Hypermodularity and community detection in hypergraphs

Charo I. del Genio
Institute of Smart Agriculture for Safe and Functional Foods and Supplements, Trakia University, Stara Zagora 6000, Bulgaria; Research Institute of Interdisciplinary Intelligent Science, Ningbo University of Technology, 315104 Ningbo, China; and School of Mathematics, North University of China, 030051 Taiyuan, China

(Received 2 December 2024; accepted 18 June 2025; published 10 July 2025)

Numerous networked systems feature a structure of nontrivial communities, which often correspond to their functional modules. Such communities have been detected in real-world biological, social, and technological systems, as well as in synthetic models thereof. While much effort has been devoted to developing methods for community detection in traditional networks, the study of community structure in networks with higher-order interactions is still not as extensively explored. In this article, we introduce a formalism for the hypermodularity of higher-order networks that allows us to use spectral methods to detect community structures in hypergraphs. We apply this approach to synthetic random networks as well as to real-world data, showing that it produces results that reflect the nature and the dynamics of the interactions modeled, thereby constituting a valuable tool for the extraction of hidden information from complex higher-order data sets.

DOI: 10.1103/58dr-wktc

I. INTRODUCTION

Many complex systems have the structure of a network, in which pairs of elements of a discrete set, called the nodes, are connected when they share a common property [1–4]. Typically, such networks have structural features with different characteristic length scales. The smallest length scale is that of the number of links (also called edges) of a given node, which takes the name of degree. The links between the neighbors of a specific node define the local neighborhood structure, which is the next larger length scale. As one considers the whole network, one may find sets of nodes whose elements have more connections amongst themselves than they have with nodes in other such sets. When this happens, these mesoscale sets take the name of communities, and the network is said to exhibit a nontrivial community structure. The presence of network communities has been detected in a large number of real-world systems, such as protein networks, social networks, food webs, and synthetic models of complex systems of diverse nature [5–14]. In general, the community structure of a network affects the behavior of the system, and often one can trace a direct correspondence between communities and structural units responsible for specific functions.

As a result, a large number of methods have been developed to detect communities in networks [14,15]. A particularly successful approach is the maximization of a quality function that measures how pronounced a partition of a network into communities is by computing the difference between the density of the observed intracommunity links and that of a suitably chosen null model. The most commonly used such function is called *modularity* [16]. However, finding the network partition that maximizes it is an NP-hard computational problem [17]. Thus, large efforts have been devoted to creating algorithms that provide the best possible approximation to the global maximum in a workable running time [18–27].

Recently, however, it has become increasingly clear that dyadic interactions are only a part of the important relationships that occur in complex systems. In fact, researchers have recognized the relevance of collective interactions for the functioning of systems as diverse as social groups, neural networks, and multispecies assemblages of microorganisms, sparking a new interest in structures with higher-order interactions [28,29]. As a result, the main type of structural backbone that is currently generating considerable attention is no longer that of a graph, but rather that of a hypergraph. In hypergraphs, edges do not necessarily connect pairs of elements, but rather they can connect triplets, quadruplets, or any other number of nodes [30].

Unfortunately, the landscape of research on the subject of community detection in hypergraphs is not as well developed as in the case of traditional graphs. However, several methods have been proposed in recent years that address the problem from different directions. In some instances, these approaches are dynamical in nature, and analyze the outcome of a stochastic process on the hypergraph to infer its structure of modules and communities [31]. Other methods, instead, rely on projecting the hypergraph onto a graph [32] or on creating an auxiliary graph on which to carry out a community-detection procedure [33]. At the cost of a significant potential to lose information, especially when performing projections [29], these techniques have the advantage of reducing the dimensionality of the problem, and resulting, therefore, in faster algorithms. A similar idea of dimensional reduction is behind methods based on embedding, in which the starting hypergraph is related to a lower-dimensional Euclidean space, and

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

communities are detected by solving an optimization problem or via the application of existing clustering algorithms [34,35]. Such embeddings can themselves be obtained via the application of spectral methods, which have already proved extremely powerful for the task of finding communities in classic networks [16]. Indeed, spectral approaches often form the basis of the more advanced higher-order methods, where they are occasionally integrated by Bayesian inference algorithms [36]. The already mentioned possibility of information loss has also led to the development of algorithms that operate directly on the whole hypergraph. In this case, tensor methods are necessary, as one needs to relate the existence of communities to some property of the adjacency tensor of the higher-order network. One possible approach is to use the idea of tensor decomposition, which has been, in fact, long suggested as being applicable to community detection [37], and has been eventually used for this goal, once more in conjunction with a Bayesian framework [38]. Alternatively, others have exploited the spectral features of higher-order equivalents of the graph Laplacian, effectively extending to higher-order structures the existing knowledge of the relation between the Laplacians of traditional graphs and the presence of communities [39]. An exception to these approaches is an extension to hypergraphs of the Chung-Lu model, which results in a modularity measure that is closely related to the one originally used in traditional networks [40,41].

Here, we introduce and characterize modularity for hypergraphs using a formalism that allows us to create a spectral algorithm for its maximization. Our method differs from the existing ones mentioned above as it does not rely on embeddings of any kind and does not require the structure of the higher-order network to be a simplicial complex. Also, while the core of our approach is a spectral method, this is applied onto a modularity tensor, rather than on the adjacency tensor or the Hodge Laplacian of the structure. Note that the specific form of the modularity measure we use is mathematically equivalent to that of Refs. [40,41]. However, we are able to operate a tensor decomposition on it because of the different formalism adopted, which rests on a derivation of a closed-form combinatorial expression that enables us to cast the hypermodularity equation in vector form. We then validate our method by analyzing random hypergraphs and real-world higher-order networks, showing that it detects communities that correspond to the similarity of cognitive processes in people and reflect the underlying social dynamics.

II. RESULTS

A. Hypermodularity

In a fully general case, a hypergraph may contain edges involving different numbers of nodes. If all the edges have the same size k, the hypergraph is called k-uniform. Note that this means that classic networks with only dyadic interactions are 2-uniform hypergraphs. However, each edge size is independent from all the others, and they all correspond to different specific orders of many-body interactions. Thus, a hypergraph H can be considered as the union of multiple independent uniform sub-hypergraphs, each consisting of edges of a single size. In turn, this consideration also applies to the task of finding the community structure of H, since, for example, a group of nodes may form strong 3-body interactions but no significant pairwise ones. Therefore, in the following, we study the problem on uniform hypergraphs, since a solution on them also provides one to the general case.

The idea behind classic network modularity is to exploit a measure based on the difference between the number of observed edges between two nodes and the one that would be expected if the placement of all the edges in the graph were randomized while keeping the degrees of the nodes unchanged. Given a partition of a network with N nodes and m edges into a set of communities, this approach results in the following expression for the modularity Q [16]:

$$Q = \frac{1}{2m}\sum_{i=1}^{N}\sum_{j=1}^{N}\left[\left(A_{i,j} - \frac{d_i d_j}{2m}\right)\delta_{C_i,C_j}\right], \tag{1}$$

where $\mathbf{A}$ is the adjacency matrix, d_i is the degree of node i, δ is Kronecker's symbol, and C_i is the community to which node i has been assigned. Applying the same idea to a k-uniform hypergraph, we introduce an expression for the *hypermodularity* of a partition:

$$Q = \frac{1}{k!m}\sum_{\{v_1,\dots,v_k\}}\left[\left(A_{v_1,\dots,v_k} - \frac{(k-1)!}{(km)^{k-1}}\prod_{i=1}^{k} d_{v_i}\right) \times \prod_{\substack{i,j=1\\ i\neq j}}^{k}\delta_{C_{v_i},C_{v_j}}\right]. \tag{2}$$

In the formula above, $\mathbf{A}$ is the adjacency tensor of the hypergraph, the combinatorial factor before the product of degrees accounts for the number of available ways to join nodes of given degrees in an edge, and the sum is on all the possible choices of k nodes, allowing for repetitions. Thus, the expression directly measures the difference between the number of hyperedges of size k that are found amongst the nodes in each community and the number that would be expected if the same hyperedges had been placed at random. The task is thus to find the partition of the network that maximizes Q. Note that $\mathbf{A}$ is more precisely described as a hypermatrix, rather than a tensor, since it does not represent a multilinear application. However, with a slight abuse of notation, we will still refer to it as a tensor, as this is the most commonly adopted terminology, especially in the applied literature. Also note that, for $k = 2$, the expression reduces to that of graph modularity, which becomes therefore a special case of this more general formulation, as desired.

The principal difference between the hypermodularity of Eq. (2) and the classic modularity for graphs is that the latter can be cast into a pure matrix equation, whereas such an expression cannot be obtained straightforwardly in the hypergraph case. Thus, one cannot directly use the same methods employed on traditional networks to produce a bipartition, which is a starting step in many community detection algorithms.

B. Spectral partitioning

To solve this problem, first note that if the hypergraph is simple, meaning that it does not have multiple edges, as it is often the case in real-world structures, then one can write $A_{v_1,\dots,v_k} = \mathbf{1}_{(v_1,\dots,v_k)\in H}$, where $\mathbf{1}$ is an indicator variable

whose value is 1 if the edge $(v_1, \ldots, v_k)$ belongs to H and 0 otherwise. Thus, Eq. (2) can be rewritten as

$$Q = \frac{1}{k!m}\sum_{\{v_1,\ldots,v_k\}}\left[\left(\mathbf{1}_{(v_1,\ldots,v_k)\in H} - \frac{(k-1)!}{(km)^{k-1}}\prod_{i=1}^{k} d_{v_i}\right) \times \prod_{\substack{i,j=1\\ i\neq j}}^{k} \delta_{C_{v_i},C_{v_j}}\right]. \tag{3}$$

But then, one can consider the whole term within the round brackets in the equation above as a hypersymmetric data tensor $\mathbf{B}$, which, in principle, can be analyzed using higher-order singular-value decomposition (SVD), whose applications to data science and machine learning have proved to be very fruitful [42–46]. Given a data tensor, its higher-order SVD is a compact SVD carried out on one of its standard flattenings. These are dimensional reductions of the tensor that compact all its dimensions, except for an arbitrarily chosen one, into a single dimension. In our case, this means that the standard flattenings of $\mathbf{B}$ are the matrices $\mathbf{E}^{(i)}$ defined by

$$E^{(i)}_{v_i,(v_1,v_2,\ldots,v_{i-1},v_{i+1},\ldots,v_k)} = \mathbf{1}_{(v_1,\ldots,v_k)\in H} - \frac{(k-1)!}{(km)^{k-1}}\prod_{j=1}^{k} d_{v_j}\,, \tag{4}$$

where i is any integer from 1 to k. Note that, effectively, the column index of $\mathbf{E}^{(i)}$ is a sort of collective index, which can take a number of values that is equal to the number of all possible combinations of allowed values for all the indices of $\mathbf{B}$ except the i-th one. Also note that, with these manipulations, we do not neglect or discard any information that is present in the initial equation.

