

Representative community divisions of networks

Alec Kirkley¹ and M. E. J. Newman^{1,2}

¹*Department of Physics, University of Michigan, Ann Arbor, Michigan 48109, USA*

²*Center for the Study of Complex Systems, University of Michigan, Ann Arbor, Michigan 48109, USA*

Methods for detecting community structure in networks typically aim to identify a single best partition of network nodes into communities, often by optimizing some objective function. However, in real-world applications there are typically many competitive partitions with objective scores close to that of the global optimum and the true community structure is more properly represented by an entire set of high-scoring partitions than by just the single optimum. Such a set can be difficult to interpret since its size can easily run to hundreds or thousands of partitions. In this paper we present a solution to this problem in the form of an efficient method that clusters similar partitions into groups and then identifies an archetypal partition as a representative of each group. The result is a succinct, human-readable summary of the form and variety of community structure in any network. We demonstrate the method on a range of example networks.

I. INTRODUCTION

Networks are widely used as a compact quantitative representation of a range of complex systems, particularly in the biological and social sciences, engineering, computer science, and physics. Many networks naturally divide into communities, densely connected groups of nodes with sparser between-group connections [1]. Identifying these groups, in the process known as community detection, can help us in understanding network phenomena such as the evolution of social relationships [2], epidemic spreading [3], and others.

There are numerous existing methods for community detection, including ones based on centrality measures [4], modularity [5], information theory [6], and Bayesian generative models [7]—see [8] for a review. Most methods represent the community structure in a network as a single network partition or division (an assignment of each node to a specific community), which is typically the one that attains the highest score according to some objective function. As pointed out by many previous authors, however, there may be multiple partitions of a network that achieve high scores, any of which could be a good candidate for division of the network [9–14]. With this in mind some community detection methods, including methods based on modularity and on generative models, return multiple plausible partitions rather than just one. But while these algorithms give a more complete picture of community structure, they have their own problems. In particular, the number of partitions returned is often very large. Even for relatively small networks the partitions may number in the hundreds or thousands, far more than any human observer can reasonably comprehend. How then are we supposed to make sense of the output of these calculations?

In some cases it may happen that all of the plausible divisions of a network are quite similar to each other, in which case we may be able to form a *consensus clustering* [15], a single partition that is representative of the entire set in the same way that the mean of a set of numbers can be a useful representation of the whole.

However, if the partitions vary substantially, then some of them may be very different from the consensus clustering and hence the consensus will fail to capture the full range of behaviors in the same way that the mean can be a poor summary statistic for broad or multimodal distributions of numbers. In cases like these, summarizing the community structure may require not just one but several representative partitions, which may themselves be consensus partitions for a local cluster of network divisions [14]. In this paper, we present a simple and efficient method for finding such representative partitions. Given a large set of possible structures returned by a community detection algorithm, our method finds a smaller set that capture the main variants and possibilities while remaining comprehensible to human users.

In effect, our method clusters the partitions into a small number of representative subsets, in a manner somewhat akin to traditional methods for clustering numerical data in high-dimensional data spaces. A few previous studies have investigated the clustering of partitions. Calatayud et al. [16] proposed an algorithm that starts with the single highest scoring partition (under whatever objective function is in use), then iterates through other divisions in order of decreasing score and assigns each to the closest cluster if the distance to that cluster is less than a certain threshold, or starts a new cluster otherwise. This approach is fast but somewhat ad hoc and highly sensitive to the distance threshold in many cases. It also has the disadvantage of requiring us to choose a distance function between partitions [17] and the results turn out to be quite sensitive to this choice also. Peixoto [14] has proposed a principled statistical method that identifies representative partitions using methods of Bayesian inference. In this method, rather than assigning a single group label to each node in a representative partition, one defines a distribution over labels that gives the marginal probability that the node is in each community among all the partitions in a cluster. This method works well in many respects but is quite complicated to explain and to implement, and even for small networks it typically returns a large number of similar representative partitions, arguably too large for

simple interpretation.

The method we propose here is based on fundamental information theoretic principles and has the advantage that it does not require the explicit choice of any partition distance function and typically returns only a small number of representative partitions, each of which is a true network partition rather than a probability distribution. Our approach is based on the principle of *minimum description length*, which states that when selecting between possible models for a data set, the best model is the one that permits the most succinct representation of the data [18]. In our context, we seek to capture the information contained in a set of community divisions returned by some community detection algorithm using a model that consists of a few representative partitions that are used to reconstruct the clusters around them. The description length principle has been used in the past for clustering real-valued (non-network) data, including methods based on Gaussian mixture models [19], hierarchical clustering [20], Bernoulli mixture models for categorical data [21], and probabilistic generative models [22]. Georgieva et al. [23], for instance, have proposed a clustering framework that is similar in some respects to ours but for real-valued vector data. As in our approach the data are thought of as a message to be transmitted in multiple parts, including the cluster centers and the data within each cluster. Georgieva et al., however, only use their measure as a quality function to assess the outputs of other clustering algorithms and not as an objective to be optimized to obtain the clusters themselves.

Our algorithm takes as input a set of divisions of a network into communities, which may be obtained in any manner we like. Common methods for generating such divisions are sampling from probabilistic models, thermal samples generated using modularity or other energy functions, or multiple runs of optimization algorithms, and our method will work with any of these. We design a partition clustering objective function using simple information theoretic arguments, and use an efficient Monte Carlo scheme to optimize this objective and identify clusters of similar partitions and a representative member of each cluster. We test the method on a range of real and synthetic networks and demonstrate that it returns substantially distinct community divisions that are a good guide to the structures present in the original sample.

II. MATERIALS AND METHODS

The primary goal of our proposed technique is to find representative partitions that summarize the community structure in a network. We call these representative partitions *modes*. Suppose we have an observed network consisting of N nodes and we have some method for finding community divisions of these nodes, also called partitions. We can represent a partition with a length- N vector \mathbf{g} that assigns to each node $i = 1 \dots N$ a label g_i indicating which community it belongs to.

We assume that there are a large number of plausible partitions and that our community detection method returns a subset of them. Normally we expect that many of the partitions would be similar to one another, differing only by a few nodes here or there. The goal of this paper is to develop a procedure for gathering such similar partitions into clusters, and generating a mode, which is itself a partition, as an archetypal representative of each cluster. For the sake of clarity, we will in this paper use the words “partition” or “division” to describe the assignment of network nodes to communities, and the word “cluster” to describe the assignment of entire partitions to groups according to the method that we describe.

In order both to divide the partitions into clusters and to find a representative mode for each cluster, we first develop a clustering objective function based on information theoretic arguments. The main concept behind our approach is a thought experiment in which we imagine transmitting our set of partitions to a receiver using a multi-step encoding chosen so as to minimize the amount of information required for the complete transmission.

A. Partition clustering as an encoding problem

Let us denote our set of partitions by D and suppose there are S partitions in the set, labeled $p = 1 \dots S$. Now imagine we wish to transmit a complete description of all elements of the set to a friend. How should we go about this? The most obvious way is to send each of the partitions separately to the receiver using some simple encoding that uses, say, numbers or symbols to represent community labels. We could do somewhat better by using an optimal prefix code such as a Huffman code [24] that economizes by representing frequently used labels with shorter code words. Even this, however, would be quite inefficient in terms of information. We can do better by making use of the fact that, as we have said, we expect many of our partitions to be similar to one another. This allows us to save information by dividing the partitions into clusters of similar ones and transmitting only a few partitions in full—one representative partition or mode for each cluster—then describing the remaining partitions by how they differ from these modes. The method is illustrated in Fig. 1.

