Party and prime minister, Alberto Núñez Feijóo, leader of Popular Party, the main opponent to Sanchez, Santiago Abascal, leader of extreme right party Vox, Yolanda Díaz, leader of the left-wing movement Sumar and vice-president of the Sanchez's government. We could guess that the leader's approval is associated to what is considered the main Spanish problem: some themes, as *Education*, are typical of left policies, while others, as *Immigration*, are themes that characterized the right. In Fig. 14, it can be found that our guess is right. As can be seen, voters that worry about *Immigration* give higher assessment to right political leaders, Feijoo and Abascal, than to left leaders, Sanchez and Diaz. Conversely, if voters worry about *Corruption*, *Job Condition*, or *Education*, then they appreciate more the left leaders. Finally, people that worry about the *Economy* give similar assessments to Diaz, Sanchez, and Feijoo, showing that on that particular issue no leader has a specific advantage over the other.

7. Conclusions

In this paper, we proposed a definition of hypergraph modularity, a statistics that is well-suited to measure the consistency of communities defined on hypergraphs. Previous attempts did not try to propose a specific metric, rather they projected hypergraphs to simple graphs, with projected arcs eventually weighted. Our intuition was that this method had some inconsistency, as a hyperedge could contribute to the definition of more than one community. Therefore, we developed exact MILP models to solve the optimal modularity partition problem for both cases, the first in which the hypergraph is projected, the second in which we optimize the specific hypergraph modularity function. Relying on the exact solution of the two MILP problems, we observed that the hypergraph modularity function recovered the true hidden patterns of the community more precisely than the projected modularity function. These results are applied to a peculiar application, that is, survey data, in which the data size is manageable by MILP models. We have shown that some questions whose answers are chosen from a list can be modelled as hypergraphs. In this way, the answers can be clustered by modularity more accurately than by using standard statistical techniques, providing new insight into these data.

Our results left open the research to some aspects of hypergraph modularity that we could not address in this contribution. Firstly, there is surely the need to implement a specific heuristic algorithm when the size of the hypergraph is so large that the MILP model cannot provide the solution in reasonable computational time. Most likely, the adaptation of the principle applied in the Louvain method could do the task pretty well. Secondly, there are many applications of graph modularity to data that can also be interpreted as hypergraph models. Therefore, once a heuristic algorithm is available, the comparison between the two models would be very interesting.

As a conclusion, all the novel methods presented in the document can be found in https://github.com/ftgarcia97/Modularity-for-Hypergraph-Clustering-Methodologies-and-Applications/tree/main. We include the Python code of exact formulations F_x^1 and $F_{z,\gamma}^2$, the LSHM algorithm and the benchmark hypergraph generator used in the experiments.

CRediT authorship contribution statement

Stefano Benati: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Justo Puerto:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Francisco Temprano:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

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.

Acknowledgements

Stefano Benati acknowledges financial support under the National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.1, Call for tender No. 104 published on 2.2.2022 by the Italian Ministry of University and Research (MUR), funded by the European Union – NextGenerationEU– Project Title Networks: decomposition, clustering and community detection – CUP E53D23006360006 - Grant Assignment Decree No. 967 adopted on 30/6/2024 by the Italian Ministry of University and Research (MUR). The authors of this research acknowledge financial support by: the research project PID2020-114594GB-C21 funded by the Spanish Ministerio de Ciencia e Innovación and Agencia Estatal de Investigación (MCIN/AEI/10.13039/501100011033); and Junta de Andalucía, Spain P18-FR-1422. The third
author also acknowledges PREDOC PAIDI2020, predoctoral fellowship financed by Consejería de Transformación Económica, Industria, Spain, Conocimiento y Universidades de la Junta de Andalucía.