In general, the compact SVD of an $N \times M$ matrix $\mathbf{X}$ provides three matrices $\mathbf{U}$, $\mathbf{\Sigma}$, and $\mathbf{V}$ such that $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\dagger}$, where $\dagger$ indicates Hermitian conjugation. More specifically, $\mathbf{U}$ and $\mathbf{\Sigma}$ are $N \times N$ and $\mathbf{V}$ is $M \times N$. Also, the columns of $\mathbf{U}$ and those of $\mathbf{V}$ are the left singular vectors and the right singular vectors of $\mathbf{X}$, respectively, whereas $\mathbf{\Sigma}$ is diagonal and it contains the (positive) singular values of $\mathbf{X}$. For our purposes, the procedure is useful because, since $\mathbf{B}$ is real, so are its flattenings $\mathbf{E}^{(i)}$. This means that the Hermitian conjugation reduces to transposition, and that the matrix $\mathbf{U}$ is guaranteed to be real and orthogonal. Thus, if we could establish a relation between the product of Kronecker deltas in Eq. (3) and some vector quantities encoding the assignments of the nodes into communities, we would be able to maximize the hypermodularity by imposing the partition described by the left singular vector corresponding to the largest singular value.

To find this vector, one can use a variety of methods. However, since we are only interested in the first left singular vector, we can use an equivalent of the power iteration, thus avoiding the need to perform a full decomposition. The procedure is as follows. First, create an initial N-dimensional vector $\mathbf{z_0}$ whose components are standard normal random variables. Normalize $\mathbf{z_0}$, and compute the vector $\mathbf{z_1} = \mathbf{E}^{(i)}(\mathbf{E}^{(i)\mathrm{T}}\mathbf{z_0})$, where T indicates transposition. Normalize $\mathbf{z_1}$ and repeat the multiplication to obtain $\mathbf{z_2}$. After a large number of iterations, the resulting vector $\mathbf{z}_\infty$ is the first left singular vector of $\mathbf{E}^{(i)}$. Then, one can determine the community assignment based on the signs of the elements of the vector: if the ith element is positive, assign node i to the first community; if it is negative, assign it to the second one.

Note that, in principle, Eq. (4) defines k different possible flattenings of the data tensor. Thus, one may wonder which one of them to use for the SVD. However, since the tensor is hypersymmetric, one can intuitively see that any dimension can be chosen for the flattening, because reordering them does not change the tensor. Thus, in the following, we use the k-th one without loss of generality, omitting from the formulas the explicit mention of the dimension used. For a more formal proof of this, see Appendix A.

The next task on the road to building a method for spectral bisection of a higher-order network is finding a way to express Eq. (3) in terms of vector quantities. To start, recall that one can write the simple Kronecker delta of Eq. (1) as

$$\delta_{C_i,C_j} = \tfrac{1}{2}(s_i s_j + 1)\,, \tag{5}$$

where s_i are spin-like community-identifying variables that can take the values $+1$ and -1, depending on the community to which node i is assigned. Next, note that, given k nodes, the product of deltas in Eq. (3) does not need to include all $\binom{k}{2}$ possible combinations thereof. In fact, because of the transitivity of equality, the product is best expressed as a chain, so that

$$\prod_{\substack{i,j=1\\ i\neq j}}^{k} \delta_{C_{v_i},C_{v_j}} = \delta_{C_{v_1},C_{v_2}}\delta_{C_{v_2},C_{v_3}}\cdots\delta_{C_{v_{k-1}},C_{v_k}}\,. \tag{6}$$

Thus, one can use Eq. (5) to explicitly rewrite the product chain for different values of k. Then, exploiting the facts that the square of any spin-like variable is equal to 1 and that the sum of all the elements of the data tensor vanishes by construction, one can find a general expression for the hypermodularity of a bipartition, namely

$$Q = \frac{1}{2^{k-1}k!m}\mathbf{s}^{\mathrm{T}}\mathbf{E}\boldsymbol{\sigma}^{(k)}\,, \tag{7}$$

where $\mathbf{s}$ is a vector containing the community assignments of all nodes, and $\boldsymbol{\sigma}^{(k)}$ is a vector with N^{k-1} components that can be written as

$$\boldsymbol{\sigma}^{(k)} = (\sigma_{1,1,\ldots,1}, \sigma_{2,1,\ldots,1}, \sigma_{3,1,\ldots,1}, \ldots, \sigma_{N,1,\ldots,1}, \sigma_{1,2,\ldots,1}, \sigma_{2,2,\ldots,1}, \sigma_{3,2,\ldots,1}, \ldots, \sigma_{N,N,\ldots,N})^{\mathrm{T}}\,. \tag{8}$$

Here, the elements of $\boldsymbol{\sigma}^{(k)}$ are characterized by $k-1$ indices whose sequence is in inverse lexicographic order, and their actual values are given by

$$\sigma^{(k)}_{\xi_1,\xi_2,\ldots,\xi_{k-1}} = \sum_{\substack{r=1\\ r\ \text{odd}}}^{k-1} \sum_{\substack{\text{ordered } r\text{-choices } \{\varsigma_1,\varsigma_2,\ldots,\varsigma_r\}\\ \text{from } \{\zeta_1,\zeta_2,\ldots,\zeta_{k-1}\}}} \alpha_1\varsigma_1\varsigma_2\cdots\varsigma_r\Big|_{\zeta_1=s_{\xi_1},\zeta_2=s_{\xi_2},\ldots,\zeta_{k-1}=s_{\xi_{k-1}}}\,, \tag{9}$$

where α_1 is the ordinal index of ς_1, so that $\alpha_1 = 1$ if $\varsigma_1 = \zeta_1$, $\alpha_1 = 2$ if $\varsigma_1 = \zeta_2$, and so on. For a full derivation of these expressions, see Appendix B.

This treatment effectively transforms the original tensor equation of Eq. (2) into a singular-value problem on a matrix, without neglecting any hyperedge present in the original network, and allowing one to find the best bisection of a higher-order network. However, it is clear that the best partition, in general, may not consist of just two communities.

A natural idea is then to apply the same method to the communities resulting from an initial partitioning, checking whether hypermodularity increases when splitting those even further. However, care must be taken when doing so. The reason is that a key step in the derivation of Eqs. (7) and (9) relies on the vanishing of the sum of all elements of $\mathbf{B}$, as detailed in Appendix B. However, when bisecting an already existing community, this condition is no longer true, as one works with the subtensor corresponding to the nodes in the community, rather than with the data tensor of the entire network.

To find the correct way to deal with this situation, consider the change in hypermodularity that would result from a further split of a community C. This can be expressed as the sum of a positive term coming from the new split and a negative one, which is the contribution of the nodes in the community to the current value of the hypermodularity. From Eq. (3), it is clear that the former term consists of all and only the elements of the subtensor of C that correspond to nodes assigned to the same community. Similarly, the latter term is the sum of all the elements of the same subtensor. Thus, it is

$$\Delta Q = \frac{1}{k!m}\left(\sum_{k\text{-sets in } C} B_{v_1,\dots,v_k}\delta_{C_{v_1},C_{v_2}}\cdots\delta_{C_{v_{k-1}},C_{v_k}} - \sum_{k\text{-sets in } C} B_{v_1,\dots,v_k}\right). \tag{10}$$

Using the same expression for the product of deltas as before, but keeping the constant term that can no longer be neglected, results in an expression that can ultimately be manipulated in a way that still separates the variables into a vector equation of the form

$$\Delta Q = \frac{1}{2^{k-1}k!m}\mathbf{s}^{\mathrm{T}}\mathbf{E}'\boldsymbol{\sigma}^{(k)}\,, \tag{11}$$

where $\mathbf{E}'$ is the flattening of a modified community subtensor $\mathbf{B}'$, obtained by adding a correction to the elements on the main hyperdiagonal:

$$B'_{v_1,\dots,v_k} = \begin{cases} B_{v_1,\dots,v_k} - \sum_{s_{w_1}}\cdots\sum_{s_{w_{k-1}}} B_{w_1,\dots,w_{k-1},v_k} & \text{if } v_1 = v_2 = \cdots = v_k \\ B_{v_1,\dots,v_k} & \text{otherwise.} \end{cases} \tag{12}$$

For a formal derivation of the equation above, see Appendix C. Note that the correction terms vanish if one considers the whole network, as expected. Thus, one can always directly use Eq. (11), since it reduces to Eq. (7) when the network has not yet been partitioned.

Using the formalism just described, we build an algorithm that searches for the best partition of a k-uniform higher-order network into communities by carrying out repeated bisections. Following the approach of Refs. [26,47], we also include refinement steps, which we briefly describe below.