Initially, let us assume that we want to divide the set D of partitions into K clusters, denoted C_k with $k = 1 \dots K$. (We will discuss how to choose K separately in a moment.) To efficiently transmit D , we first transmit K representative modes, which themselves are members of D , with group labels $\hat{\mathbf{g}}^{(k)}$. Then for each individual partition in D we transmit which cluster, or equivalently which mode, it belongs to and then the partition itself by describing how it differs from that mode. Since the latter information will be smaller if a partition is more similar to its assigned mode, choosing a set of modes that are accurately representative of all partitions

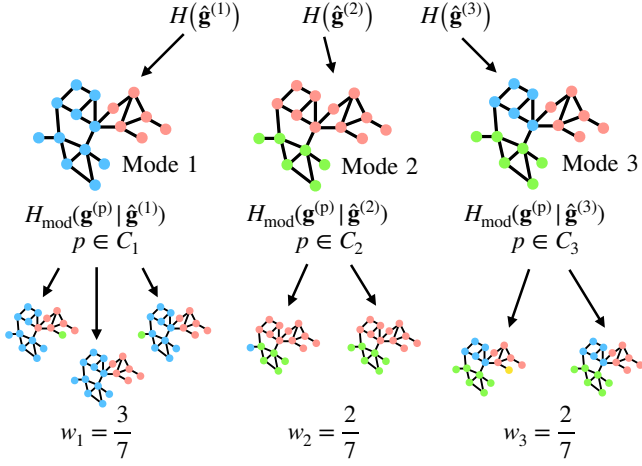


FIG. 1: Illustration of the transmission of a set of partitions for a network. We first transmit a small set of “modes,” archetypal partitions drawn from the larger set. Then each partition from the complete set is transmitted by describing how it differs from the most similar of the modes.

will naturally minimize the total information, and we use this criterion to derive the best set of modes. This is the minimum description length principle, as applied to finding the optimal clusters and modes.

Following this plan, the total description length per sampled partition can be written in the form

$$\mathcal{L}_{\text{total}} = \frac{N}{S} \sum_{k=1}^K H(\hat{\mathbf{g}}^{(k)}) + H(\mathbf{c}) + \frac{N}{S} \sum_{k=1}^K \sum_{p \in C_k} H_{\text{mod}}(\mathbf{g}^{(p)} | \hat{\mathbf{g}}^{(k)}). \quad (1)$$

The first term represents the amount of information required to transmit the modes and is simply equal to the sum of their entropies:

$$H(\hat{\mathbf{g}}^{(k)}) = - \sum_{r=1}^{n_{m_k}} \frac{a_r^{(m_k)}}{N} \log \frac{a_r^{(m_k)}}{N}. \quad (2)$$

Here m_k is the partition label p of the k th mode, n_p is the number of communities in partition p , and $a_r^{(p)}$ is the number of nodes in partition p that have community label r .

The second term in Eq. 1 represents the amount of information needed to specify which cluster, or alternatively which mode, each partition in D belongs to:

$$H(\mathbf{c}) = - \sum_{k=1}^K \frac{c_k}{S} \log \frac{c_k}{S}, \quad (3)$$

where $c_k = |C_k|$ is the number of partitions (out of S total) that belong to mode k .

The third term in 1 represents the amount of information needed to specify each of the individual partitions $\mathbf{g}^{(p)}$ in terms of their modes $\hat{\mathbf{g}}^{(k)}$:

$$H_{\text{mod}}(\mathbf{g}^{(p)} | \hat{\mathbf{g}}^{(k)}) = H(\mathbf{g}^{(p)} | \hat{\mathbf{g}}^{(k)}) + \frac{1}{N} \log \Omega(p, m_k). \quad (4)$$

H_{mod} is the *modified conditional entropy* of the group labels of $\mathbf{g}^{(p)}$ given the group labels of $\hat{\mathbf{g}}^{(k)}$ [25]. The normal (non-modified) conditional entropy is

$$H(\mathbf{g}^{(p)} | \hat{\mathbf{g}}^{(k)}) = - \sum_{r=1}^{n_{m_k}} \sum_{s=1}^{n_p} \frac{t_{rs}^{m_k p}}{N} \log \frac{t_{rs}^{m_k p}}{a_r^{(m_k)}}, \quad (5)$$

where $t_{rs}^{m_k p}$ is the number of nodes simultaneously classified into community r in partition $\mathbf{g}^{(m)}$ and community s in partition $\mathbf{g}^{(p)}$. The matrix of elements t^{m_p} for any pair of partitions m, p is known as a *contingency table*, and Eq. 5 measures the amount of information needed to transmit $\mathbf{g}^{(p)}$ given that we already know both $\hat{\mathbf{g}}^{(k)}$ and the contingency table. To actually transmit the partitions in practice we would also need to transmit the contingency table, and the second term in Eq. 4 represents the information needed to do this. The quantity $\Omega(p, m)$ is equal to the number of possible contingency tables t^{m_p} with row and column sums $a_r^{(m)}$ and $a_s^{(p)}$ respectively. This quantity can be computed exactly for smaller contingency tables and there exist good approximations to its value for larger tables [25].

The modified conditional entropy, including the $\log \Omega$ term, thus measures the total amount of information needed to transmit the partition $\mathbf{g}^{(p)}$ after having already transmitted its mode $\hat{\mathbf{g}}^{(k)}$. The $\log \Omega$ term is often omitted from calculations of conditional entropy, but it turns out to be crucial in the current application. Without it, one can minimize the conditional entropy simply by making the number of groups in the modal partition very large, with the result that the minimum description length solution is biased toward modes with many groups. The additional term avoids this bias.

A detailed derivation of Eq. 1 is given in Appendix A. By minimizing this quantity we can now find the best set of modes to describe a given set of partitions.

B. Choosing the number of clusters

So far we have assumed that we know the number K of clusters of partitions, or equivalently the number of modes. In practice we do not usually know K , and moreover there is not normally even one “correct” value for a given network. Different values of K can give useful answers for the same network, depending on how much granularity we wish to see in the community structures returned by the method. How then are we to choose the value of K ?

One approach is to use the description length itself to choose K . Low values of K , corresponding to only a

small number of modes, will give inefficient descriptions of the data because many partitions will not be similar to any of the modes. Conversely, high values of K will give inefficient partitions because we will waste a lot of information describing all the modes. In between, at some moderate value of K , lies the maximally efficient choice. Thus, we might imagine we could simply look for the minimum description length among all values of K to find the best value. An analogous method is used, for example, for choosing the optimal number of bins for histograms and often works well in that context [26, 27].

For our problem, however, this approach does not usually give a useful answer because the number of modes it returns depends on the number of partitions S in the sample, increasing as the value of S increases. In the case of histograms this is desirable—you want to use more bins when you have more data—but in the case of community structure it usually is not. Normally we would like our representation of the space of community structures to capture the fundamental features of the network independent on how we choose to sample those features, including how many samples we draw. Moreover, we find that the number of modes becomes unmanageably large for sample sizes S greater than a few thousand and the individual modes themselves differ only very slightly in their composition.