Appendix Additional information from survey data

Problem	Frequency	Group
El cambio climático	0.087620988	2
El Gobierno y partidos o políticos/as concretos/as	0.134742741	1
La vivienda	0.088639837	4
Lo que hacen los partidos políticos	0.061640346	3
La corrupción y el fraude	0.04330107	3
Los problemas políticos en general	0.223892002	2
El paro	0.225165563	2
La inseguridad ciudadana	0.045593479	2
Los problemas relacionados con la juventud. Falta de apoyo y oportunidades a los/as jóvenes	0.071574121	4
La crisis económica. los problemas de índole económica	0.286296485	2
El mal comportamiento de los/as políticos/as	0.124299542	2
Los problemas relacionados con la calidad del empleo	0.13932756	4
Las desigualdades. incluida la de género. las diferencias de clases. la pobreza	0.055781966	4
Los problemas de índole social	0.072847682	4
Los extremismos	0.042536933	4
La inmigración	0.110290372	1
La falta de acuerdos, unidad y capacidad de colaboración. Situación e inestabilidad política	0.051451859	4
Los nacionalismos	0,020122262	1
La falta de confianza en los/las políticos/as y las instituciones	0.007131941	4
Los problemas relacionados con el agua, la sequía	0.075904228	2
La crisis de valores	0.06444218	1
Ley de Amnistía	0.020631686	2
La educación	0.109526235	5
La sanidad	0.099337748	5
Aumento de la crispación social, revueltas sociales	0.020631686	4
Papel de los medios de comunicación y redes: desinformación, manipulación informativa, difusión de bulos	0.028527764	4
Los problemas relacionados con la mujer	0.004075395	4
Otras respuestas	0.038716251	1
La Administración de Justicia	0.01986755	3
El modelo productivo español. La falta de inversión en industrias e I+D	0.02572593	1
La independencia de Cataluña	0.031329598	1
Los problemas de la agricultura, ganadería y pesca	0.015282731	1
La subida de impuestos	0.013754457	1
El funcionamiento de los servicios públicos	0.036933265	1
La subida de tarifas energéticas	0.010443199	2
Ninguno	0.041263372	6
Las hipotecas	0.001528273	4
Guerra de Ucrania y Rusia	0.007896077	2
El funcionamiento de la democracia	0.037697402	1
Los estatutos de autonomía	0.013245033	1
Las incertidumbres ante el futuro, la inseguridad y el miedo al futuro	0.014773306	2
Política exterior y relaciones internacionales. Papel de España en el marco internacional	0.008405502	1
Las drogas	0.005094244	1
La España vaciada, la despoblación	0.003056546	5
El medio ambiente	0.014263882	3
Las pensiones	0.007641365	1
Poca conciencia ciudadana (falta de civismo, de sentido espíritu cívico)	0.00942435	4
Uso, regulación y seguridad en tecnologías para los/as menores, adolescentes y jóvenes	0.005603668	4

Los problemas relacionados con los/as autónomos/as	0.003820683	1
La falta de servicios públicos. Los recortes	0.00356597	4
Ciberseguridad: uso, delitos y regulación de nuevas tecnologías	0.000764137	3
Los bancos	0.006877229	1
La violencia de género	0.01171676	2
El racismo	0.004584819	4
La Monarquía	0.00229241	2
La emigración	0.003056546	5
La ocupación de viviendas	0.003820683	1
Las guerras en general	0.006367804	2
Las infraestructuras	0.002801834	1

Dataset	n	m	Hyperedge size	Number of hyperedges
Eurobarometer	15	1989	1	19
			2	313
			3	406
			4	1251
Spanish barometer	58	3849	1	225
			2	553
			3	3071

Data availability

Data will be made available on request.