C. Refinement steps

After operating a bisection, we carry out a first Kernighan-Lin-type refinement [48] by computing how the hypermodularity would change if one were to switch the placement of each node from the community to which it was assigned to the other one. All the nodes are considered, and then, in a greedy fashion, the best move is accepted. The remaining nodes are then considered again, and the best move is accepted each time, until all the nodes have been evaluated. The best middle point throughout the procedure is then found. If it corresponds to an increase in hypermodularity, the moves up to that point are permanently accepted; otherwise, the state of the network is reverted to the one at the start of the procedure. The refinement is then repeated until no further improvement is obtained. To compute the change in hypermodularity that each move would yield, first note that changing the assignment of a single node i results in a single element of the vector $\mathbf{s}$ switching sign. Similarly, the vector $\boldsymbol{\sigma}^{(k)}$ only changes in the elements that correspond to combinations of $k-1$ nodes involving node i. Thus, we can write $\boldsymbol{\sigma}^{(k)} \to \boldsymbol{\sigma}'^{(k)} = \boldsymbol{\sigma}^{(k)} + \boldsymbol{\Delta\sigma}^{(k)}$ and $\mathbf{s} \to \mathbf{s}' = \mathbf{s} + \boldsymbol{\Delta s}$, where all the elements of $\boldsymbol{\Delta s}$ are 0, except for the i-th one, which is equal to $-2s_i$. Then, write the hypermodularity of the partition after the change as

$$\begin{aligned} Q' &= \frac{1}{2^{k-1}k!m}(\mathbf{s}+\boldsymbol{\Delta s})^{\mathrm{T}}\mathbf{E}'(\boldsymbol{\sigma}^{(k)}+\boldsymbol{\Delta\sigma}^{(k)}) \\ &= \frac{1}{2^{k-1}k!m}(\mathbf{s}^{\mathrm{T}}\mathbf{E}'\boldsymbol{\sigma}^{(k)} + \mathbf{s}^{\mathrm{T}}\mathbf{E}'\boldsymbol{\Delta\sigma}^{(k)} + \boldsymbol{\Delta s}^{\mathrm{T}}\mathbf{E}'\boldsymbol{\sigma}^{(k)} \\ &\quad + \boldsymbol{\Delta s}^{\mathrm{T}}\mathbf{E}'\boldsymbol{\Delta\sigma}^{(k)})\,. \end{aligned} \tag{13}$$

From this equation, it follows that

$$\begin{aligned} \Delta Q &= \frac{1}{2^{k-1}k!m}(\mathbf{s}^{\mathrm{T}}\mathbf{E}'\boldsymbol{\Delta\sigma}^{(k)} + \boldsymbol{\Delta s}^{\mathrm{T}}\mathbf{E}'\boldsymbol{\sigma}^{(k)} + \boldsymbol{\Delta s}^{\mathrm{T}}\mathbf{E}'\boldsymbol{\Delta\sigma}^{(k)}) \\ &= \frac{1}{2^{k-1}k!m}(\mathbf{s}'^{\mathrm{T}}\mathbf{E}'\boldsymbol{\Delta\sigma}^{(k)} + \boldsymbol{\Delta s}^{\mathrm{T}}\mathbf{E}'\boldsymbol{\sigma}^{(k)})\,. \end{aligned} \tag{14}$$

This form is particularly convenient, because the vector $\boldsymbol{\sigma}^{(k)}$ is already known from the previous step. Thus, the second term can be readily computed as the product of the i-th row of $\mathbf{E}'$ and $\boldsymbol{\sigma}^{(k)}$, multiplied by $-2s_i$, and the first term can be calculated similarly, by accounting for all and only the combinations of factors that include node i.

After all existing communities have been evaluated for further splits, and the local node-level refinement has been considered, we perform a global node-level optimization. The structure of this step is again that of an iterated greedy algorithm that evaluates the change in hypermodularity resulting from moving each node from the community to which it is currently assigned to any other possible community. As before, the best move is accepted, and the evaluation is repeated

for the remaining nodes, until all nodes have been moved. Then, the middle point in the sequence of moves corresponding to the largest change in hypermodularity is found, and all the moves up to that point are permanently accepted if such change is positive. The step is repeated until it produces an improvement in hypermodularity.

The change in hypermodularity caused by moving node i to community α consists of two contributions. The first is a negative one, which is due to all and only the combinations of k nodes in the original community of i such that at least one of them is node i itself. The second is a positive change that is due to all and only the combinations of k-nodes in community α, with the temporary inclusion of node i within it, such that at least one of them is node i itself. Both are sums of the elements of $\mathbf{B}$ corresponding to the combinations of nodes taken, divided by $k!m$, and with the sign of the negative contribution changed.

Finally, if the current partition consists of more than one community, a community-level optimization is carried out. As for the previous two steps, also this is an iterated greedy algorithm, repeated so long as the value of hypermodularity increases. The moves considered are the mergings of pairs of communities, akin to the main procedure of the Louvain algorithm [20]. Merging community α and community β causes a change in hypermodularity that is due to all and only the combinations of k nodes such that at least one is in community α and one is in community β. As in the previous step, each combination contributes an amount equal to the corresponding element of $\mathbf{B}$ divided by $k!m$. This means that the potential changes in hypermodularity can be represented as a matrix whose dimension is the current number of communities. When a move is accepted, it is convenient to take all the nodes belonging to the community with the higher index and assign them to the other one. This way, for all subsequent moves, the only matrix elements that change are those involving the community with the lower index, and, because of symmetry, only half a row and half a column need to be updated.

An example of the functioning of our method on a small 3-uniform hypergraph with 12 nodes is provided in Appendix D.

D. Numerical validation

To demonstrate the applicability of the method, we implemented the algorithm described above, and analyzed both synthetic and real-world higher-order networks. We started with the study of ensembles of Erdős-Rényi-like random k-uniform hypergraphs, to determine the behavior of hypermodularity on networks without a well-defined community structure. In these networks, every possible hyperedge of size k exists independently with the same probability p. We ran the algorithm on networks of size 10, 20, 50, and 100, with $k = 3$ and $k = 4$. The behavior of the average maximum hypermodularity as a function of p, shown in Fig. 1, shows that, on connected networks, the best partition never corresponds, on average, to a hypermodularity greater than 0.2. However, as the connection probability decreases, and the networks start fragmenting into separate connected components, the maximum value of the hypermodularity increases even as high as 0.8. At first, this may seem evidence of a problem with the measure, as there appear to exist partitions of completely random networks whose hypermodularity is close to the theoretical maximum of 1. However, such values are actually spurious and they are, in fact, due to the fragmentation of the networks. To understand how this happens, consider a limit case of a hypergraph formed by m isolated hyperedges. Its maximum hypermodularity, which is found by assigning all the nodes of each edge to a separate community, is $1 - \frac{(k-1)!}{(km)^{k-1}}$, which, for large m, is approximately 1. However, the hypermodularity of the individual components, which consist of a single edge, is identically 0. This shows that one must be very careful in using hypermodularity on networks that are not connected, and that in such cases the right approach is to treat each component as an independent network. Thus, our results also indicate that concerns of overfitting, as have been occasionally voiced also for traditional modularity, are mostly unfounded, and one can take the value of 0.2 as a sort of physiological expected maximum hypermodularity of a random network, due to the non-vanishing probability that, even in a random structure, some sets of nodes may be more densely linked internally than they are between each other.

We then tested the method on real-world data sets, using contact information from a primary school [49,50] and from a high school [50,51]. In the former case, the pupils are classified according to their grade, with two separate classes running for each year from 1 to 5. The teachers, one per class, were also involved in the study. The high-school students, instead, are classified according to the topic of the classes they were following. Specifically, the data set contains three classes of biology, three of mathematics and physics, two of chemistry and one of psychology. In both cases, we detected the community structure for edges of sizes 2 to 5, since no edges of larger size were recorded.

The primary school results, shown in Fig. 2, are quite revealing. When we use $k = 2$, which corresponds to a traditional network, the algorithm divides the students into 5 communities, each corresponding to a different grade, with the exception that all the pupils of one of the classes of year 4 are assigned to the same community as the pupils of year 5. This partition, with a modularity of 0.3066, is consistent with younger children tending to separate by age in their individual interactions, and with such boundaries becoming less rigid as they grow up. The communities of 3-people interactions ($k = 3$) reveal a similar situation. Year-3 students keep to themselves, whereas one entire class of year-1 pupils joins the two of year-2 students to form a community. The pupils of the last two years are together in the same community. This division, with a hypermodularity of 0.71, is again consistent with the dynamics just described. Year-3 students consider the two lower grades too young to form social interactions with, and they are themselves considered too young by the two higher grades. The data for $k = 4$ show an intriguing division, with a hypermodularity of 0.7632. Years 2 and 3 are joined in a single community, and years 1, 4, and 5 form another one, with two individual isolated students. A possible explanation for the former is that in looser groups of four people, it is easier to include one or two members of different age without introducing too much disparity of maturity than it is in tighter groups of 2 or 3 people. The latter community is perhaps more striking. The age difference between the individuals in year 1 and those in year 4 and year 5 is of