We would prefer a number of modes that remains constant as S becomes large. A natural way to achieve this is to impose a penalty on the description length objective function using a multiplier or “chemical potential” that couples linearly to the value of K thus:

$$\begin{aligned} \mathcal{L}_{\text{total}} = & \frac{N}{S} \sum_{k=1}^K H(\hat{\mathbf{g}}^{(k)}) + H(\mathbf{c}) \\ & + \frac{N}{S} \sum_{k=1}^K \sum_{p \in C_k} H_{\text{mod}}(\mathbf{g}^{(p)} | \hat{\mathbf{g}}^{(k)}) + \lambda K. \end{aligned} \quad (6)$$

This is the objective function we use in our calculations. It is straightforward to show that (unlike the description length by itself) this form makes the optimal number of clusters K independent of S —see Appendix B for a derivation, where we also show that the same guarantee cannot be made for the unpenalized description length.

Equation 6 contains the free parameter λ , whose value affects the number of modes, but if we want that number to be small (as we normally do) then we expect λ to be of order unity. In practice, we find that the choice $\lambda = 1$ works well and this is the value we use for all the example applications presented here.

C. Minimizing the description length

Our goal is now to find the set of modes $\hat{\mathbf{g}}$ that minimize Eq. 6. This could be done using any of a variety of optimization methods, but here we make use

of a greedy algorithm that employs a sequence of elementary moves that merge and split clusters, inspired by a similar merge-split algorithm for sampling community structures described in [28]. We start by randomly dividing our set D of partitions into some number K_0 of initial clusters, then identify the mode $\hat{\mathbf{g}}^{(k)}$ of each cluster C_k as the partition $p \in C_k$ that minimizes $H(\mathbf{g}^{(p)}) + \sum_{q \in C_k} H_{\text{mod}}(\mathbf{g}^{(q)} | \mathbf{g}^{(p)})$. In other words, the initial mode for each cluster is the partition p that is closest to all others in terms of modified conditional entropy, accounting for the entropy of p itself.

Computing the modified conditional entropy, Eq. 4, has time complexity $O(N)$, which means it takes $O(NS^2/K_0^2)$ steps to compute each mode exactly if the initial clusters are the same size. This can be slow in practice, but we can obtain a good approximation substantially faster by Monte Carlo sampling. We draw a random sample X of partitions from the cluster (without replacement) and then minimize $H(\mathbf{g}^{(p)}) + (c_k/|X|) \sum_{q \in X} H_{\text{mod}}(\mathbf{g}^{(q)} | \mathbf{g}^{(p)})$, where as previously c_k is the size of the cluster. Good results can be obtained with relatively small samples. In our calculations we use $|X| = 30$. The time complexity of this calculation is $O(NS/K_0)$, a significant improvement given that sample sizes S can run into the thousands or more. We also store the values of $H(\mathbf{g}^{(p)})$ and $H_{\text{mod}}(\mathbf{g}^{(q)} | \mathbf{g}^{(p)})$ as they are computed so that they do not need to be recomputed on subsequent steps of the algorithm.

Now that we have an initial set of clusters and representative modes, the algorithm proceeds by repeatedly proposing one of the following moves at random, accepting it only if it reduces the total description length in Eq. 6:

1. Pick a partition $\mathbf{g}^{(p)}$ at random and assign it to the closest mode $\hat{\mathbf{g}}^{(k)}$.
2. Pick two clusters $C_{k'}$ and $C_{k''}$ at random and merge them into a single cluster C_k , recomputing the cluster mode as before.
3. Pick a cluster C_k at random and split it into two clusters $C_{k'}$ and $C_{k''}$ using a k -means style algorithm: we select two modes at random from C_k and assign each partition in C_k to the closer of the two (in terms of modified conditional entropy). Then we recompute the modes for each resulting cluster and repeat until convergence is reached.

These steps together constitute a complete algorithm for minimizing the description length and optimizing the clusters, but we find that the efficiency of the algorithm can be further improved by adding a fourth move thus:

4. Perform step 2, then immediately perform step 3 using the merged cluster from step 2.

This extra move, inspired by a similar one in the community merge-split algorithm of [28], helps with the rapid optimization of partition assignments between pairs of clusters.

We continue performing these moves until a prescribed number of consecutive moves are rejected without improving the description length, then the algorithm halts. We find that this procedure returns very consistent results despite its random nature. It has $O(NS)$ time complexity per move in the worst case (which occurs when there is just a single cluster), and is fast in practice. In particular, it is typically much faster than the community detection procedure itself for current community detection algorithms, so it adds little to the overall time needed to analyze a network. We give a range of example applications in the next section.

III. RESULTS

In this section we demonstrate the application of our method to a number of example networks, both real and computer generated. For each example we perform community detection by fitting to the non-parametric degree-corrected block model [29] and sampling 10 000 community partitions from the posterior distribution of the model by Markov chain Monte Carlo using the algorithm of [28]. These samples are then clustered using the method of this paper with the cluster penalty parameter set to $\lambda = 1$, the number of Monte Carlo samples for estimating modes to $|X| = 30$, and the number of initial modes to $K_0 = 1$. We also calculate for each mode k a weight $w_k = c_k/S$ equal to the fraction of all partitions in D that fall in cluster k , to assess the relative sizes of the clusters.

A. Synthetic networks

As a first test of our method, we apply it to a set of synthetic (i.e., computer-generated) networks specifically constructed to display varying degrees of ambiguity in their community structure. Figure 2A shows results for a network generated using the planted partition model, a symmetric version of the stochastic block model [30, 31] in which N nodes are assigned in equal numbers to q communities, and between each pair of nodes i, j an edge is placed with probability p_{in} if i and j are in the same community or p_{out} if i and j are in different communities. In our example we generated a network with $N = 100$ nodes, $q = 4$ communities, and $p_{\text{in}} = 0.25$, $p_{\text{out}} = 0.02$. Though it contains four communities, by its definition, this network should exhibit only a single mode, the structure “planted” into it in the network generation process. There will be competing individual partitions, but they should be distributed evenly around the single modal structure rather than multimodally around two or more structures. And indeed our algorithm correctly infers this as shown in the figure: it returns a single representative structure in which all nodes are grouped correctly into their planted communities. Given the random nature of the community detection algorithm it would be

possible for a small number of nodes to be incorrectly assigned in the modal structure, simply by chance, but in the present case this did not happen and every node is assigned correctly.

For a second, more demanding example we construct a network using the full (non-symmetric) stochastic block model, which is more flexible than the planted partition model. If \mathbf{g} denotes a vector of community assignments as previously, then an edge in the model is placed between each node pair i, j independently at random with probability ω_{g_i, g_j} , where the ω_{g_i, g_j} are parameters that we choose. For our example we create a network with three communities and with parameters of the form

$$\omega = \begin{bmatrix} p_s & p_m & p_b \\ p_m & p_s & p_b \\ p_b & p_b & p_s \end{bmatrix}, \quad (7)$$

where p_s is the within-group edge probability, p_m and p_b are between-group probabilities, and $p_s > p_m > p_b$. In our particular example the network has $N = 99$ nodes divided evenly between the three groups and $p_s = 0.27$, $p_m = 0.08$, $p_b = 0.01$. This gives the network a nested structure in which there is a clear separation between group 3 and the rest, and a weaker separation between groups 1 and 2. This sets up a deliberate ambiguity in the community structure: does the “correct” structure have three groups or just two? As shown in Fig. 2B, our method accurately pinpoints this ambiguity, finding two representative modes for the network, one with three separate communities and one where communities 1 and 2 are merged together.