References

[1] Fortunato S. Community detection in graphs. Phys Rep 2010;(486):75–174.
[2] Fortunato S, Hric D. Community detection in networks: A user's guide. Phys Rep 2016;(659):75–174.
[3] Fortunato S, Newman M. 20 years of network community detection. Nat Phys 2022;18(8):848–50.
[4] Newman MEJ. Analysis of weighted networks. Phys Rev E 2004;70(5):056131.
[5] Agarwal G, Kempe D. Modularity-maximizing graph communities via mathematical programming. Eur Phys J B 2008;66(3):409–18. http://dx.doi.org/10.1140/epjb/e2008-00425-1.
[6] Grötschel M, Wakabayashi Y. A cutting plane algorithm for a clustering problem. Math Program 1989;(45):59–96.
[7] Dinh TN, Thai MT. Towards optimal community detection: From trees to general weighted networks. Internet Math 2015;11(3):181–200.
[8] Aloise D, Cafieri S, Caporossi G, Hansen P, Perron S, Liberti L. Column generation algorithms for exact modularity maximization in networks. Phys Rev E - Stat Nonlinear, Soft Matter Phys 2010;82(4).
[9] Aref S, Chheda H, Mostajabdaveh M. The bayan algorithm: Detecting communities in networks through exact and approximate optimization of modularity. 2022, http://dx.doi.org/10.48550/arXiv.2209.04562, CoRR.
[10] Clauset A, Newman M, Moore C. Finding community structure in very large networks. Phys Rev E - Stat Nonlinear, Soft Matter Phys 2004;70(6 2):066111/1–6.
[11] Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008;2008(10).
[12] Liu X, Murata T. Advanced modularity-specialized label propagation algorithm for detecting communities in networks. Phys A 2010;389(7):1493–500.
[13] Lou H, Li S, Zhao Y. Detecting community structure using label propagation with weighted coherent neighborhood propinquity. Phys A 2013;392(14).
[14] Sobolevsky S, Campari R, Belyi A, Ratti C. General optimization technique for high-quality community detection in complex networks. Phys Rev E - Stat Nonlinear, Soft Matter Phys 2014;90(1).
[15] Traag V, Waltman L, van Eck N. From louvain to leiden: Guaranteeing well-connected communities. Sci Rep 2019;9(1).
[16] Nascimento M, Pitsoulis L. Community detection by modularity maximization using GRASP with path relinking. Comput Oper Res 2013;40(12):3121–31.
[17] Aloise D, Caporossi G, Hansen P, Liberti L, Perron S, Ruiz M. Modularity maximization in networks by variable neighborhood search. Graph Partitioning Graph Clust 2013;588:113–27.
[18] Džamić D, Aloise D, Mladenović N. Ascent–descent variable neighborhood decomposition search for community detection by modularity maximization. Ann Oper Res 2019;272(1–2):273–87.
[19] Sobolevsky S, Belyi A. Graph neural network inspired algorithm for unsupervised network community detection. Appl Netw Sci 2022;7. http://dx.doi.org/10.1007/s41109-022-00500-z.
[20] Aref S, Mostajabdaveh M, Chheda H. Heuristic modularity maximization algorithms for community detection rarely return an optimal partition or anything similar. Lect Notes Comput Sci (Incl Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 2023;14076 LNCS:612–26. http://dx.doi.org/10.1007/978-3-031-36027-5-48.
[21] Aref S, Mostajabdaveh M. Analyzing modularity maximization in approximation, heuristic, and graph neural network algorithms for community detection. J Comput Sci 2024;78. http://dx.doi.org/10.1016/j.jocs.2024.102283, cited By 1.
[22] Fortunato S, Barthélemy M. Resolution limit in community detection. Proc Natl Acad Sci USA 2007;104(1):36–41.
[23] Botta F, Del Genio C. Finding network communities using modularity density. J Stat Mech Theory Exp 2016;2016(12).
[24] Costa A, Ng TS, Foo LX. Complete mixed integer linear programming formulations for modularity density based clustering. Discrete Optim 2017;(25):141–58.