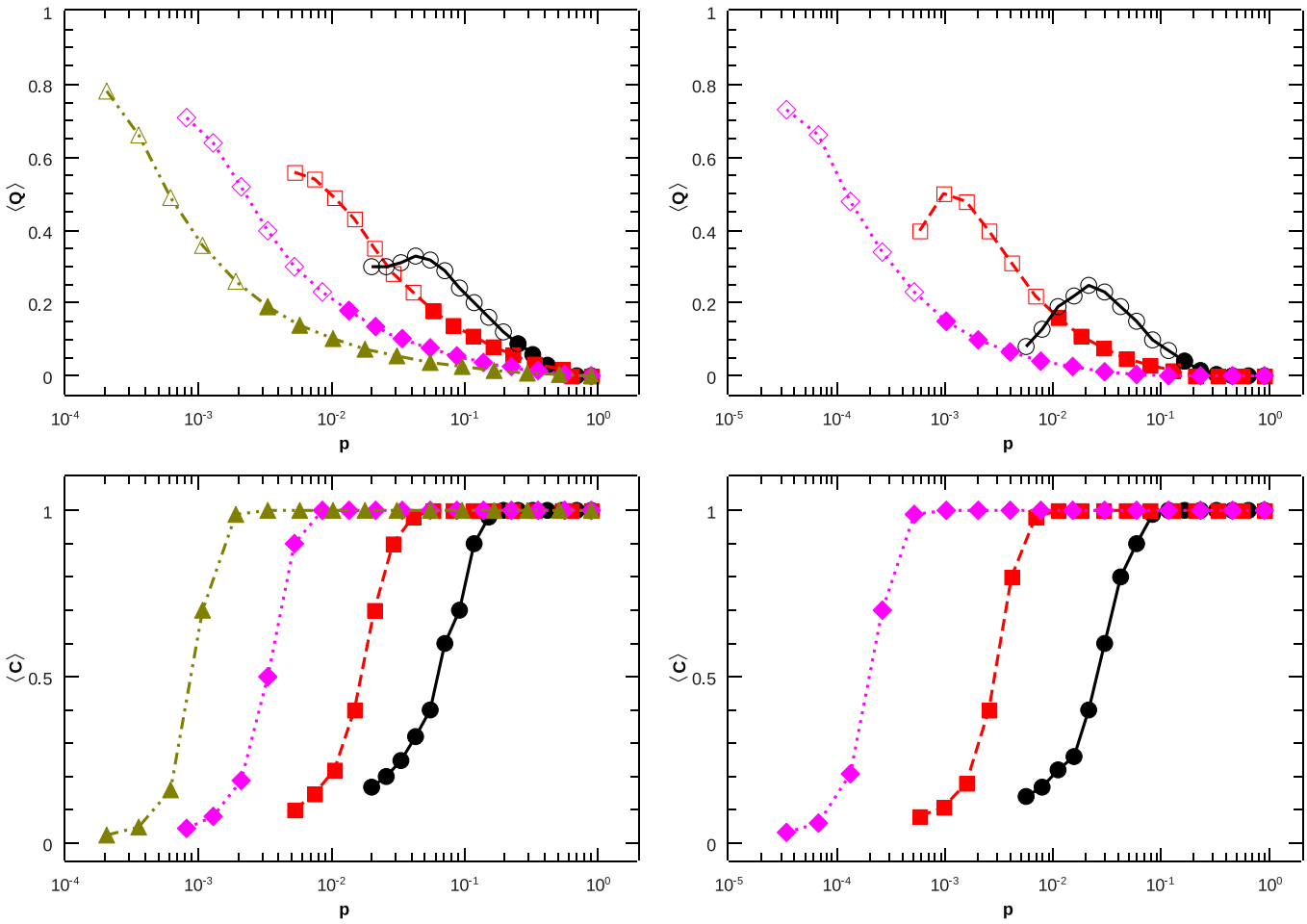


FIG. 1. Maximum hypermodularity in random hypergraphs is never above **0.2**. So long as the networks remain connected, the average maximum hypermodularity $\langle Q \rangle$ of random k-uniform hypergraphs remains below 0.2, as shown here for $k = 3$ (left-hand top panel) and $k = 4$ (right-hand top panel). As the connection probability p decreases, the networks start to fragment, and the value of the maximum hypermodularity spuriously increases (empty symbols). The bottom panels show the average fraction of nodes in the components of the networks $\langle C \rangle$. For all panels, black circles and solid lines correspond to $N = 10$, red squares and dashed lines to $N = 20$, purple diamonds and dotted lines to $N = 50$, and green triangles and dotted-and-dashed lines to $N = 100$.

3 to 4 years, and this community seems to result from the presence of siblings, who join the age-based social group of their brothers and sisters. Indeed, the average age gap between siblings in the period where the data were collected was, in fact, 3.5 years [52]. Finally, for $k = 5$, the network is almost completely fragmented, and the only nodes left with some edges separate into communities that are either uniform, or with at most one grade of difference between the members. Note that the network of 5-people interactions consists of 6 components, 5 of which contain a single hyperedge, and thus have a vanishing hypermodularity. The division of the only nontrivial component has a hypermodularity of 0.662.The analysis of the high-school data set, shown in Fig. 3, also provides insights into the social dynamics of the groups of students. When one considers only dyadic interactions, the network separates into communities that are completely identified not only by the topic studied, but also by the specific class in which the students are enrolled. This partition has a modularity of 0.5915, and the only exception is that the mathematics and physics students form a single community, rather than splitting into their three classes. This evidently disproves the stereotype that sees mathematics and physics students as socially inept. In fact, in this study, they are the only group of people who socialize across classes, albeit in the same topic. When we take $k = 3$, the network is very highly modular, with 3 large communities plus an individual node and a hypermodularity of 0.8313. Here, the communities are entirely topic-based, but the students of psychology are joined with those of mathematics and physics. A possible explanation for this is that both physicists and psychologists study complex systems, even though using different means of representation. Thus, they have a larger affinity than one may imagine at first thought. At the next larger size of edges, $k = 4$, the meta-group of mathematics-physics-psychology students joins that of chemistry ones, with the notable exception of 9 mathematics-and-physics students who are now assigned to the community of biology. Also, the network starts fragmenting, with an isolated clique of 5 biology students. This partition of the large component has a hypermodularity of 0.8527, and it is mostly caused by a few students who act as bridges between disciplines, and whose importance increases as the network has substantially fewer edges. For example,

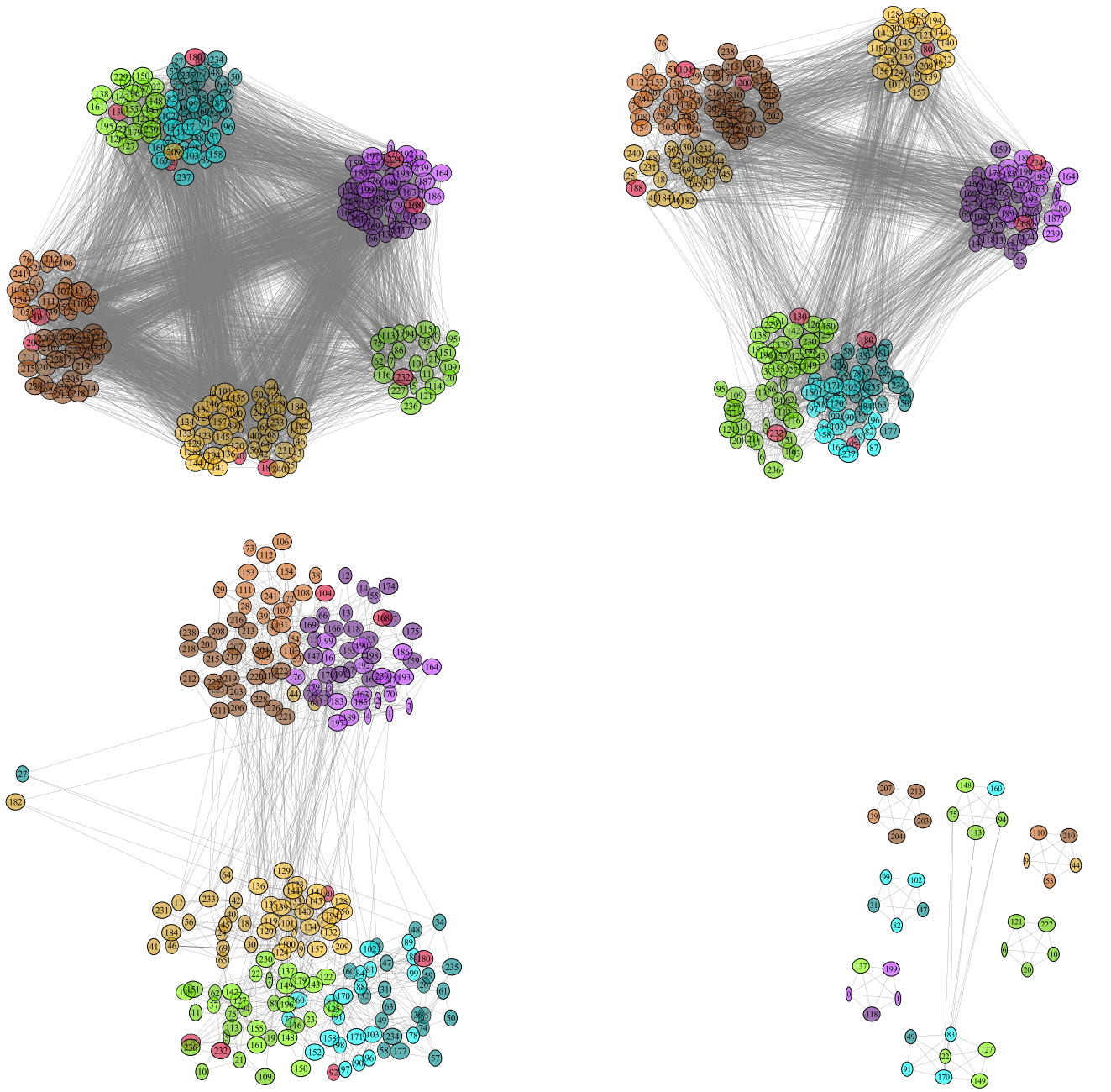


FIG. 2. Higher-order communities of primary-school pupils reveal underlying social dynamics. The communities of 2-people and 3-people interactions (top-left and top-right panels, respectively) show that tight-group interactions between primary-school pupils mostly occur within age groups. Communities of looser groups of 4 people (bottom-left) blur this line, with younger siblings being accepted within groups of older children. Finally, interactions of 5 pupils at a time (bottom-right) are rare, and possibly stronger, thereby restoring the age separation seen for $k = 2$ and $k = 3$. In all panels, yellow nodes correspond to pupils in year 1, brown nodes to pupils in year 2, violet nodes to pupils in year 3, green nodes to pupils in year 4, blue nodes to pupils in year 5, and red nodes to teachers. Each year consists of two separate classes, signified by the shade of the color. Note that in preparing the figure, hyperedges ($k > 2$) have been projected onto cliques for ease of visualization.

psychology student 325 is well connected with other psychology students and with mathematics-physics ones, chemistry student 284 is highly connected with a clique of psychology students and with numerous other chemistry ones, and mathematics-physics students 125 and 176 have several links to other students in the same class as well as to biology ones. Finally, when $k = 5$ very few nodes remain with some edges. Five of them are a clique of chemistry students, and the other three components consist entirely of students of the same topic (two components for biology and one for chemistry). This suggests that these components are study groups, and the larger communities identified within them are their core members. It should be noted, nonetheless, that only the larger biology component is reasonably modular, with its partition corresponding to a hypermodularity of 0.3319, whereas both other components have a hypermodularity of 0.1723, which is below what one would expect from a random hypergraph, as discussed above.