A third synthetic example network is shown in Fig. 2C, the “ring of cliques” network proposed by Good et al. [12], in which a set of cliques (i.e., complete sub-graphs) are joined together by single edges to create a loop. In their studies, Good et al. found this network to have ambiguous community structure in which the cliques joined together in pairs rather than forming separate communities on their own. Since there are two symmetry-equivalent ways to divide the ring into clique pairs this also means there are two equally good divisions of the network into communities. Good et al. performed their community detection using modularity maximization, but similar behavior is seen with the method used here. Most sampled community structures show the same division into pairs of cliques, except for a node or two that may get randomly assigned to a different community. Our algorithm readily picks out this structure as shown in Fig. 2C, finding two modes that correspond to the two rotationally equivalent configurations. Moreover, the two modes have approximately equal weight w_k in the sampling, indicating that the Monte Carlo algorithm spent a roughly equal amount of time on partitions near each mode.

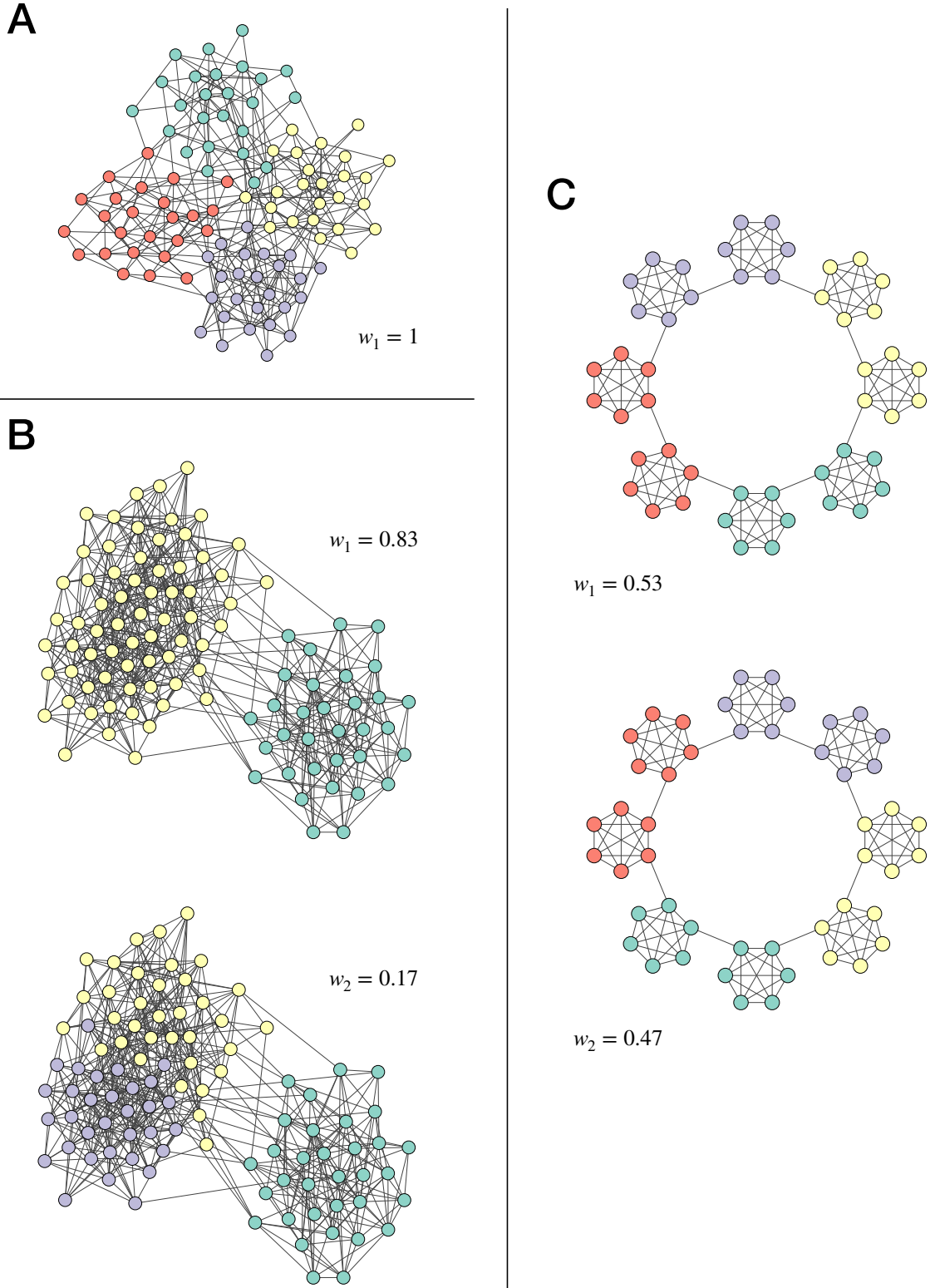


FIG. 2: Representative modes and their corresponding weights for three synthetic example network, identified by minimizing Eq. 6 with $\lambda = 1$ for 10,000 community partition samples. (A) Planted partition model with 100 nodes, four communities, and connection probabilities $p_{\text{in}} = 0.25$ and $p_{\text{out}} = 0.02$. (B) Network of 100 nodes generated using the stochastic block model with a mixing matrix of the form given in Eq. 7 with $p_s = 0.27$, $p_m = 0.08$, and $p_b = 0.01$. (C) Ring of eight cliques of size six nodes each, connected by single edges, based on the example in [12].

B. Real networks

Turning now to real-world networks, we show that our method can also accurately summarize community structure found in a range of practical domains. (Further examples are given in Appendix C.) The results demonstrate not only that the method works but also that real-world networks commonly do have multimodal community structure that is best summarized by two or more modes rather than by just a single consensus partition, although our method will return a single partition when it is justified—see Section III A.

Figure 3A shows results for one well-studied network, the co-purchasing network of books about politics compiled by Krebs (unpublished, but see [32]), where two books are connected by an edge if they were frequently purchased by the same buyers. It has been conjectured that this network contains two primary communities, corresponding to politically left- and right-leaning books, but the network contains more subtle divisions as well. A study by Peixoto [14] found 11 different types of structure—what we are here calling “modes.” Many of these modes, however, differed only slightly, by the reassignment of a few nodes from one community to another. Applying our method to the network we find, by contrast, just two modes as shown in the figure, suggesting that our algorithm is penalizing minor variations in structure more heavily than that of Ref. [14]. The two modes we find have four communities each. In the one on the left in Fig. 3A these appear to correspond approximately to books that are politically liberal (red), center-left (purple), center-right (green), and conservative (yellow); in the one on the right they are left-liberal (green), liberal (red), center (purple), and conservative (yellow).

Figure 3B shows a different kind of example, a social network of self-reported friendships among US high school students drawn from the National Longitudinal Study of Adolescent to Adult Health (the “Add Health” study) [33, 34]. The particular network we examine here is network number 5 from the study with 157 students. (Two nodes with degree zero were removed from the network before running the analysis.) As the figure shows, the method in this case finds three modes, each composed of half a dozen core communities of highly connected nodes whose boundaries shift somewhat from one mode to another, as well as a set of centrally located nodes (pale pink and yellow in the figure) that seem to move between communities in different modes. The movement of nodes from one community to another may be a sign of different roles played by core and peripheral members of social circles, or of students with a broad range of friendships.

In Fig. 3C, we show a third type of network, a geographic network of census tracts in the city of Chicago (USA). In this network the nodes represent the census tracts and two nodes are joined by an edge if the two corresponding tracts share a border [35]. Community detection applied to this network tends to find contiguous

local neighborhoods. Our algorithm finds three modes that differ primarily in the communities on the south-west side of the city where the density of census tracts is lower (though it is unclear whether this is the driving factor in the variation of community structure).