[25] Calderoni F, Brunetto D, Piccardi C. Communities in criminal networks: A case study. Soc Networks 2017;48:116–25.
[26] Avellone A, Benati S, Grassi R, Rizzini G. On finding the community with maximum persistence probability. 4OR 2023.
[27] Zhang S, Wang R-S, Zhang X. Identification of overlapping community structure in complex networks using fuzzy c-means clustering. Phys A 2007;(374):483–90.
[28] Benati S, Ponce D, Puerto J, Rodríguez-Chía AM. A branch-and-price procedure for clustering data that are graph connected. European J Oper Res 2022;297(3):817–30. http://dx.doi.org/10.1016/j.ejor.2021.05.043.
[29] Benati S, Puerto J, Rodríguez-Chía AM, Temprano F. Overlapping communities detection through weighted graph community games. PLoS One 2023;18(4):1–35.
[30] Ruan Y, Liu S, Tang J, Guo Y, Yu T. GLC: A dual-perspective approach for identifying influential nodes in complex networks. Expert Syst Appl 2025;268:126292. http://dx.doi.org/10.1016/j.eswa.2024.126292.
[31] Cascón J, González-Arteaga T, de Andrés Calle R. A new preference classification approach: The λ-dissensus cluster algorithm. Omega (United Kingdom) 2022;111.
[32] Díaz R, Fernández E, Figueira J-R, Navarro J, Solares E. A new hierarchical multiple criteria ordered clustering approach as a complementary tool for sorting and ranking problems. Omega (United Kingdom) 2023;117.
[33] Bulò S, Pelillo M. A game-theoretic approach to hypergraph clustering. Adv Neural Inf Process Syst 2009;22.
[34] Zhou D, Huang J, Schölkopf B. Learning with hypergraphs: Clustering, classification, and embedding. Adv Neural Inf Process Syst 2006;19.
[35] Zien J, Schlag M, Chan P. Multilevel spectral hypergraph partitioning with arbitrary vertex sizes. Computer- Aided Des Integr Circuits Syst IEEE Trans on 1999;18:1389–99. http://dx.doi.org/10.1109/43.784130.
[36] Rodriguez-Velazquez JA. On the Laplacian spectrum and walk-regular hypergraphs. Linear Multilinear Algebra 2003;51:285–97. http://dx.doi.org/10.1080/0308108031000084374.
[37] Agarwal S, Lim J, Zelnik-Manor L, Perona P, Kriegman D, Belongie S. Beyond pairwise clustering. 2, 2005, p. 838–45. http://dx.doi.org/10.1109/CVPR.2005.89, vol. 2,
[38] Li P, Milenkovic O. Inhomogeneous hypergraph clustering with applications. Adv Neural Inf Process Syst 2017;30.
[39] Benati J. A network model for multiple selection questions in opinion surveys. Qual Quant 2024;58:1163–79.
[40] Lancichinetti A, Fortunato S, Radicchi F. Benchmark graphs for testing community detection algorithms. Phys Rev E 2008;78:046110.
[41] Purkait P, Chin T-J, Sadri A, Suter D. Clustering with hypergraphs: The case for large hyperedges. IEEE Trans Pattern Anal Mach Intell 2017;39(9):1697–711. http://dx.doi.org/10.1109/TPAMI.2016.2614980.
[42] Kumar T, Vaidyanathan S, Ananthapadmanabhan H, Parthasarathy S, Ravindran B. Hypergraph clustering by iteratively reweighted modularity maximization. Appl Netw Sci 2020;5(1):1–22.
[43] Newman MEJ. Networks: An Introduction. Oxford University Press; 2010.
[44] Nicosia V, Mangioni G, Carchiolo V, Malgeri M. Extending the definition of modularity to directed graphs with overlapping communities. J Stat Mech Theory Exp 2009;((03) P03024).
[45] Shi J, Malik J. Normalized cuts and image segmentation. IEEE Trans Patterns Anal Mach Intell 2000;22(8).
[46] Xu L, Li W, Schuurmans D. Fast normalized cut with linear constraints. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE; 2009, p. 2866–73.
[47] Dhillon IS, Guan Y, Kulis B. Kernel k-means: spectral clustering and normalized cuts. In: Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining. 2004, p. 551–6.