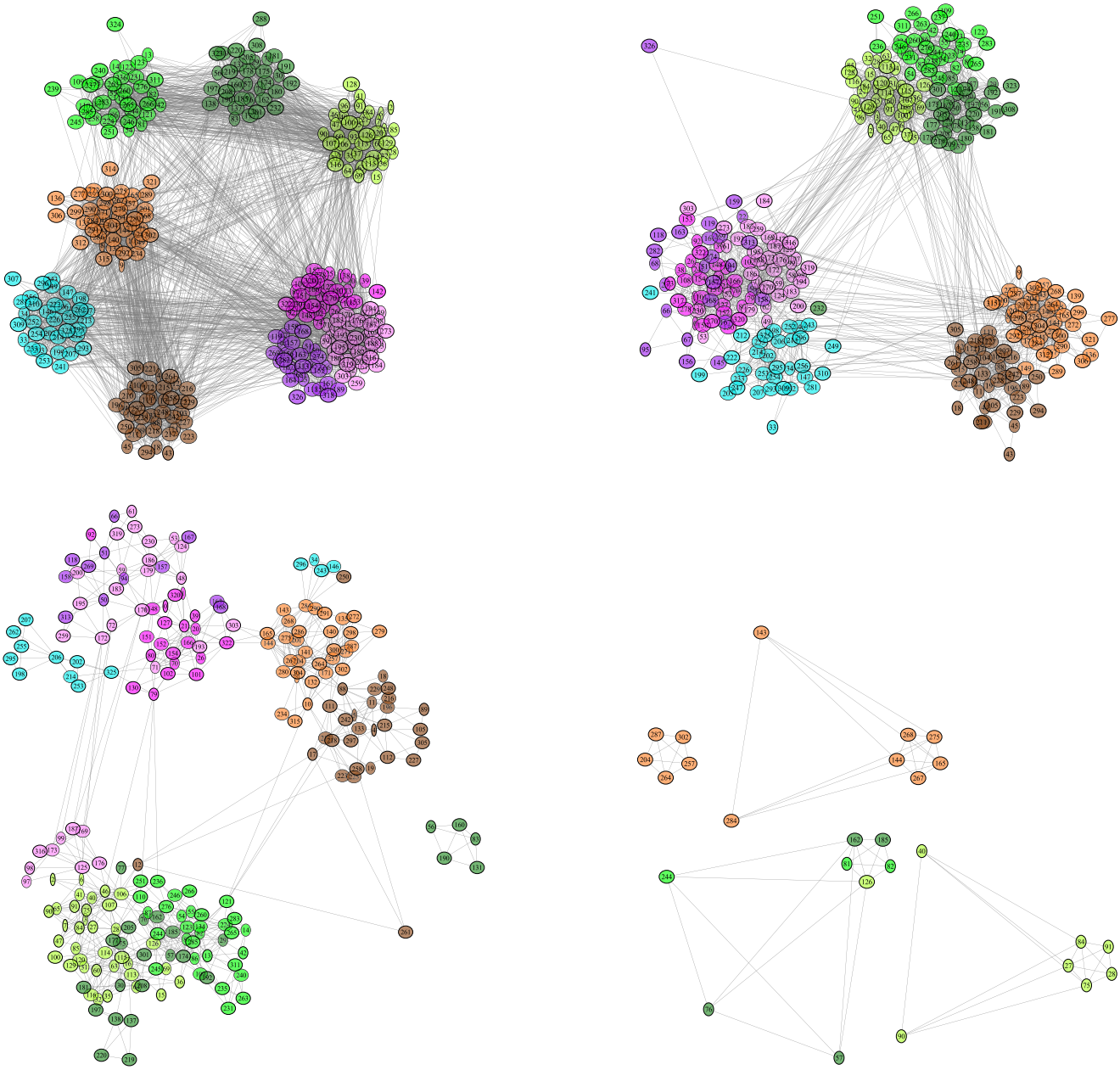


FIG. 3. Higher-order communities of high-school students provide insights into closeness of interests. The communities of dyadic interactions between high-school students (top-left) show a strong division between topics (colors) and classes (shades), with the exception of mathematics and physics students (violet), who interact across classes, challenging the stereotypical opinion of more mathematically minded people being unable to form meaningful social contacts. The communities of groups of 3 people (top-right) reveal the proximity of thought between psychology and mathematics-physics students, who are now joined in a single community. The 4-people interactions (bottom-left) extend this insight, revealing how biology students (green) are substantially isolated, whereas almost all the other students belong to the same community, with the exception of a small separate clique. Finally, at $k = 5$, the network consists of only 4 small study groups. In all panels, green nodes correspond to biology students, violet nodes to mathematics and physics students, brown nodes to chemistry students, and blue nodes to psychology students. All topics except psychology include more than one class, signified by the shade of the color. Note that in preparing the figure, hyperedges ($k > 2$) have been projected onto cliques for ease of visualization.

III. CONCLUSIONS

In this article, we have considered the task of detecting communities in networks with interactions beyond pairwise ones. Our way to address this problem is by defining a hypermodularity function for hypergraphs using a formalism that is conducive to the use of spectral methods to find the bisection of the network that maximizes it. Specifically, we have shown that this can be achieved by the use of higher-order singular value decomposition, if one rewrites the condition that the elements of a given set of nodes all belong to the same community as a separable expression of spin-like variables. To do so, we demonstrated a closed combinatorial expression that allows one to ultimately turn the hypermodularity equation into a vector form.

After implementing an algorithm based on these ideas, we applied it to random hypergraphs, finding a physiological value of maximum hypermodularity that one can expect to find in networks without an imposed community structure. This is due to the nonzero probability that, even placing all edges at random, one can end up with modules of nodes whose internal edges are denser than their external ones. Notably, the hypermodularity of random hypergraphs never exceeded 0.2, suggesting that this value can be considered as a rough threshold to determine the presence of strong communities.

Note that this result offers a natural explanation of the often-mentioned issue of "overfitting" in modularity and related measures. The existence of this problem is usually explained by noting that, occasionally, networks generated randomly from an Erdős-Rényi ensemble can admit partitions with nonzero modularity. Indeed, we have shown that values of hypermodularity as high as 0.8 can be found when the starting network is not connected. However, we strongly believe that applying any community detection method to a disconnected network is a methodological error. Since no connections exist between components, any communities within the network will have to be confined each to a single component. But then, rather than the overall value of the modularity, one should consider the components individually, and measure the modularity of each by itself. Additionally, the rough limit of 0.2 for the maximum hypermodularity of a partition of a random hypergraph can be interpreted as related to the probability that, placing a number of edges at random, a set of more densely connected nodes arises. Also note that this value, and indeed the entire trend shown in the top panels of Fig. 1, is very much similar to what has been measured in traditional graphs [53], showing that the behavior of hypermodularity is consistent with its originating principles.

In the more than two decades since the seminal article in Ref. [53], these considerations have led to the creation of numerous methods based on Bayesian paradigms, and most notably on the use of stochastic block models. These are a class of models to build random graphs based on the idea of fixing a number of sets of nodes as communities, and placing edges internal and external to such modules with some chosen probabilities. These methods constitute an alternative to other community-detection approaches. However, they also present notable problems, such as the tendency to produce poor partitions, sometimes including disconnected communities, on some networks [54].

In turn, we do not expect hypermodularity to be exempt from drawbacks either. In fact, we believe that it is likely affected by a higher-order version of the resolution limit [55]. However, we also conjecture that the resolution limit may have a lower importance than it has on traditional networks, due to the combinatorial factor weighing the product of degrees in the hypermodularity equation. In the near future, we plan to investigate this question in detail and quantitatively evaluate the limitations of hypermodularity, potentially addressing them via suitable modifications of its formulation.

By maximizing hypermodularity in its current form, our algorithm was able to detect communities at different levels of collective interaction when applied to real-world data. The communities thus identified can all be explained by assuming reasonable mechanisms for their formation. In fact, it can be argued that the communities found on social contact networks represent similarity of thought processes and reflect the biases of the individuals. Thus, we believe that our formulation of hypermodularity and our algorithm constitute powerful tools for the analysis of networks with higher-order interactions, and are highly valuable methods for extracting hidden information from data sets of diverse nature.

ACKNOWLEDGMENTS

The author acknowledges fruitful discussions with Ekaterina Vasilyeva and Stefano Boccaletti. This work was supported by the Bulgarian Ministry of Education and Science under Project No. BG-RRP-2.004-0006-C02.

DATA AVAILABILITY

The data that support the findings of this article are openly available [56]. The code implementing the algorithm described in this article and used to generate the data is openly available [57].

APPENDIX A: IDENTITY OF THE FLATTENINGS

To formally show why all the flattenings are identical, start by explicitly writing the k-th standard flattening of $\mathbf{B}$:

$$\mathbf{E}^{(k)} = \begin{pmatrix} B_{1,1,\dots,1} & B_{2,1,\dots,1} & \cdots & B_{N,N,\dots,1} \\ B_{1,1,\dots,2} & B_{2,1,\dots,2} & \cdots & B_{N,N,\dots,2} \\ \vdots & \vdots & \ddots & \vdots \\ B_{1,1,\dots,N} & B_{2,1,\dots,N} & \cdots & B_{N,N,\dots,N} \end{pmatrix}. \tag{A1}$$

This equation shows that the row index of $\mathbf{E}^{(k)}$ corresponds to the k-th index of $\mathbf{B}$. Also, the column index of $\mathbf{E}^{(k)}$, which numerically ranges from 1 to N^{k-1}, is equal to the positional number of the corresponding choice of values for the remaining indices of $\mathbf{B}$ in an inverse lexicographic order. In other words, each element $E^{(k)}_{i,j}$ corresponds to the element of $\mathbf{B}$ whose k-th index is equal to i and whose other indices $v_1, \dots, v_{k-1}$ satisfy the equation

$$j = v_1 N^0 + (v_2 - 1)N^1 + (v_3 - 1)N^2 + \cdots + (v_{k-1} - 1)N^{k-2}. \tag{A2}$$

Effectively, the equation above provides the unique $k-1$ digits of the column numbers of the elements of $\mathbf{E}^{(k)}$ expressed in a base-N positional system.