IV. DISCUSSION

In this paper we have presented a method for summarizing the complex output of community detection algorithms by generating a small number of archetypal network partitions that are broadly representative of high-scoring partitions in general. The method is based on fundamental information theoretic principles, employing a clustering objective function equal to the description length required to transmit a set of partitions using a specific multi-step encoding that we describe. We have developed an efficient algorithm to minimize this objective and we give examples of applications to both synthetic and real-world networks that exhibit nontrivial multimodal community structure.

One can envisage many potential applications of this approach. As mentioned in Section III B, the representative community partitions for a social network could highlight distinct roles or reveal information about the diversity of a node’s social circle. In networks for which we have additional node metadata we could investigate how individual attributes are associated with the representative partitions. Multimodal community structure may also be of interest in spatial networks, for instance for assessing competing partitions, as in mesh segmentation in engineering and computer graphics [36]. More generally, in the same way that any measurement can be supplemented with an error estimate, any community structure analysis could be supplemented with an analysis of competing partitions to help understand whether the optimal division is representative of the structure of the network as a whole.

The techniques presented in this paper could be extended in a number of ways. Our framework is applicable to any set of partitions—not just community divisions of a network but partitions of any set of objects or data items—so it could be applied in any situation where there are multiple competing ways to cluster objects. All that is needed is an appropriate measure of the information required to encode representative objects and their corresponding clusters. One potential application within network science could be to the identification of representative networks within a set sampled from some generative model, such as an exponential random graph model [37]. These extensions, however, we leave for future work.

Acknowledgments: This work was funded in part by the US Department of Defense NDSEG fellowship program (AK) and by the National Science Foundation under grant number DMS-2005899 (MEJN).

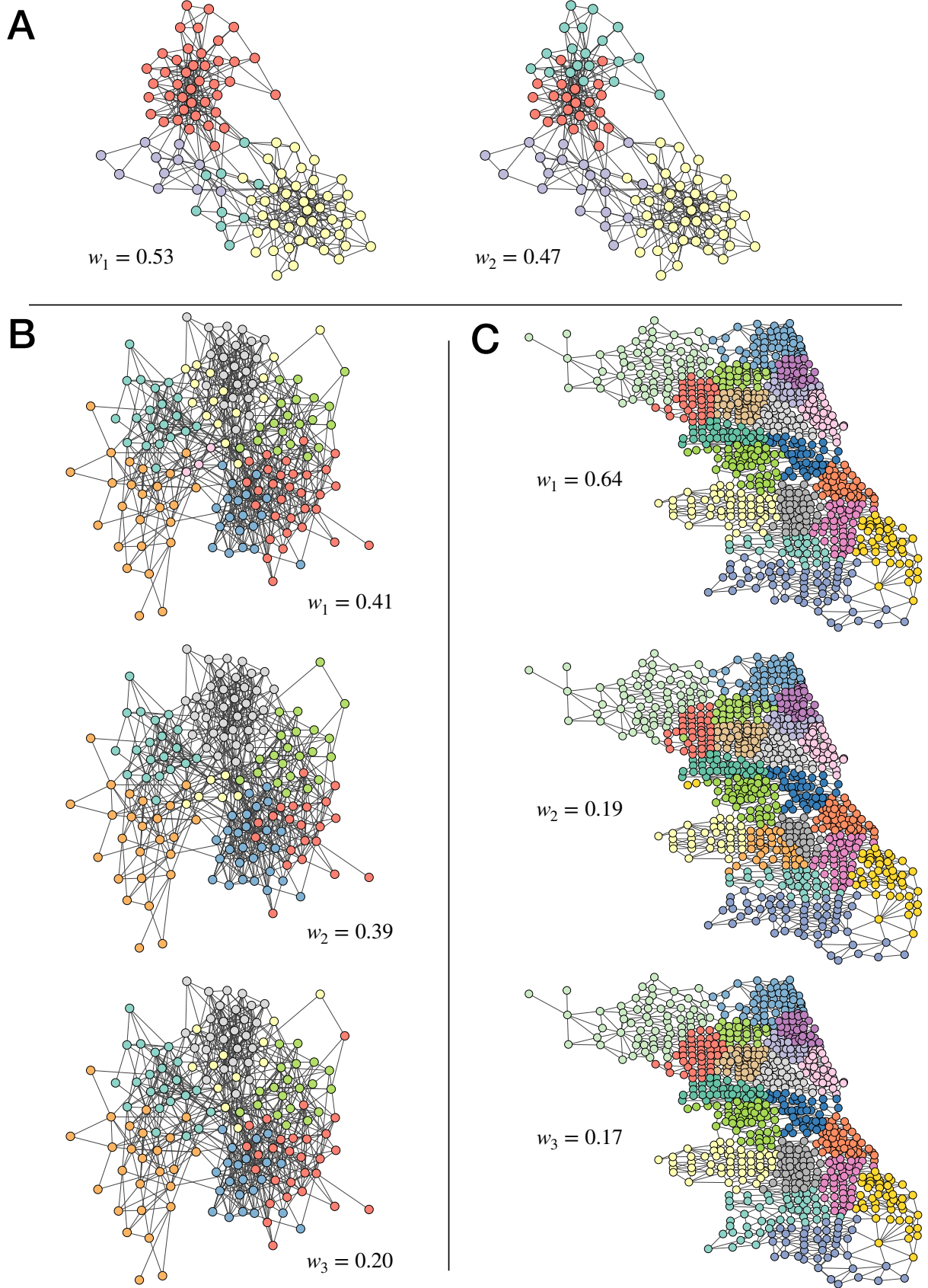


FIG. 3: Representative modes and their corresponding weights for three real-world example network, identified by minimizing Eq. 6 with $\lambda = 1$ for 10,000 community partition samples. (A) Network of political book co-purchases [32]. (B) High school friendship network [33, 34]. (C) Network of adjacent census tracts in the city of Chicago [35].