[48] Ponce D, Puerto J, Temprano F. Mixed-integer linear programming formulations and column generation algorithms for the minimum normalized cuts problem on networks. European J Oper Res 2024;316(2):519–38. http://dx.doi.org/10.1016/j.ejor.2024.02.033.
[49] Chekuri C, Xu C. Minimum cuts and sparsification in hypergraphs. SIAM J Comput 2018;47(6):2118–56.
[50] Kamiński B, Poulin V, Prałat P, Szufel P, Théberge F. Clustering via hypergraph modularity. PLoS One 2019;14(11):e0224307.
[51] Fox K, Panigrahi D, Zhang F. Minimum cut and minimum k-cut in hypergraphs via branching contractions. ACM Trans Algorithms 2023;19(2):1–22.
[52] Kamiński B, Misiorek P, Prałat P, Théberge F. Modularity based community detection in hypergraphs. J Complex Networks 2024;12. URL https://api.semanticscholar.org/CorpusID:270711398.
[53] Chodrow PS, Veldt N, Benson AR. Generative hypergraph clustering: From blockmodels to modularity. Sci Adv 2021;7(28):eabh1303. http://dx.doi.org/10.1126/sciadv.abh1303.
[54] Chodrow P. Configuration models of random hypergraphs. J Complex Networks 2020;8. http://dx.doi.org/10.1093/comnet/cnaa018.
[55] Ales Z, Knippel A. The K-partitioning problem: Formulations and branch-and-cut. Networks 2020;76(3):323–49.
[56] Benati S, Puerto J, Rodríguez-Chía AM. Clustering data that are graph connected. European J Oper Res 2017;261(1):43–53.
[57] Benati S, Puerto J, Rodríguez-Chía AM. Clustering data that are graph connected. European J Oper Res 2017;261(1):43–53. http://dx.doi.org/10.1016/j.ejor.2017.02.009, URL https://www.sciencedirect.com/science/article/pii/S0377221717301145.
[58] Tandon A, Albeshri A, Thayananthan V, Alhalabi W, Radicchi F, Fortunato S. Community detection in networks using graph embeddings. Phys Rev E 2021;103:022316. http://dx.doi.org/10.1103/PhysRevE.103.022316, URL https://link.aps.org/doi/10.1103/PhysRevE.103.022316.
[59] Kamiński B, Pralat P, Theberge F. Hypergraph artificial benchmark for community detection (h–ABCD). J Complex Networks 2023;11. http://dx.doi.org/10.1093/comnet/cnad028.
[60] Fred A, Jain A. Robust data clustering. 2003 IEEE Comput Soc Conf Comput Vis Pattern Recognit 2003. Proc 2003;2. II–II.
[61] Collins LM, Dent CW. Omega: A general formulation of the rand index of cluster recovery suitable for non-disjoint solutions. Multivar Behav Res 1988;23(2):231–42. http://dx.doi.org/10.1207/s15327906mbr2302_6.
[62] Mahmoudi A, Jemielniak D. Proof of biased behavior of normalized mutual information. Sci Rep 2024;14. http://dx.doi.org/10.1038/s41598-024-59073-9.
[63] Feo TA, Resende MG. A probabilistic heuristic for a computationally difficult set covering problem. Oper Res Lett 1989;8(2):67–71. http://dx.doi.org/10.1016/0167-6377(89)90002-3.
[64] Marti R, Ribeiro C. Multi-start methods for combinatorial optimization. European J Oper Res 2013;226:1–8. http://dx.doi.org/10.1016/j.ejor.2012.10.012.
[65] Benati S. Categorical data fuzzy clustering: An analysis of local search heuristics. Comput Oper Res 2008;35(3):766–75. http://dx.doi.org/10.1016/j.cor.2006.05.001.
[66] Rousseeuw PJ. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 1987;20:53–65. http://dx.doi.org/10.1016/0377-0427(87)90125-7, URL https://www.sciencedirect.com/science/article/pii/0377042787901257.
[67] Macon KT, Mucha PJ, Porter MA. Community structure in the united nations general assembly. Phys A 2012;391(1):343–61. http://dx.doi.org/10.1016/j.physa.2011.06.030.
[68] Voeten E, Strezhnev A, Bailey M. United nations general assembly voting data. 2009, http://dx.doi.org/10.7910/DVN/LEJUQZ.
[69] European Commission B. Eurobarometer 99.4 (2023). 2024, http://dx.doi.org/10.4232/1.14167, GESIS, Cologne. ZA7997 Data file Version 1.0.0.
[70] Centro de Investigacion Sociologicas. Spanish barometro. 2024, DA3441, https://www.cis.es/en/detalle-ficha-estudio?idEstudio=14774.