Now consider the generic x-th standard flattening of $\mathbf{B}$:

$$\mathbf{E}^{(x)} = \begin{pmatrix} B_{1,1,\dots,1,1,1,\dots,1} & B_{2,1,\dots,1,1,1,\dots,1} & \cdots & B_{N,N,\dots,N,1,N,\dots,N} \\ B_{1,1,\dots,1,2,1,\dots,1} & B_{2,1,\dots,1,2,1,\dots,1} & \cdots & B_{N,N,\dots,N,2,N,\dots,N} \\ \vdots & \vdots & \ddots & \vdots \\ B_{1,1,\dots,1,N,1,\dots,1} & B_{2,1,\dots,1,N,1,\dots,1} & \cdots & B_{N,N,\dots,N,N,N,\dots,N} \end{pmatrix}. \tag{A3}$$

Here, the row index of $\mathbf{E}^{(x)}$, which is the index of $\mathbf{B}$ that changes between the elements in each column, is the x-th one. So, given an element $E^{(x)}_{i,j}$, it corresponds to the element of $\mathbf{B}$ whose x-th index is equal to i and whose other indices are such that

$$j = v_1 N^0 + (v_2 - 1)N^1 + \cdots + (v_{x-1} - 1)N^{x-2} + (v_{x+1} - 1)N^{x-1} + \cdots + (v_k - 1)N^{k-2} \,. \tag{A4}$$

Then, fix a row index l and a column index m for the two flattenings, and take their elements $E^{(k)}_{l,m}$ and $E^{(x)}_{l,m}$. The former maps to the element of the data tensor $B_{v_1,v_2,\ldots,v_{x-1},v_x,v_{x+1},\ldots,v_{k-1},l}$ whose last index is equal to l and whose other indices correspond to the unique decomposition of m in powers of N. The other, in the same way, corresponds instead to $B_{v_1,v_2,\ldots,v_{x-1},l,v_{x+1},\ldots,v_k}$. Thus, the indices of the latter element of $\mathbf{B}$ are a permutation of those of the former. But, since $\mathbf{B}$ is hypersymmetric, $B_{v_1,v_2,\ldots,v_{x-1},v_x,v_{x+1},\ldots,v_{k-1},l} = B_{v_1,v_2,\ldots,v_{x-1},l,v_{x+1},\ldots,v_k}$, which, in turn, means that, for any choice of l and m, $E^{(x)}_{l,m} = E^{(k)}_{l,m}$ for all x. Thus, all the standard flattenings of the data tensor of a simple hypergraph are identical, and in the following we use the k-th ones without loss of generality, omitting explicit mention of the dimension used for their construction.

APPENDIX B: BUILDING A VECTOR EQUATION

To turn Eq. (3) into a vector equation, first use Eq. (5) to explicitly rewrite the product chain for $k = 3$:

$$\delta_{C_{v_1},C_{v_2}}\delta_{C_{v_2},C_{v_3}} = \tfrac{1}{2}(s_1 s_2 + 1)\tfrac{1}{2}(s_2 s_3 + 1) = \tfrac{1}{4}\left(s_1 s_2^2 s_3 + s_1 s_2 + s_2 s_3 + 1\right) . \tag{B1}$$

Note that in the equation above, s_1, s_2, and s_3 indicate the community assignments of the nodes chosen as v_1, v_2, and v_3, and *not* those of nodes 1, 2, and 3. Thus, for $k = 3$, Eq. (3) becomes

$$\begin{aligned} Q &= \frac{1}{3!m}\sum\nolimits_{\{v_1,v_2,v_3\}}\left(B_{v_1,v_2,v_3}\prod_{\substack{i,j=1\\ i\neq j}}^{3}\delta_{C_{v_i},C_{v_j}}\right) = \frac{1}{3!m}\sum\nolimits_{\{v_1,v_2,v_3\}}\left[B_{v_1,v_2,v_3}\frac{1}{4}\left(s_1 s_2^2 s_3 + s_1 s_2 + s_2 s_3 + 1\right)\right] \\ &= \frac{1}{3!m}\sum\nolimits_{\{v_1,v_2,v_3\}}\left[B_{v_1,v_2,v_3}\frac{1}{4}(s_1 s_3 + s_1 s_2 + s_2 s_3 + 1)\right] , \end{aligned} \tag{B2}$$

where, in the last line, we have used $s_i^2 = 1$. But the sum above runs over all combinations of 3 nodes, which means that all the elements of $\mathbf{B}$ are considered. In turn, this means that, since the sum of all the elements of $\mathbf{B}$ vanishes by construction, we can drop any constant term from the equation, obtaining

$$Q = \frac{1}{3!m}\sum\nolimits_{\{v_1,v_2,v_3\}}\left[B_{v_1,v_2,v_3}\frac{1}{4}(s_1 s_3 + s_1 s_2 + s_2 s_3)\right] = \frac{1}{3!m}\sum\nolimits_{\{v_1,v_2,v_3\}}\left\{B_{v_1,v_2,v_3}\frac{1}{4}[s_3(s_1 + s_2) + s_1 s_2]\right\} . \tag{B3}$$

But, again because of the sum and of the nature of $\mathbf{B}$, the index of s_1 in the last term above is a dummy index. Thus, we can change that s_1 to s_3, and get

$$Q = \frac{1}{3!m}\sum\nolimits_{\{v_1,v_2,v_3\}}\left[B_{v_1,v_2,v_3}\frac{1}{4}s_3(s_1 + 2s_2)\right] , \tag{B4}$$

which we rewrite as

$$Q = \frac{1}{3!m}\sum\nolimits_{\{v_1,v_2,v_3\}} B_{v_1,v_2,v_3} S^{(3)} \,, \tag{B5}$$

where we have put

$$S^{(3)} = \tfrac{1}{4}s_3(s_1 + 2s_2) \,. \tag{B6}$$

This equation conveniently separates the three spin-like variables into a product of one of them and an expression featuring the other two. Thus, the equation for hypermodularity can be rewritten as

$$Q = \frac{1}{4 \cdot 3!m}\mathbf{s}^{\mathrm{T}}\mathbf{E}\boldsymbol{\sigma}^{(3)} \,, \tag{B7}$$

where $\mathbf{s}$ is a vector containing the community assignments of all nodes, and $\boldsymbol{\sigma}^{(3)}$ is an N^2-dimensional vector whose elements are the values of the expression $s_1 + 2s_2$ computed for all possible pairs of nodes in the same order as those corresponding to the columns of $\mathbf{E}$.

To find a general-use formula, repeat the calculation for $k = 4$, by writing the chain product of deltas in terms of spin-like variables, dropping any constants and replacing dummy indices with the 4th one:

$$\begin{aligned} S^{(4)} &= \delta_{C_{v_1},C_{v_2}}\delta_{C_{v_2},C_{v_3}}\delta_{C_{v_3},C_{v_4}} = \tfrac{1}{2}(s_1 s_2 + 1)\tfrac{1}{2}(s_2 s_3 + 1)\tfrac{1}{2}(s_3 s_4 + 1) \\ &= \tfrac{1}{8}\left(s_1 s_2^2 s_3^2 s_4 + s_1 s_2^2 s_3 + s_1 s_2 s_3 s_4 + s_1 s_2 + s_2 s_3^2 s_4 + s_2 s_3 + s_3 s_4 + 1\right) \\ &= \tfrac{1}{8}[s_4(s_1 + s_2 + s_3 + s_1 s_2 s_3) + s_1 s_2 + s_1 s_3 + s_2 s_3 + 1] \\ &\to \tfrac{1}{8}[s_4(s_1 + s_2 + s_3 + s_1 s_2 s_3) + s_1 s_2 + s_1 s_3 + s_2 s_3] \end{aligned}$$

$$\rightarrow \tfrac{1}{8}[s_4(s_1+s_2+s_3+s_1s_2s_3)+s_4s_2+s_4s_3+s_4s_3]$$
$$\rightarrow \tfrac{1}{8}s_4(s_1+2s_2+3s_3+s_1s_2s_3), \tag{B8}$$

where we have used a right arrow, rather than the equal sign, to indicate functional, but not formal, equality. Similarly, for $k=5$, one gets

$$S^{(5)} = \tfrac{1}{16}s_5(s_1+2s_2+3s_3+4s_4+s_1s_2s_3+s_1s_2s_4+s_1s_3s_4+2s_2s_3s_4), \tag{B9}$$

and for $k=6$ it is

$$S^{(6)} = \tfrac{1}{32}s_6(s_1+2s_2+3s_3+4s_4+5s_5+s_1s_2s_3+s_1s_2s_4+s_1s_2s_5+s_1s_3s_4+s_1s_3s_5+s_1s_4s_5+2s_2s_3s_4 + 2s_2s_3s_5+2s_2s_4s_5+3s_3s_4s_5+s_1s_2s_3s_4s_5). \tag{B10}$$

It becomes then clear that the vector expression for the hypermodularity of a bipartition, which we wrote in Eq. (B7) for $k=3$, can be generalized to any k in the form given by Eqs. (7) and (9), which we rewrite here for the sake of completeness:

$$Q = \frac{1}{2^{k-1}k!m}\mathbf{s}^{\mathrm{T}}\mathbf{E}\boldsymbol{\sigma}^{(k)} \tag{B11}$$

$$\sigma^{(k)}_{\xi_1,\xi_2,\dots,\xi_{k-1}} = \sum_{\substack{r=1\\ r\ \text{odd}}}^{k-1} \sum_{\substack{\text{ordered } r\text{-choices } \{\varsigma_1,\varsigma_2,\dots,\varsigma_r\}\\ \text{from } \{\zeta_1,\zeta_2,\dots,\zeta_{k-1}\}}} \alpha_1\varsigma_1\varsigma_2\cdots\varsigma_r \Bigg|_{\zeta_1=s_{\xi_1},\zeta_2=s_{\xi_2},\dots,\zeta_{k-1}=s_{\xi_{k-1}}}, \tag{B12}$$

where α_1 is the ordinal index of ς_1, so that $\alpha_1=1$ if $\varsigma_1=\zeta_1$, $\alpha_1=2$ if $\varsigma_1=\zeta_2$, and so on.

Note that this demonstration not only offers a straightforward way to employ spectral methods in search of the best partition of a higher-order network into two communities, but it also shows that in classification tasks, the assignment corresponding to the first higher-order singular vector of a data tensor is indeed the best one, and not just an approximation of it, in turn providing an explanation of the success of methods based on higher-order SVD in machine learning.