-
- [1] M. Newman, *Networks*. Oxford University Press, Oxford, 2nd edition (2018).
- [2] P. Bedi and C. Sharma, Community detection in social networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **6**, 115–135 (2016).
- [3] W. Huang and C. Li, Epidemic spreading in scale-free networks with community structure. *Journal of Statistical Mechanics* **2007**, P01014 (2007).
- [4] M. Girvan and M. E. J. Newman, Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **99**, 7821–7826 (2002).
- [5] M. E. J. Newman, Fast algorithm for detecting community structure in networks. *Phys. Rev. E* **69**, 066133 (2004).
- [6] M. Rosvall and C. T. Bergstrom, Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci. USA* **105**, 1118–1123 (2008).
- [7] T. P. Peixoto, Bayesian stochastic blockmodeling. In *Advances in Network Clustering and Blockmodeling*, P. Doreian, V. Batagelj, A. Ferligoj (editors), pp. 289–332, Wiley, New York (2019).
- [8] S. Fortunato, Community detection in graphs. *Phys. Rep.* **486**, 75–174 (2010).
- [9] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral, Modularity from fluctuations in random graphs and complex networks. *Phys. Rev. E* **70**, 025101 (2004).
- [10] C. P. Massen and J. P. K. Doye, Identifying “communities” within energy landscapes. *Phys. Rev. E* **71**, 046101 (2005).
- [11] J. Reichardt and S. Bornholdt, Statistical mechanics of community detection. *Phys. Rev. E* **74**, 016110 (2006).
- [12] B. H. Good, Y.-A. de Montjoye, and A. Clauset, Performance of modularity maximization in practical contexts. *Phys. Rev. E* **81**, 046106 (2010).
- [13] M. A. Riolo and M. E. J. Newman, Consistency of community structure in complex networks. *Phys. Rev. E* **101**, 052306 (2020).
- [14] T. P. Peixoto, Revealing consensus and dissensus between network partitions. *Phys. Rev. X* **11**, 021003 (2021).
- [15] A. Lancichinetti and S. Fortunato, Consensus clustering in complex networks. *Sci. Rep.* **2**, 1–7 (2012).
- [16] J. Calatayud, R. Bernardo-Madrid, M. Neuman, A. Rojas, and M. Rosvall, Exploring the solution landscape enables more reliable network community detection. *Phys. Rev. E* **100**, 052308 (2019).
- [17] N. X. Vinh, J. Epps, and J. Bailey, Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research* **11**, 2837–2854 (2010).
- [18] P. D. Grünwald and A. Grünwald, *The Minimum Description Length Principle*. MIT Press, Cambridge, MA (2007).
- [19] J. Tabor and P. Spurek, Cross-entropy clustering. *Pattern Recognition* **47**, 3046–3059 (2014).
- [20] R. S. Wallace and T. Kanade, Finding natural clusters having minimum description length. In *10th International Conference on Pattern Recognition*, pp. 438–442, IEEE Press, Hoboken (1990).
- [21] T. Li, S. Ma, and M. Ogihara, Entropy-based criterion in categorical clustering. In *Proceedings of the Twenty-first International Conference on Machine Learning*, p. 68, Association for Computing Machinery, New York (2004).
- [22] M. Narasimhan, N. Jojic, and J. A. Bilmes, Q-clustering. *Advances in Neural Information Processing Systems* **18**, 979–986 (2005).
- [23] O. Georgieva, K. Tschumitschew, and F. Klawonn, Cluster validity measures based on the minimum description length principle. In *Proceedings of the International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pp. 82–89, Springer-Verlag, Berlin (2011).
- [24] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley, New York (1991).
- [25] M. E. J. Newman, G. T. Cantwell, and J.-G. Young, Improved mutual information measure for clustering, classification, and community detection. *Phys. Rev. E* **101**, 042304 (2020).
- [26] D. P. Doane, Aesthetic frequency classifications. *The American Statistician* **30**, 181–183 (1976).
- [27] P. Hall, Akaike’s information criterion and Kullback-Leibler loss for histogram density estimation. *Probability Theory and Related Fields* **85**, 449–467 (1990).
- [28] T. P. Peixoto, Merge-split Markov chain Monte Carlo for community detection. *Phys. Rev. E* **102**, 012305 (2020).
- [29] T. P. Peixoto, Nonparametric Bayesian inference of the microcanonical stochastic block model. *Phys. Rev. E* **95**, 012317 (2017).
- [30] P. W. Holland, K. B. Laskey, and S. Leinhardt, Stochastic blockmodels: First steps. *Social Networks* **5**, 109–137 (1983).
- [31] B. Karrer and M. E. J. Newman, Stochastic blockmodels and community structure in networks. *Phys. Rev. E* **83**, 016107 (2011).
- [32] M. E. J. Newman, Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA* **103**, 8577–8582 (2006).
- [33] P. S. Bearman, J. Moody, and K. Stovel, Chains of affection: The structure of adolescent romantic and sexual networks. *Am. J. Sociol.* **110**, 44–91 (2004).
- [34] J. R. Udry, P. S. Bearman, and K. M. Harris, National Longitudinal Study of Adolescent Health (1997). This research uses data from Add Health, a program project directed by Kathleen Mullan Harris and designed by J. Richard Udry, Peter S. Bearman, and Kathleen Mullan Harris at the University of North Carolina at Chapel Hill, and funded by grant P01-HD31921 from the Eunice Kennedy Shriver National Institute of Child Health and Human Development, with cooperative funding from 23 other federal agencies and foundations. Special acknowledgment is due Ronald R. Rindfuss and Barbara Entwisle for assistance in the original design. Information on how to obtain the Add Health data files is available on the Add Health website (<http://www.cpc.unc.edu/addhealth>). No direct support was received from grant P01-HD31921 for this analysis.
- [35] A. Kirkley, Information theoretic network approach to socioeconomic correlations. *Phys. Rev. Research* **2**, 043212 (2020).
- [36] A. Shamir, A survey on mesh segmentation techniques. In *Computer Graphics Forum*, volume 27, pp. 1539–1556, The Eurographics Association and John Wiley & Sons (2008).

[37] D. Lusher, J. Koskinen, and G. Robins, *Exponential Random Graph Models for Social Networks: Theory, Methods, and Applications*. Cambridge University Press,

Cambridge (2012).

Appendix A: Derivation of the description length

In this appendix we derive the description length, Eq. 1, used in our calculations. The description length is equal to the amount of information needed to transmit the complete set of sampled partitions. We break up the transmission procedure into four separate steps:

1. We transmit S vectors $\mathbf{a}^{(p)}$, one for each $p = 1 \dots S$. If partition p has n_p non-empty communities, then there are $\binom{N-1}{n_p-1}$ ways to choose the values in the vector $\mathbf{a}^{(p)}$ and hence $\binom{N-1}{n_p-1}$ possible messages that may need to be transmitted to the receiver to communicate $\mathbf{a}^{(p)}$. In binary, our encoding thus requires $\log \binom{N-1}{n_p-1}$ bits, where \log denotes the logarithm base 2. (Strictly the number of bits is equal to the smallest integer that is greater than or equal to this number, but the difference is negligible for large N .) The information required for transmitting all count vectors $\mathbf{a}^{(p)}$ is then

$$L_1 = \sum_{p=1}^S \log \binom{N-1}{n_p-1}. \quad (\text{A1})$$

This quantity does not depend on the choice of modes or on the partitions themselves, so we can ignore it when we optimize the total description length of our encoding. It is conceptually important, however, that the $\mathbf{a}^{(p)}$ are transmitted first, as they are needed for constructing efficient encodings for other quantities.

2. Next we transmit the full set of group labels $\hat{\mathbf{g}}^{(k)}$ for each of the mode partitions, exploiting the fact that we now know the label count vector $\mathbf{a}^{(m_k)}$ for each mode. The number of possible sets of group labels consistent with this vector is given by $N! / \prod_{r=1}^{n_{m_k}} a_r^{(m_k)}!$ and hence the number of bits required to transmit a particular set is

$$L_2 = \sum_{k=1}^K \log \left(\frac{N!}{\prod_{r=1}^{n_{m_k}} a_r^{(m_k)}!} \right). \quad (\text{A2})$$

3. For each partition p , we transmit the partition number m_k of the mode to which it belongs. This effectively specifies the clusters themselves. This can be done efficiently by first transmitting the size $c_k = |C_k|$ of each of the K clusters. There are $\binom{S-1}{K-1}$ possible choices such that $\sum_{k=1}^K c_k = S$, so it takes $\log \binom{S-1}{K-1}$ bits to transmit any one choice. Then, given the c_k there are $S! / \prod_{k=1}^K c_k!$ possible ways to assign the partitions to the clusters, so the total number of bits

required to transmit the cluster labels for all partitions is

$$L_3 = \log \binom{S-1}{K-1} + \log \left(\frac{S!}{\prod_{k=1}^K c_k!} \right). \quad (\text{A3})$$

4. Finally, we transmit the groups labels $\mathbf{g}^{(p)}$ for each individual partition other than the modes, making use of the fact that the modes have already been transmitted. We do this in two steps:

- (a) We first transmit the contingency table $\mathbf{t}^{m_k p}$. Since the receiver knows $\mathbf{a}^{(m_k)}$ and $\mathbf{a}^{(p)}$, they also know the row and column sums of $\mathbf{t}^{m_k p}$ because

$$\sum_r t_{rs}^{m_k p} = a_s^{(p)} \quad (\text{A4})$$

and

$$\sum_s t_{rs}^{m_k p} = a_r^{(m_k)}. \quad (\text{A5})$$

If there are $\Omega(m_k, p)$ possible contingency tables with these row and column sums, then it takes $\log \Omega(m_k, p)$ bits to transmit the contingency table $\mathbf{t}^{m_k p}$. Closed-form expressions for $\Omega(m_k, p)$ exist for smaller tables. For larger ones there are good approximations, as described in Ref. [25].

- (b) Given the contingency table, the number of partitions consistent with the table is $\prod_{r=1}^{n_{m_k}} [a_r^{(m_k)}! / \prod_{s=1}^{n_p} t_{rs}^{m_k p}!]$ and the number of bits needed to transmit one partition is the log of this number.

The total number of bits required for transmitting the non-mode partitions is thus

$$L_4 = \sum_{k=1}^K \sum_{\substack{p \in C_k \\ p \neq m_k}} \log \prod_{r=1}^{n_{m_k}} \frac{a_r^{(m_k)}!}{\prod_{s=1}^{n_p} t_{rs}^{m_k p}!} + \sum_{k=1}^K \sum_{\substack{p \in C_k \\ p \neq m_k}} \log \Omega(m_k, p). \quad (\text{A6})$$

In practice, the exclusion of the term $p = m_k$ from the sums makes little difference and can be neglected without significantly changing the results.

Combining everything, the total description length for the model is

$$L_{\text{total}} = L_1 + L_2 + L_3 + L_4. \quad (\text{A7})$$

For our purposes it is convenient to normalize this as description length per sample, which gives

$$\mathcal{L}_{\text{total}} = \frac{1}{S}(L_1 + L_2 + L_3 + L_4). \quad (\text{A8})$$

We can convert this quantity to more familiar language by using Stirling's approximation, whose leading terms for base-2 logarithms can be written in the form

$$\log x! \simeq x \log x - \frac{x}{\ln 2}. \quad (\text{A9})$$

Dropping the term L_1 from Eq. A8 as discussed previously, we then have

$$\begin{aligned} \mathcal{L}_{\text{total}} \simeq & \frac{N}{S} \sum_{k=1}^K H(\hat{\mathbf{g}}^{(k)}) + H(\mathbf{c}) \\ & + \frac{N}{S} \sum_{k=1}^K \sum_{p \in C_k, p \neq m_k} H_{\text{mod}}(\mathbf{g}^{(p)} | \hat{\mathbf{g}}^{(k)}) \\ & + \frac{S-1}{S} \log(S-1) - \frac{S-K}{S} \log(S-K) \\ & - \frac{K-1}{S} \log(K-1), \end{aligned} \quad (\text{A10})$$

where $H(\hat{\mathbf{g}}^{(k)})$ is given by Eq. 2, $H(\mathbf{c})$ is given by Eq. 3, and $H_{\text{mod}}(\mathbf{g}^{(p)} | \hat{\mathbf{g}}^{(k)})$ is given by Eq. 4.

To this expression we now add an additional term $+\lambda K$ to control the number of clusters K , as in Eq. 6. As shown in Appendix B, with this term included the optimal value of K is asymptotically independent of S , so we can safely assume that $S \gg K$ as S becomes large, which allows us to drop the last three terms in A10, giving the form in Eq. 1:

$$\begin{aligned} \mathcal{L}_{\text{total}} \simeq & \frac{N}{S} \sum_{k=1}^K H(\hat{\mathbf{g}}^{(k)}) + H(\mathbf{c}) \\ & + \frac{N}{S} \sum_{k=1}^K \sum_{p \in C_k, p \neq m_k} H_{\text{mod}}(\mathbf{g}^{(p)} | \hat{\mathbf{g}}^{(k)}), \end{aligned} \quad (\text{A11})$$

up to an additive constant.

Appendix B: Number of clusters

In this appendix we demonstrate that the optimal value of K in Eq. 6 is asymptotically constant as the number of samples S grows. For the purposes of our argument we assume that all partitions p have the same number of groups $n_p = n$, that the number of nodes N is fixed and $N \gg n$, and that the cluster sizes c_k are approximately equal. We do not neglect the last three terms in Eq. A10 as we did previously, since our argument here is what allows us to neglect these terms in the first place.

In terms of S , K , and N , the leading order scaling of each of the terms in Eq. A10, along with the linear penalty term $+\lambda K$, is

$$\begin{aligned} \mathcal{L}(S, K) \sim & \frac{KN}{S} \log n \\ & + \frac{N(S-K)}{S} \tilde{H}_{\text{mod}}(K) \\ & + \frac{S-1}{S} \log(S-1) \\ & - \frac{S-K}{S} \log(S-K) \\ & - \frac{K-1}{S} \log(K-1) \\ & + \log K + \lambda K, \end{aligned} \quad (\text{B1})$$

where $\tilde{H}_{\text{mod}}(K)$ is a typical scale for $H_{\text{mod}}(\mathbf{g}^{(p)} | \hat{\mathbf{g}}^{(k)})$. In general $\tilde{H}_{\text{mod}}(K)$ is a decreasing function of K , since a larger number of clusters allows partitions to be assigned to closer modes. We ignore the $\log \Omega/N$ contribution to H_{mod} , as it scales like $n^2 \log N/N$ [25] and can be neglected by comparison with the $O(\log n)$ contribution from the standard conditional entropy when $N \gg n$.

For fixed S , a local minimum of Eq. B1 with respect to K occurs at the first value of K for which

$$\mathcal{L}(S, K+1) - \mathcal{L}(S, K) > 0. \quad (\text{B2})$$

To demonstrate that the optimal value of K remains constant as S increases, we let $S \rightarrow \infty$ in Eq. B1 and show that we can always satisfy Eq. B2 with a finite value of K that is independent of S . Letting $S \rightarrow \infty$ in Eq. B1 with K constant and substituting into Eq. B2 gives

$$\log(1 + 1/K) + \lambda + N[\tilde{H}_{\text{mod}}(K+1) - \tilde{H}_{\text{mod}}(K)] > 0, \quad (\text{B3})$$

where we have discarded terms of order $\log S/S$ and smaller. Rearranging gives

$$\tilde{H}_{\text{mod}}(K) - \tilde{H}_{\text{mod}}(K+1) < \frac{\lambda}{N} + \frac{1}{N} \log(1 + 1/K). \quad (\text{B4})$$

Because $H_{\text{mod}}(K)$ is a decreasing function of K , this inequality will always be satisfied for some constant K , since $H_{\text{mod}}(K) - H_{\text{mod}}(K+1)$ approaches 0 from above and the right-hand side is bounded below by the strictly positive constant λ/N . Thus the optimal value of K in Eq. B1 is asymptotically constant as S grows.

Note that we cannot make the same argument for the unnormalized description length of Eq. A7. In that case the inequality analogous to Eq. B4 is

$$\tilde{H}_{\text{mod}}(K) - \tilde{H}_{\text{mod}}(K+1) < \frac{1}{N} \log(1 + 1/K), \quad (\text{B5})$$

but the right-hand side of this expression goes to zero as K becomes large, so we cannot guarantee there is a finite value of K that satisfies the inequality. In practice, we find that this inequality is not satisfied in many test networks, the optimal K growing monotonically with S .