APPENDIX C: REPEATED BISECTIONS

To find the expression for the correction to use when repeating the bisection process, start from Eq. (10), which we rewrite here for convenience:

$$\Delta Q = \frac{1}{k!m}\left(\sum_{k\text{-sets in } C} B_{v_1,\dots,v_k}\delta_{C_{v_1},C_{v_2}}\cdots\delta_{C_{v_{k-1}},C_{v_k}} - \sum_{k\text{-sets in } C} B_{v_1,\dots,v_k}\right). \tag{C1}$$

Now, write the product of deltas in terms of the spin-like variables. However, note that, for the reasons discussed in Sec. II B, the constant term can no longer be neglected. This results in

$$\Delta Q = \frac{1}{k!m}\left[\frac{1}{2^{k-1}}\sum_{s_{v_1}}\cdots\sum_{s_{v_k}} B_{v_1,\dots,v_k}\left(s_{v_k}f\left(s_{v_1},\dots,s_{v_{k-1}}\right)+1\right) - \sum_{s_1}\cdots\sum_{s_{v_k}} B_{v_1,\dots,v_k}\right], \tag{C2}$$

where we have explicitly expanded the sums and f is a polynomial in the first $k-1$ spin-like variables, as in Eq. (B6) and in Eqs. (B8)–(B10). Then, expanding the product, we obtain

$$\begin{aligned}
\Delta Q &= \frac{1}{k!m}\left(\frac{1}{2^{k-1}}\sum_{s_{v_1}}\cdots\sum_{s_{v_k}} B_{v_1,\dots,v_k}s_{v_k}f\left(s_{v_1},\dots,s_{v_{k-1}}\right) + \frac{1}{2^{k-1}}\sum_{s_{v_1}}\cdots\sum_{s_{v_k}} B_{v_1,\dots,v_k} - \sum_{s_{v_1}}\cdots\sum_{s_{v_k}} B_{v_1,\dots,v_k}\right)\\
&= \frac{1}{k!m}\left(\frac{1}{2^{k-1}}\sum_{s_{v_1}}\cdots\sum_{s_{v_k}} B_{v_1,\dots,v_k}s_{v_k}f\left(s_{v_1},\dots,s_{v_{k-1}}\right) - \frac{2^{k-1}-1}{2^{k-1}}\sum_{s_{v_1}}\cdots\sum_{s_{v_k}} B_{v_1,\dots,v_k}\right)\\
&= \frac{1}{2^{k-1}k!m}\left[\sum_{s_{v_1}}\cdots\sum_{s_{v_k}} B_{v_1,\dots,v_k}s_{v_k}f\left(s_{v_1},\dots,s_{v_{k-1}}\right) - (2^{k-1}-1)\sum_{s_{v_1}}\cdots\sum_{s_{v_k}} B_{v_1,\dots,v_k}\right]\\
&= \frac{1}{2^{k-1}k!m}\left[\sum_{s_{v_1}}\cdots\sum_{s_{v_k}} B_{v_1,\dots,v_k}s_{v_k}f\left(s_{v_1},\dots,s_{v_{k-1}}\right)\right.\\
&\quad\left. - (2^{k-1}-1)\sum_{s_{v_1}}\cdots\sum_{s_{v_k}}\delta_{v_1,v_2}\cdots\delta_{v_{k-1},v_k}\sum_{s_{w_1}}\cdots\sum_{s_{w_{k-1}}} B_{w_1,\dots,w_{k-1},v_k}\right]\\
&= \frac{1}{2^{k-1}k!m}\left[\sum_{s_{v_1}}\cdots\sum_{s_{v_k}} B_{v_1,\dots,v_k}s_{v_k}f\left(s_{v_1},\dots,s_{v_{k-1}}\right)\right.\\
&\quad\left. - (2^{k-1}-1)\sum_{s_{v_1}}\cdots\sum_{s_{v_k}}\delta_{v_1,v_2}\cdots\delta_{v_{k-1},v_k}\sum_{s_{w_1}}\cdots\sum_{s_{w_{k-1}}} B_{w_1,\dots,w_{k-1},v_k}\frac{s_{v_k}f\left(s_{v_1},\dots,s_{v_{k-1}}\right)}{2^{k-1}-1}\right],
\end{aligned} \tag{C3}$$

where, in the last line, we have used the fact that $s_{v_k} f(s_{v_1}, \dots, s_{v_{k-1}}) = 2^{k-1} - 1$ when $v_1 = v_2 = \cdots = v_k$. But then, it is

$$\Delta Q = \frac{1}{2^{k-1} k! m} \sum_{s_{v_1}} \cdots \sum_{s_{v_k}} s_{v_k} \left(B_{v_1,\dots,v_k} - \delta_{v_1,v_2} \cdots \delta_{v_{k-1},v_k} \sum_{s_{w_1}} \cdots \sum_{s_{w_{k-1}}} B_{w_1,\dots,w_{k-1},v_k} \right) f(s_{v_1}, \dots, s_{v_{k-1}}) , \tag{C4}$$

which means that we can express the change in hypermodularity as in Eqs. (11) and (12), which we rewrite below:

$$\Delta Q = \frac{1}{2^{k-1} k! m} \mathbf{s}^{\mathrm{T}} \mathbf{E}' \boldsymbol{\sigma}^{(k)} \tag{C5}$$

$$B'_{v_1,\dots,v_k} = \begin{cases} B_{v_1,\dots,v_k} - \sum_{s_{w_1}} \cdots \sum_{s_{w_{k-1}}} B_{w_1,\dots,w_{k-1},v_k} & \text{if } v_1 = v_2 = \cdots = v_k \\ B_{v_1,\dots,v_k} & \text{otherwise.} \end{cases} \tag{C6}$$

APPENDIX D: A WORKED EXAMPLE

Here, we provide an example of how our method works, by going step-by-step through the partitioning of a 12-node 3-uniform hypergraph. The hypergraph consists of three groups of nodes, namely nodes 1–4, 5–8, and 9–12, with all possible hyperedges of size 3 within each group. Additionally, as illustrated in Fig. 4(a), edges (3, 4, 7), (4, 5, 7), (6, 8, 11) and (8, 9, 11) join the cliques.

In the beginning, all the nodes belong to the same community, and the hypermodularity is 0. Computing the left-singular vector corresponding to the largest singular value results in a partition where nodes 1, 2, 3, 4, 6, and 7 are assigned to community 1, and the remaining ones are assigned to community 2. Since this partition, depicted in Fig. 4(b), has a hypermodularity of 0.375, it is accepted.

Next, a local node-level refinement is attempted. The best set of moves consists in shifting nodes 5 and 8 from community 2 to community 1. These moves yield an increase in hypermodularity of 0.144531. Thus, they are accepted, resulting in the partition shown in Fig. 4(c), with hypermodularity of 0.519531.

No global node-level refinement is possible, since it would be just a repetition of the last refinement step. Similarly, joining the two communities would make the hypermodularity vanish, and therefore no further modifications happen at this stage.

Trying to bisect community 1 using the higher-order SVD does not yield an increase in hypermodularity. Therefore, no bisection is performed on it. However, the best result from the local node-level refinement step consists in moving nodes 5–8 to a new community, which we label community 3. This move yields an increase of hypermodularity of 0.116699. Thus, it is accepted, and the resulting partition, shown in Fig. 4(d), has hypermodularity of 0.63623.

Next, a bisection is attempted on community 2, but this would not result in an increase of hypermodularity, and therefore none is performed. Similarly, local and global node-level refinements and community-joining attempts all would decrease the value of hypermodularity, and no such moves are accepted. At this point, community 2 is marked as blocked, since no further improvement can be obtained by working solely on it.

Similar attempts at continuing the process, first on community 1 and then on community 3, do not produce any improvement. Thus, the whole procedure stops. The final result is that the method correctly split the original network into the three densely-connected modules, producing a partition with a significant value of hypermodularity.

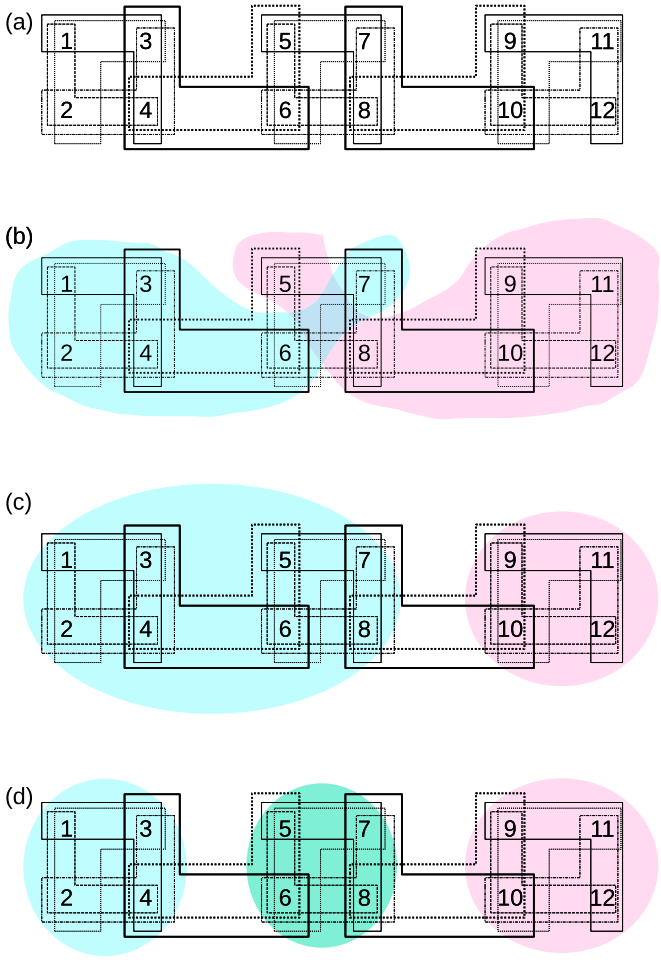


FIG. 4. Worked example of the hypermodularity maximization algorithm. (a) The network is a 12-node 3-uniform hypergraph consisting of three cliques, specifically on nodes 1–5, 5–8, and 9–12, joined by edges (3, 4, 7), (4, 5, 7), (6, 8, 11), and (8, 9, 11). All the edges within each clique are shown here as lines of different style. The edges that join the cliques are slightly thicker than the others. (b) After the first bisection, the best partition found consists of two communities, shown here in blue and pink, with a hypermodularity of 0.375. (c) After the first local node-level refinement, the communities change, and the hypermodularity rises to 0.519531. (d) Further passes of the method split the blue community into two, shown here in blue and emerald green. At this point, no further improvement is possible, and the algorithm stops. The hypermodularity of the final partition is 0.63623.