Appendix C: Additional example applications

In Fig. 4 we show two additional example applications of our method. Figure 4A shows a network of collaborations among researchers in the field of network science [38], which exhibits highly multimodal community structure. In a manner reminiscent of the artificial network of cliques in Fig. 2C, this network consists of many small, tightly connected groups of nodes, which can be arranged in various ways to form plausible community

divisions. As we might expect, the modes identified for this network appear to be comprised of a few of these possible arrangements.

In Fig. 4B we show the modes of a network of associations among terrorists involved in the 2004 Madrid train bombing [39]. In this case, we see that the community structure in the upper region of the network is uncertain, resulting in two substantially distinct community divisions appearing as modes.

-
- [38] M. E. J. Newman, Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E* **74**, 036104 (2006).
- [39] B. Hayes, Connecting the dots: Can the tools of graph

theory and social-network studies unravel the next big plot? *American Scientist* **94**, 400–404 (2006).

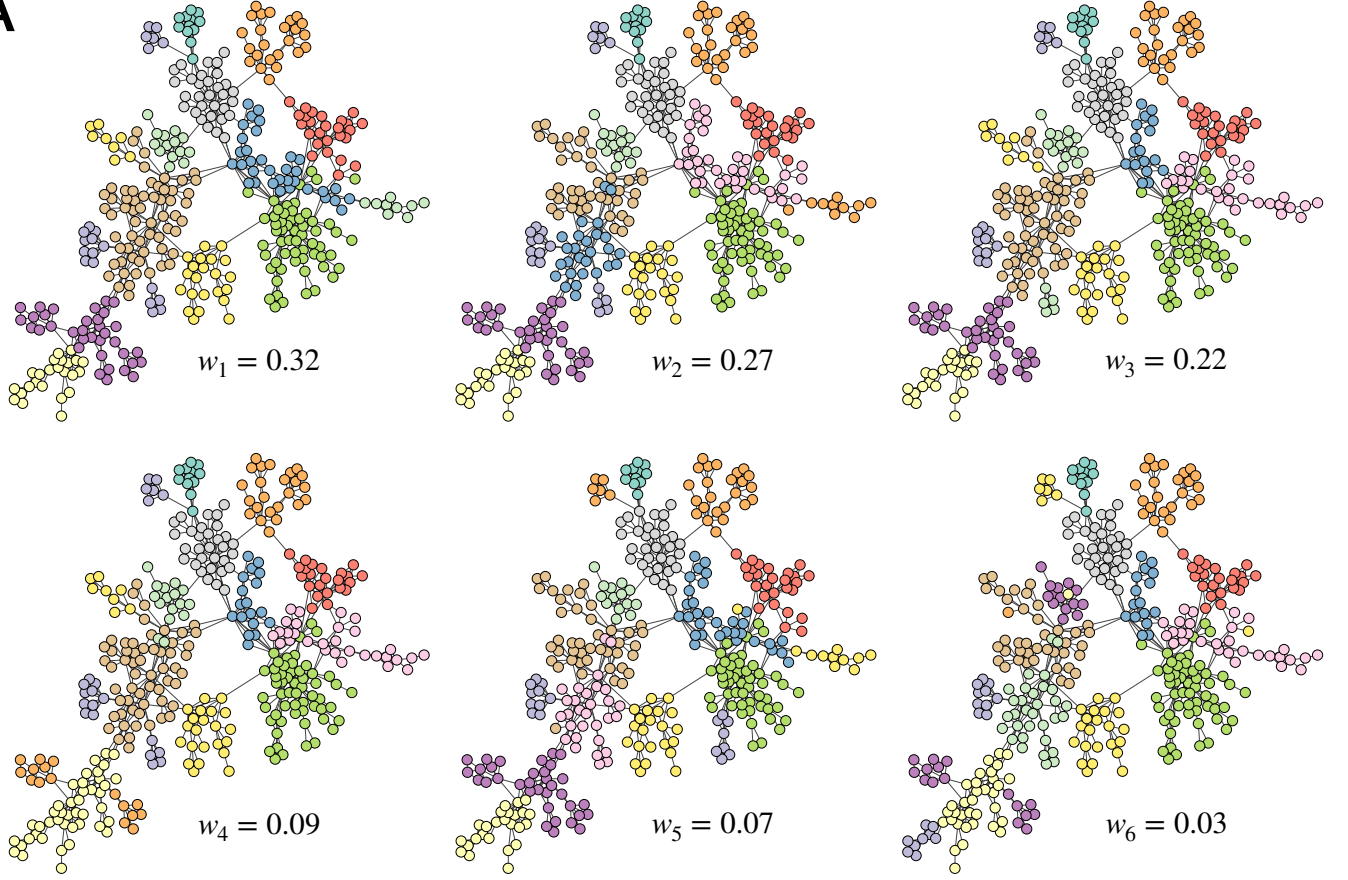
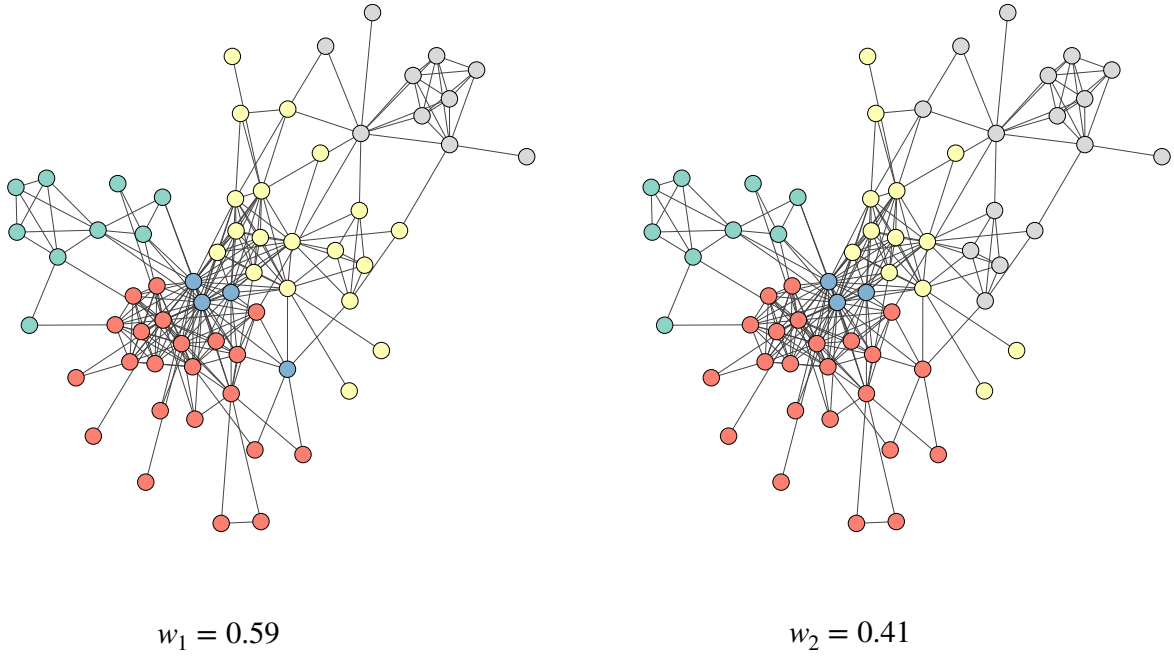
A**B**

FIG. 4: Representative modes and their corresponding weights identified by minimizing Eq. 6 with $\lambda = 1$ for 10,000 community partition samples. (A) Collaboration network among network scientists [38]. (B) Network of terrorist associations [39].