[1] R. Albert and A.-L. Barabási, Statistical mechanics of complex networks, Rev. Mod. Phys. **74**, 47 (2002).

[2] M. E. J. Newman, Structure and function of complex networks, SIAM Rev. **45**, 167 (2003).

[3] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, Complex networks: structure and dynamics, Phys. Rep. **424**, 175 (2006).

[4] S. Boccaletti, G. Bianconi, R. Criado, C. I. del Genio, J. Gómez-Gardeñes, M. Romance, I. Sendiña-Nadal, Z. Wang, and M. Zanin, The structure and dynamics of multilayer networks, Phys. Rep. **544**, 1 (2014).

[5] S. L. Pimm, Structure of food webs, Theor. Popul. Biol. **16**, 144 (1979).

[6] G. P. Garnett, J. P. Hughes, R. M. Anderson, B. P. Stoner, S. O. Aral, W. L. Whittington, H. H. Handsfield, and K. K. Holmes, Sexual mixing patterns of patients attending sexually transmitted diseases clinics, Sex. Transm. Dis. **23**, 248 (1996).

[7] G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee, Self-organization and identification of web communities, Computer **35**, 66 (2002).

[8] K. A. Eriksen, I. Simonsen, S. Maslov, and K. Sneppen, Modularity and extreme edges of the internet, Phys. Rev. Lett. **90**, 148701 (2003).

[9] A. E. Krause, K. A. Frank, D. M. Mason, R. E. Ulanowicz, and W. W. Taylor, Compartments revealed in food-web structure, Nature (London) **426**, 282 (2003).

[10] D. Lusseau and M. E. J. Newman, Identifying the role that animals play in their social networks, Proc. R. Soc. Lond. B Biol. **271**, S477 (2004).

[11] R. Guimerà and L. A. N. Amaral, Functional cartography of complex metabolic networks, Nature (London) **433**, 895 (2005).

[12] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature (London) **435**, 814 (2005).

[13] M. Huss and P. Holme, Currency and commodity metabolites: their identification and relation to the modularity of metabolic networks, IET Syst. Biol. **1**, 280 (2007).

[14] S. Fortunato, Community structure in graphs, Phys. Rep. **486**, 75 (2010).

[15] H. Cherifi, G. Palla, B. K. Szymanski, and X. Lu, On community structure in complex networks: Challenges and opportunities, Appl. Netw. Sci. **4**, 117 (2019).

[16] M. E. J. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. USA **103**, 8577 (2006).

[17] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner, On modularity clustering, IEEE T. Knowl. Data En. **20**, 172 (2008).

[18] M. Chen, K. Kuzmin, and B. K. Szymanski, Community detection via maximization of modularity and its variants, IEEE Trans. Comput. Soc. Syst. **1**, 46 (2004).

[19] J. Duch and A. Arenas, Community detection in complex networks using extremal optimization, Phys. Rev. E **72**, 027104 (2005).

[20] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. (2008) P10008.

[21] A. Noack and R. Rotta, Multi-level algorithms for modularity clustering, Lect. Notes Comput. Sci. **5526**, 257 (2009).

[22] Y. Sun, B. Danila, K. Josić, and K. E. Bassler, Improved community structure detection using a modified fine-tuning strategy, Europhys. Lett. **86**, 28004 (2009).

[23] B. H. Good, Y.-A. de Montjoye, and A. Clauset, Performance of modularity maximization in practical contexts, Phys. Rev. E **81**, 046106 (2010).

[24] E. Le Martelot and C. Hankin, Multi-scale community detection using stability as optimisation criterion in a greedy algorithm, Proc. Int. Conf. Knowledge Discovery and Information Retrieval (KDIR 2011), 216 (2011).

[25] S. Sobolevsky, R. Campari, A. Belyi, and C. Ratti, General optimization technique for high-quality community detection in complex networks, Phys. Rev. E **90**, 012811 (2014).

[26] S. Treviño III, A. Nyberg, C. I. del Genio, and K. E. Bassler, Fast and accurate determination of modularity and its effect size, J. Stat. Mech. (2015) P02003.

[27] X. Lu, B. Cross, and B. Szymanski, Asymptotic resolution bounds of generalized modularity and multi-scale community detection, Inform. Sci. **525**, 54 (2020).

[28] F. Battiston, G. Cencetti, I. Iacopini, V. Latora, M. Lucas, A. Patania, J.-G. Young, and G. Petri, Networks beyond pairwise interactions: structure and dynamics, Phys. Rep. **874**, 1 (2020).

[29] S. Boccaletti, P. De Lellis, C. I. del Genio, K. Alfaro-Bittner, R. Criado, S. Jalan, and M. Romance, The structure and dynamics of networks with higher order interactions, Phys. Rep. **1018**, 1 (2023).

[30] C. Berge, *Graphs and Hypergraphs* (Elsevier, New York, 1973).

[31] A. Eriksson, T. Carletti, R. Lambiotte, A. Rojas, and M. Rosvall, Flow-based community detection in hypergraphs, in *Higher-order Systems*, edited by F. Battiston and G. Petri (Springer, Cham, 2022).

[32] T. Kumar, S. Vaidyanathan, H. Ananthapadmanabhan, S. Parthasarathy, and B. Ravindran, Hypergraph clustering by iteratively reweighted modularity maximization, Appl. Net. Sci. **5**, 52 (2020).

[33] G. Contreras-Aso, R. Criado, G. Vera de Salas, and J. Yang, Detecting communities in higher-order networks by using their derivative graphs, Chaos Soliton. Fract. **177**, 114200 (2023).

[34] J. C. Wright Billings, M. Hu, G. Lerda, A. N. Medvedev, F. Mottes, A. Onicas, A. Santoro, and G. Petri, Simplex2Vec embeddings for community detection in simplicial complexes, arXiv:1906.09068.

[35] Y. Zhen and J. Wang, Community detection in general hypergraph via graph embedding, J. Am. Stat. Ass. **118**, 1620 (2023).

[36] M. C. Angelini, F. Caltagirone, F. Krzakala, and L. Zdeborová, Spectral detection on sparse hypergraphs, Proc. 2015 53rd Ann. Allerton Conf. Commun. Control Comp., 66 (2015).

[37] S. Rabanser, O. Shchur, and S. Günnemann, Introduction to tensor decompositions and their applications in machine learning, arXiv:1711.10781.

[38] Z. T. Ke, F. Shi, and D. Xia, Community detection for hypergraph networks via regularized tensor power iteration, arXiv:1909.06503.

[39] S. Krishnagopal and G. Bianconi, Spectral detection of simplicial communities via Hodge Laplacians, Phys. Rev. E **104**, 064303 (2021).

[40] B. Kamiński, V. Poulin, P. Prałat, P. Szufel, and F. Théberge, Clustering via hypergraph modularity, PLoS One **14**, e0224307 (2019).

[41] B. Kamiński, P. Misiorek, P. Prałat, and F. Théberge, Modularity based community detection in hypergraphs, J. Compl. Netw. **12**, cnae041 (2024).

[42] A. Rajwade, A. Rangarajan, and A. Banerjee, Image denoising using the higher order singular value decomposition, IEEE T. Pattern Anal. **35**, 849 (2013).

[43] P. Sankaranarayanan, T. E. Schomay, K. A. Aiello, and O. Alter, Tensor GSVD of patient- and platform-matched tumor and normal DNA copy-number profiles uncovers chromosome arm-wide patterns of tumor-exclusive platform-consistent alterations encoding for cell transformation and predicting ovarian cancer survival, PLoS One **10**, e0121396 (2015).

[44] Y.-H. Taguchi, Identification of candidate drugs using tensor-decomposition-based unsupervised feature extraction in integrated analysis of gene expression between diseases and DrugMatrix datasets, Sci. Rep. **7**, 13733 (2017).

[45] Y.-H. Taguchi, Tensor decomposition-based unsupervised feature extraction applied to matrix products for multi-view data processing, PLoS One **13**, e0183933 (2017).

[46] B. Vilas Boas, W. Zirwas and M. Haardt, Deep-LaRGE: higher-order SVD and deep learning for model order selection in MIMO OFDM systems, Proceedings of the 26th International ITG Workshop on Smart Antennas and 13th Conference on Systems, Communications, and Coding, Braunschweig, 1 (2023).

[47] F. Botta and C. I. del Genio, Finding network communities using modularity density, J. Stat. Mech. (2016) 123402.

[48] B. W. Kernighan and S. Lin, An efficient heuristic procedure for partitioning graphs, Bell Syst. Tech. J. **49**, 291 (1970).

[49] J. Stehlé, N. Voirin, A. Barrat, C. Cattuto, L. Isella, J.-F. Pinton, M. Quaggiotto, W. Van den Broeck, C. Régis, B. Lina, and P. Vanhems, High-resolution measurements of face-to-face contact patterns in a primary school, PLoS One **6**, e23176 (2011).

[50] P. S. Chodrow, N. Veldt, and A. R. Benson, Generative hypergraph clustering: From blockmodels to modularity, Sci. Adv. **7**, eabh1303 (2021).

[51] R. Mastrandrea, J. Fournet, and A. Barrat, Contact patterns in a high school: A comparison between data collected using wearable sensors, contact diaries and friendship surveys, PLoS One **10**, e0136497 (2015).

[52] C. R. Schwartz, C. Doren, and A. Li, Trends in years spent as mothers of young children the role of completed fertility, birth spacing, and multiple partner fertility, in *Analyzing Contemporary Fertility*, edited by R. Schoen (Springer, Cham, 2020).

[53] R. Guimerà, M. Sales-Pardo, and L. Amaral, Modularity from fluctuations in random graphs and complex networks, Phys. Rev. E **70**, 025101(R) (2004).

[54] M. Park, D. W. Feng, S. Digra, T.-A. Vu-Le, L. Anne, G. Chacko, and T. Warnow, Improved community detection using stochastic block models, arXiv:2502.00686.

[55] S. Fortunato and M. Barthélemy, Resolution limit in community detection, Proc. Natl. Acad. Sci. USA **104**, 36 (2007).

[56] https://codeberg.org/paraw/HyperMod.

[57] https://charodelgenio.weebly.com/community-detection-in-higher-order-networks.html.